UCTG: A Unified Controllable Text Generation Framework for Query Auto-Completion

Zhipeng Li [1]⋆, Shuang Zheng [1]⋆, Jiaping Xiao[2], Xianneng Li [1]†, Lei Wang[3]

[1] Dalian University of Technology , [2]Nanyang Technological University, [3]Meituan
(lizhipeng,zhengshuang99)@mail.dlut.edu.cn,jiaping001@e.ntu.edu.sg,
wanglei46@meituan.com,xianneng@dlut.edu.cn

Abstract

In the field of natural language generation (NLG), controlling text generation (CTG) is critical, particularly in query auto-completion (QAC) where the need for personalization and diversity is paramount. However, it is essentially challenging to adapt to various control objectives and constraints, which results in existing CTG approaches meeting with mixed success. This paper presents UCTG, a unified controllable text generation framework, which introduces a novel prompt learning method for CTG. Specifically, this framework seamlessly integrates a control module, a prompt module, and a generation module. The control module leverages a fine-tuned model to distill user preference features and behavioral patterns from historical data, incorporating human feedback into the model's loss functions. These features are then transformed by the prompt module into vectors that guide the generation module. As such, the text generation can be flexibly controlled without modifying the task settings. By employing this unified approach, UCTG significantly improves query accuracy and coherence in tasks with different objectives and constraints, which is validated by extensive experiments on the Meituan and AOL real-world datasets. UCTG not only improves text generation control in QAC but also sets a new framework for flexible NLG applications.

1 Introduction

Recently, large-scale pre-trained language models (PLMs) have developed as a powerful tool applied in various applications. A PLM can be considered a well-informed knowledge base, allowing for text generation without relying much on extra domain knowledge. Despite their outstanding capabilities, PLMs present a significant challenge in controllability. The diverse and often imbalanced nature of pre-training data can lead to uncontrollable content generation. Controllability is vital in text generation, necessitating the imposition of various constraints to meet specific requirements across different scenarios, such as adherence to storylines in story generation, emotion or topic constraints in dialogues, and personalization in Query Auto-Completion (QAC) (Yin et al., 2020). Furthermore, there's an ethical need to avoid generating harmful or biased content in AI applications.

To address these challenges, the current methodology in controllable text generation (CTG) involves introducing control signals into PLMs, ensuring the generated content aligns with specified conditions (Prabhumoye et al., 2020). Compared to other controllable approaches, such as retraining models at the data level or post-processing the generated results, which are more costly, lack sufficient labeled data with a control signal, and may not yield satisfactory results, fine-tuning the PLMs is the most effective, direct, and cost-efficient method. These fine-tuning techniques (Min et al., 2023), such as adaptive modules (Lin et al., 2021) , prompt-based approaches (Li and Liang, 2021a) (Lester et al., 2021), and instruction tuning (Ouyang et al., 2022), have been well investigated and employed.

However, existing methods are struggling in scenarios requiring simultaneous control over multiple conditions and the integration of human knowledge, as in QAC. QAC is a key technique employed by search engines to enhance user queries, aiming to better comprehend user intent. Traditionally, suggestions have depended on either term-frequency-based methods, lacking semantic understanding of the query, or word-embedding-based methods with little personalization efforts (Zhong et al., 2020). These methods are usually known as the most popular completion (MPC) (Bar-Yossef and Kraus, 2011), where the scores are significantly

⋆ equal contribution, † corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 679–688
January 19–24, 2025. ©2025 Association for Computational Linguistics

high for popular queries and notably low for rare queries (Fiorini and Lu, 2018). Thus, personalization and diversity are two crucial control constraints for QAC during text generation, which are challenging to handle simultaneously by existing CTG methods.

This paper introduces UCTG, a unified CTG framework addressing these limitations in QAC text generation. This framework consists of three main components, namely the control module, the prompt module, and the generation module. This UCTG framework not only enhances the controllability and personalization of text generation in QAC but also offers a versatile model for various NLG applications, setting a new benchmark in the field. Compared to other QAC methods, the key contributions of this paper are:

- We propose a novel prompt mechanism for integrating human feedback into the design of prompts for CTG, using fine-tuned BERT models to encapsulate user preferences and behavior patterns, thereby enriching the information embedded in prompts. In the generation module, these “meaningful vector prompts" serve as a prompt learning tool for PLMs, allowing for controlled, personalized, and context-aware text generation in QAC scenarios.
- We design a unified framework capable of setting multiple control conditions simultaneously for text generation, allowing for greater flexibility and effectiveness in various text generation scenarios, particularly in QAC.
- We conduct extensive experiments on the Meituan and AOL real-world datasets, showcasing UCTG's robustness in improving query accuracy and coherence, thereby validating its practical efficacy in diverse QAC contexts.

2 Related Work

2.1 Controllable Text Generation

The most direct way is to conduct fine-tuning on the Pre-trained Language Models (PLMs), enabling cost-effective execution of the Control Text Generation (CTG) task. The Adapted Module, Prompt and Instruction Tuning are three commonly used fine-tuning methods for Control Text Generation (CTG).

The adaptive modules fundamentally seek to bridge the gap between the controlled attributes and the Pre-trained Language Models (PLMs)(Lin et al., 2021; Zhang et al., 2020, 2019). The prompt-based methods essentially leverages the characteristics of Pre-trained Language Models (PLM) during the pre-training phase. This methods guide the PLM to generate constrained text by selecting a suitable prompt during the fine-tuning stage, aiming to achieve controllability(Zhang and Song, 2022). InstructGPT, a notable recent work, employs instruction tuning to guide the language model and produce desired, human-like content. This approach enables precise control over the model, ensuring the generation of answers that align with human expectations (Ouyang et al., 2022). The challenge lies in finding ways to comprehensively and securely align human instructions with Pre-trained Language Models (PLMs).

2.2 Query Auto-Completion

Query Auto-Completion (QAC) is a technique used in search engines and recommendation systems to suggest and complete user queries (Gog et al., 2020) (Bar-Yossef and Kraus, 2011) (Zhong et al., 2020). It aims to provide users with relevant query suggestions as they type, improving the search experience. The current approaches to QAC can be categorized into two main types: one is the retrieve-and-rank method, and the other is the text generation method.

The conventional approach of QAC is Most Popular Completion (MPC), which ranks query candidates based on popularity derived from historical query logs. However, MPC tends to provide poor predictions when the query prefix is extremely short. To improve ranking quality, retrieve-and-rank methods have been proposed. Subsequently, these retrieved queries are ranked using features such as frequency, similarity to the previous query, user profile, etc (Shokouhi, 2013; Cai et al., 2014). However, this method faces challenges with cold start and inaccurate suggestion of short prefixes.

Another research direction uses seq2seq and language models for generating query suggestions based on a given prefix (Dehghani et al., 2017; Sordoni et al., 2015). Mustar et al. perform fine-tuning on pre-trained language models, such as BERT, to produce auto-complete suggestions (Mustar et al., 2020). Although this approach addresses the issue of ’unseen queries’ by overcoming the limitation of the candidate pool through parameterized models, it faces a more significant challenge of ’weak personalization’. This is due to the insufficient utilization of valuable personal information, such as the historical behavior sequence (Yin et al., 2020). At the same time, this approach could potentially generate non-sensical auto-complete suggestions. More over, when selecting a pre-trained model for text generation, the higher occurrence of certain items in the training data increases the likelihood of their appearance during generation. Consequently, this gives rise to the issue of lacking diversity in text generation.

Thus, when choosing the generative model for QAC tasks, effective control over the personalization and diversity of the generated text content is crucial. However, all existing QAC methods currently are unable to accomplish query completion under different control objectives. The framework proposed in this paper is capable of accommodating query text completion under different control conditions.

2.3 Prompt Learning

Prompt learning is an innovative learning strategy applied to a Pre-trained Language Model (PLM). It transforms a downstream task into [MASK] prediction format using the PLM by incorporating templates into the input texts. The design of prompt templates is a core component in prompt learning (Zhang and Wang, 2023). Prompt can be categorized as discrete, continuous and hybrid prompt (Petroni et al., 2019; Schick and Schütze, 2020; Li and Liang, 2021b; Liu et al., 2021). In this paper, to more effectively incorporate prior human preferences as prior knowledge for CTG, we have designed a novel prompt module. This prompt utilizes a high-dimensional user behavior model representation extracted from a fine-tuned BERT model as the prompt. It is employed to control the text generation of the PTMs.

3 THE UCTG FRAMEWORK

We propose the UCTG framework for controllable text generation with prompt learning, a soft-prompt-based approach to guide text generation models for generating controllable queries under multiple control conditions.

In the control module, we designed a novel prompt module to incorporate prior human preferences as prior knowledge for CTG. The core idea is to derive a controllable continuous vector from user historical behavior data. We leverage a BERT model, fine-tuned on extensive user input and behavior data, to learn representations of user behavior patterns, encompassing both queries and final user actions. We introduce two loss functions in this module, designed to capture both user preference feedback and diversify candidate items simultaneously, thereby refining control signals for QAC text generation. Then, the prompt module transforms these refined user behavior patterns into meaningful vector prompts, aligning them with the GPT model's vector space. This novel approach of creating prompts, rooted in user behavior and preferences, offers a more nuanced and effective control mechanism compared to traditional text prompts and soft prompts (Liu et al., 2023), which often lack this level of personalization and context awareness. In the generation module, these “meaningful vector prompts" serve as a prompt learning tool for PLMs, allowing for controlled, personalized, and context-aware text generation in QAC scenarios. Note that this prompt is a high-dimensional feature vector with specific semantic meanings, thereby influencing and controlling the output of PTMs.

The framework can flexibly incorporate multiple control signals. By setting appropriate loss functions in the BERT model of the control module, various types of human feedback signals can be introduced. These may include user preferences, user's desire for diversity in generated content, user's preferences for attributes in generated content, and more. The UCTG provides a framework for controllable text generation, allowing the generation of text with controllable properties under different conditions.

3.1 Problem Formulation

We use A, B, and Y to denote the sets of user profiles, user's history inputs, and user's history behaviors, respectively. The function f represents the user's behavior pattern, describing the functional relationship between user profiles, user input queries, and their final decision-making behavior.

We employ a deep learning model to learn the function f. In this paper, we utilize the BERT model for this purpose. In the BERT model, we use click behavior and user preferences for diverse queries as supervised signals for loss functions, thereby obtaining prior knowledge from user feedback.

$$f(B_1, \cdots, B_n, A_1, \cdots, A_m) = \{y_c, y_d\}. \quad (1)$$

We extract the high-dimensional hidden layer of

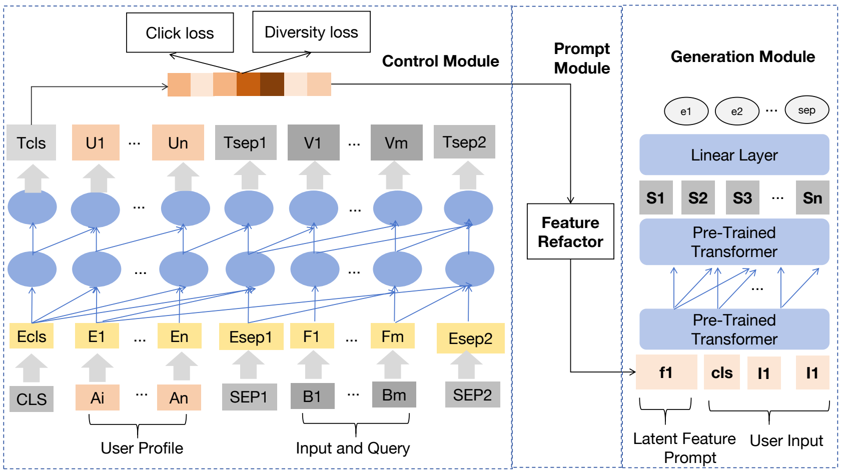

Figure 1: A high-level overview of the UCTG framework. This framework integrates three modules for QAC. The control module employs a fine-tuned BERT model to analyze user input and behavior data with multiple control conditions, extracting user preferences and behavioral patterns. The prompt module then translates these patterns into structured prompts, aligning them with the GPT model's vector space for effective control signal integration. Lastly, the generation module utilizes these tailored prompts to guide the GPT model, ensuring controlled and personalized text generation for various QAC scenarios.

BERT as the prompt vector. This prompt vector contains rich control information that can be utilized to control the generation of text in the text generation module.

$$C_{prompt} = E(f(*)) \tag{2}$$

In the context of CTG with PLMs, the majority of methods leverage the generative model as the foundation and direct it to produce the intended text. Generally, CTG tasks treat PLMs as conditional generation models. The objective of conditional text generation can be formulated as follows.

$$P(X|C_{prompt}) = \prod_{i=1}^{n} p(x_i|x_{<i}, C_{prompt}). \tag{3}$$

Where C_{prompt} represents the controlled conditions, which will be incorporated into the PLM in a specific form. Each user input $I = \{i_1, i_2, \ldots, i_m\} \in \mathcal{I}$ consists of a sequence of m words. And X is the generated text that incorporates the knowledge encoded in the PLM and complies with the control conditions.

3.2 UCTG Framework and Modules

3.2.1 Control Module

In our model, we use pre-trained BERT to learn the relation between user profile, user history input, and the user's history behavior. By configuring the loss of the BERT model, user feedback can be effectively incorporated to obtain control information. We combine token sequences of user profile A and user's history inputs B as the input of the Bert model. [CLS] is used as the representation for the whole input. [SEP] not only marks the sentence boundary but is also used by the model to learn when to terminate the decoding process.

The attention mechanism allows for capturing the dependencies between different representations, regardless of their distance in the sequence.

3.2.2 Prompt Module

Prompt learning essentially enhances the information density of the input, which means the addition of more prior knowledge. The denser the prior information provided by the prompt, the more effective the text-generation performance of the generative model. However, simple hard prompts and soft prompts are insufficient in providing sufficient prior information. In this paper, the high-dimensional features extracted by the BERT model in the control module are used as prompt vectors for the GPT-2 model. The prompt module is composed of a multi-layer neural network that maps the high-dimensional features extracted by Bert to the text embedding space of the GPT model. It resizes the feature vectors generated by the control module.

3.2.3 Generation Module

The input of the generation module consists of the prompt vector from the prompt module and the vector of the user's input text. In this paper, the generation module is based on the GPT2 model. With the control signal of human feedback contained in the prompt vector, the GPT2 model can generate queries that are more in line with user preferences and more diverse for users.

3.3 Multi-Task Learning of Control Module

Due to the rich user feedback information contained in the user's historical behavioral data, to incorporate multi-perspective human feedback as prior knowledge more flexibly and provide a unified framework, we adopted a multi-task approach with various control conditions in the control module. Through the design of loss functions, multiple control signals are simultaneously configured. To learn from the supervision of customer behaviors, we introduce a click-through rate (CTR) prediction task and a query item diversity preference task into our framework. The outputs of the Bert model for these two tasks are $\hat{y_1}$ and $\hat{y_2}$.

$$\hat{y_1} = \sigma(W_1 H^l + b_1) \tag{4}$$

$$\hat{y_2} = \sigma(W_2 H^l + b_2) \tag{5}$$

The loss functions of these two tasks are as follows.

$$min_{\Theta} \mathcal{Z} = -\frac{1}{B}\sum_{i=1}^{B} yi \log{(\hat{y_i})} + (1 - y_i) \log{(1 - \hat{y_i})} \tag{6}$$

In the CTR Prediction task, the output of the Bert is a probability $\hat{y_1}$, which represents the CTR of the specific query item candidate. For training, we construct positive and negative samples from query logs: the clicked queries as positive samples, and the randomly selected queries that haven't been clicked as negative samples. In the Query Item Diversity Preferences task, we use user preferences for popular and long-tail queries as supervised signals for training the model. The output of the Bert for this task is $\hat{y_2}$. This task provides prior information about user preferences for the diversity of queries.

4 Experiments

4.1 Datesets

We conduct experiments on the widely used benchmark dataset AOL and a real-world dataset from Meituan. The statistics information about the AOL and MeiTuan datasets is shown in Table 1.

Dateset	AOL	MeiTuan
Number of users	66,000	439,431
Number of queries	1,484,974	412,226
Number of items	1,243,631	6,238,211
Number of samples	3,614,503	31,993,676

Table 1: Statistics of AOL and MeiTuan

The Meituan query dataset includes search session ID, date, user ID, user's personal information, user input, candidate words recommended by Meituan, user's click on the candidate query, and whether the user made a purchase after clicking the candidate query, among other behavior feedback data. Compared to the AOL dataset, which only contains user input and the final clicked website name, the Meituan query log dataset has richer user behavior feedback, allowing for better modeling of user preference features.

4.2 Evaluation Metrics

We evaluate all the baselines and the proposed model with two evaluation metrics. We use the Bilingual Evaluation Understudy (BLEU) to assess the quality of our generated text and the Gini coefficient to evaluate the diversity of our generated text.

4.2.1 Baselines

We have selected generative baselines for comparison. The following baselines are used for our experimental evaluation: Long Short-Term Memory (LSTM)(Hochreiter and Schmidhuber, 1997), Gated Convolutional Neural Network(GatedCNN)(Dauphin et al., 2017), Transformer (Vaswani et al., 2017), The Generative Pre-trained Transformer 2 (GPT-2)(Radford et al., 2019).

4.3 Results and Analysis

4.3.1 Overall Performance Comparison

Table 2 summarizes the experimental results on the Meituan Query Log datasets, respectively. Overall, our proposed UCTG framework outperforms all the traditional generative models.

First, the Transformer has shown superior generation performance among traditional generative models compared to traditional RNN-based and CNN-based models. This is because the Transformer excels at natural language modeling for capturing long-range context dependencies. Second, compared to hard and soft prompts, the models

Models	BLEU-1↑	BLEU-2↑	BLEU-3↑	BLEU-4↑	$BLEU_{ave}$ ↑	Gini↓
LSTM	0.1916	0.1094	0.0732	0.0577	0.1080	0.1136
GRU	0.1907	0.1085	0.0729	0.0576	0.1074	0.1151
GatedCNN	0.1870	0.1076	0.0723	0.0568	0.1059	0.1066
Transformer	0.2252	0.1383	0.0937	0.0729	0.1325	0.2861
hard-prompt fine tuning	0.2052	0.1216	0.0825	0.0665	0.1190	-
soft-prompt fine tuning	0.1388	0.0801	0.0552	0.0446	0.0797	-
GPT2-distil	0.1844	0.1068	0.0732	0.0585	0.1057	0.1230
UCTG GPT2-distil (F)	0.1945	0.1148	0.0776	0.0613	0.1120	0.1091
UCTG GPT2-distil (NF)	0.1949	0.1152	0.0776	0.0612	0.1122	0.1089
GPT-2	0.2001	0.1180	0.0815	0.0654	0.1163	0.1220
UCTG GPT2(F)	0.2134	0.1277	0.0870	0.0687	0.1242	
UCTG GPT2 (NF)	0.2645	**0.1684**	**0.1152**	**0.0894**	**0.1594**	0.1425
UCTG multitask GPT2 (F)	0.2035	0.1194	0.0795	0.0630	0.1163	0.0707
UCTG multitask GPT2 (N)	0.2038	0.1196	0.0796	0.0631	0.1165	

Table 2: Overall performance of the models on Meituan Query Log datasets. 'F(freeze)' refers to updating only the parameters of the prompt module without updating the parameters of GPT2. 'NF(no freeze)' refers to updating both the parameters of the prompt module and GPT2. 'multitask tuning' indicates that during the training of BERT in the Control module, it undergoes multi-task training. Hard-prompt finetuning uses the user's historical clicks as prompt information; Soft-prompt finetuning uses two token positions as adjustable prompt parameters.

Models	BLEU-2↑	BLEU-3↑	BLEU-4↑	$BLEU_{ave}$ ↑	BLEU-2↑	BLEU-3↑	BLEU-4↑	$BLEU_{ave}$ ↑
LSTM	0.1094	0.0732	0.0577	0.1080	0.1548	0.1134	0.0867	0.1447
GRU	0.1085	0.0729	0.0576	0.1074	0.1543	0.1126	0.0861	0.1440
GatedCNN	0.1076	0.0723	0.0568	0.1059	0.0895	0.0502	0.0372	0.0741
Transformer	0.1383	0.0937	0.0729	0.1325	0.1568	0.1190	0.1097	0.1562
GPT2-base	0.1180	0.0815	0.0654	0.1163	0.1558	0.1184	0.0909	0.1503
UCTG tuning GPT2-base (F)	0.1277	0.0870	0.0687	0.2546	0.1742	0.1317	0.1029	0.1658
UCTG tuning GPT2-base (NF)	0.1684	0.1152	0.0894	0.1594	0.1744	0.1318	0.1031	0.1660

Table 3: Results of the models on Meituan and AOL datasets

fine-tuned within the UCTG framework demonstrate better performance. Third, the Gini index of the model fine-tuned with UCTG (multi-task) prompts is significantly lower than that of the normally trained model. Other generation models also demonstrate varying degrees of enhanced Matthew effect. The notable decrease in the Gini index after fine-tuning with the UCTG framework confirms the effectiveness of the embeddings extracted by the control module.

In general, the GPT-2 model, after being fine-tuned with the UCTG framework, exhibits improved performance in text control by utilizing directed prompts from the control module.

4.3.2 Ablation Study

In this section, we conducted ablation studies to evaluate the impact of different model components on overall performance. The ablation studies include: 1) GPT-2 models of different sizes (Figure 2)); different prompt tuning methods (Figure 3); and 3) different control module loss functions (Figure 4). The detailed results of the ablation experiments are presented in the appendix. The corresponding results demonstrate the effectiveness of our proposed framework and its components.

4.3.3 Verification experiment in the AOL Dataset

Table 3 presents the performance metrics of the models on two datasets: Meituan Query Log and AOL Query Log. The performance trends of the models on the AOL dataset are generally similar to those on the Meituan dataset. Additionally, it can be observed that the improvement in performance after fine-tuning with the UCTG framework is more pronounced on the Meituan dataset compared to the AOL dataset. This can be attributed to the fact that the Meituan dataset is domain-specific.

5 Conclusion

In this paper, we introduce a Unified Controllable Text Generation Framework with novel prompt learning based on human feedback to enable controllable text generation. This framework could provide flexible configurations of control conditions tailored to various tasks. Extensive experiments on the Meituan and AOL datasets show that our method surpasses state-of-the-art baselines. In practical applications of QAC, our proposed framework effectively addresses the issues of lack of personalization and diversity in CTG by incorporating valuable human feedback as prior knowledge.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant 72071029 and 72231010.

References

Ziv Bar-Yossef and Naama Kraus. 2011. Context-sensitive query auto-completion. In *Proceedings of the 20th international conference on World wide web*, pages 107–116.

Fei Cai, Shangsong Liang, and Maarten De Rijke. 2014. Time-sensitive personalized query auto-completion. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1599–1608.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1747–1756.

Nicolas Fiorini and Zhiyong Lu. 2018. Personalized neural language models for real-world query auto completion. *arXiv preprint arXiv:1804.06439*.

Simon Gog, Giulio Ermanno Pibiri, and Rossano Venturini. 2020. Efficient and effective query auto-completion. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2271–2280.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021a. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021b. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The adapter-bot: All-in-one controllable conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16081–16083.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2020. Using bert and bart for query suggestion. In *Joint Conference of the Information Retrieval Communities in Europe*, volume 2621. CEUR-WS. org.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. *arXiv preprint arXiv:2005.01822*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Milad Shokouhi. 2013. Learning to personalize query auto-completion. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 103–112.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, pages 553–562.

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

Di Yin, Jiwei Tan, Zhe Zhang, Hongbo Deng, Shujian Huang, and Jiajun Chen. 2020. Learning to generate personalized query auto-completions via a multi-view multi-task attentive approach. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2998–3007.

Hanqing Zhang and Dawei Song. 2022. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. *arXiv preprint arXiv:2210.09551*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Zizhuo Zhang and Bang Wang. 2023. Prompt learning for news recommendation. *arXiv preprint arXiv:2304.05263*.

Jianling Zhong, Weiwei Guo, Huiji Gao, and Bo Long. 2020. Personalized query suggestions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1645–1648.

A Appendix

A.1 Evaluation Metrics

BLEU: For our experiments, BLEU evaluates the degree of lexical match between the ground-truth complete query and the first-ranked generated query.

Gini coefficient: In this paper, we use the Gini coefficient to measure the diversity of generated content, specifically examining the distribution proportion of popular and long-tail items in the generated content. The lower the value, the fairer the generation system.

$$Gini = \frac{1}{n-1}\sum_{j=1}^{n}(2j-n-1)p(j) \qquad (7)$$

where n is the total number of candidate queries, j is the index of each query, and $p(j)$ is the proportion of the total candidate queries that belong to the j-th query.

A.2 The Detailed Results of the Ablation Study

Figure 2 demonstrates that the UCTG with the base size exhibits better text generation performance compared to the UCTG with the distil size, while keeping other modules constant and only varying the size of the GPT-2 model.

The results in Figure 3 demonstrate the comparison between the UCTG fine-tuning framework and two conventional prompt fine-tuning methods (hard prompt fine-tuning and soft prompt fine-tuning). As shown in the figure, the performance improvement from UCTG fine-tuning significantly surpasses that of the other two methods.

The results in Figure 4 demonstrate a significant decrease in the Gini index is observed after fine-tuning the GPT-2 model using the prompt vectors generated by the control module with multi-task loss. This ablation experiment highlights the deeper value of the UCTG framework. By substituting the loss function of the control module in the UCTG framework, the control module can generate prompt vectors with specific control effects and effectively control the behavior of the GPT-2 model.

A.3 Case Study and Visualization

To examine what BERT has learned in our UCTG frame work, Figure 5, 6 and 7 depicts the visual representation of prompt vector. These four figures depict the visual representation of embedding vectors using a subset consisting of 1000 data points from the test dataset. All the visualizations are generated using the TensorFlow Embedding projector.

Figures 5 utilize T-SNE for visualizing 1000 data points. Taking the bottom left corner of Figure 5 as an example, upon examining the corresponding users and their input information for each point within the cluster, it is discovered that this cluster includes “The user 2907007597.0 inputted the word fried chicken”,“The user 1533072098.0

https://projector.tensorflow.org/

inputted the word braised chicken","The user 613293787.0 inputted the word fried chicken" and "The user 40962466.0 inputted the word stuffed chicken with pig stomach" etc. From the textual content, it can be inferred that the embedding vectors correspond to user inputs related to chicken-related food. Furthermore, the historical clicks of these users are predominantly associated with food as well. This further confirms the alignment between the control module's modeling of user preferences in the UCTG framework and human common knowledge. It also underscores the practical significance of the embeddings generated by the control module, as they can provide prompts to control the text generation of GPT-2.

Figures 6, 7 employ UMAP (Uniform Manifold Approximation and Projection), a dimensionality reduction algorithm for high-dimensional data, to visualize 1000 text data points and their embeddings. UMAP, as compared to T-SNE, is capable of capturing the global structure. From the visualization of the dimensionality reduction, it can be observed that even with the change in the visualization algorithm, "The user 2907007597.0 inputted the word fried chicken", "The user 2561854230.0 inputted the word roast chicken" and "The user 2270230980.0 inputted the word chicken soup", etc. still cluster together. These texts primarily revolve around food-related topics, and the historical clicks of these users further indicate their preferences in the food domain.

The visualization results effectively demonstrate that the high-dimensional vectors generated by the control module in the UCTG framework genuinely represent the users' behavioral preferences and characteristics. This further confirms that the notable enhancement in model performance after fine-tuning UCTG in specific experiments is indeed attributed to the prompting effect of the prompt vectors.

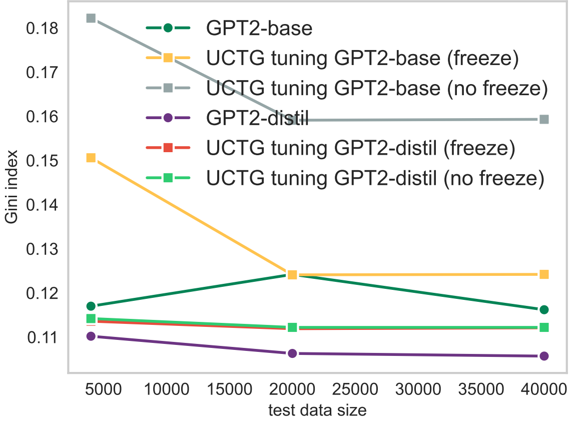

Figure 2: Generation module with different size GPT2

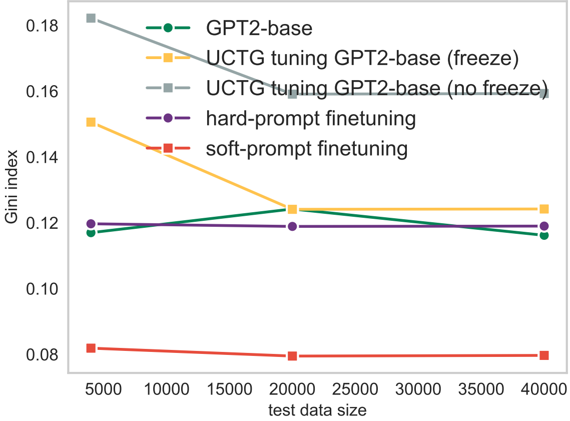

Figure 3: Prompt module with different prompt tuning approaches

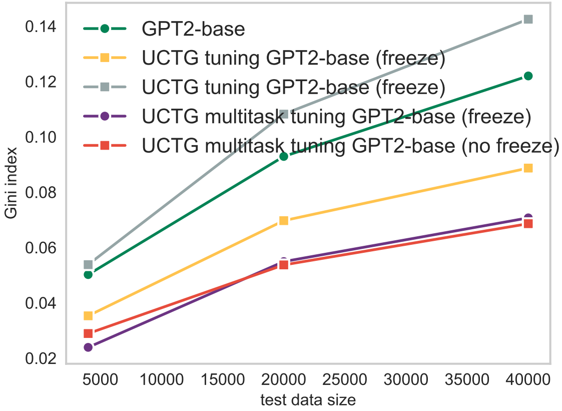

Figure 4: Ablation study of UCTG under different control conditions

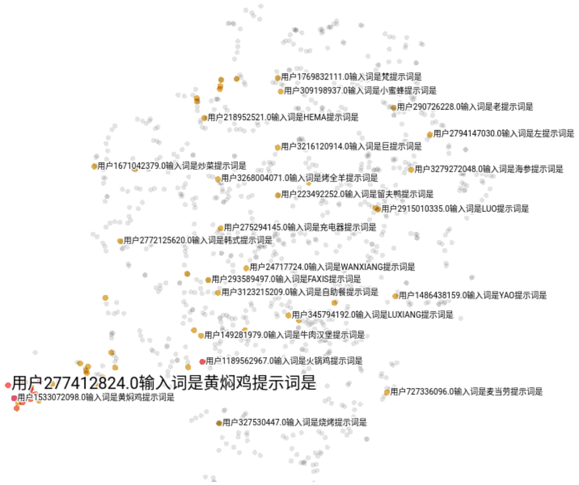

Figure 5: Natural text sentences

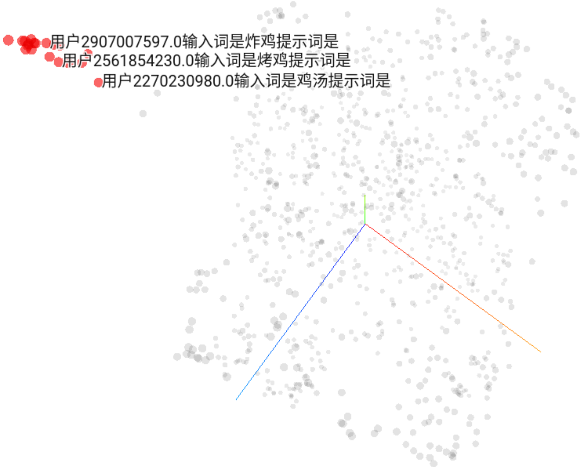

Figure 6: Natural text sentences using UMAP

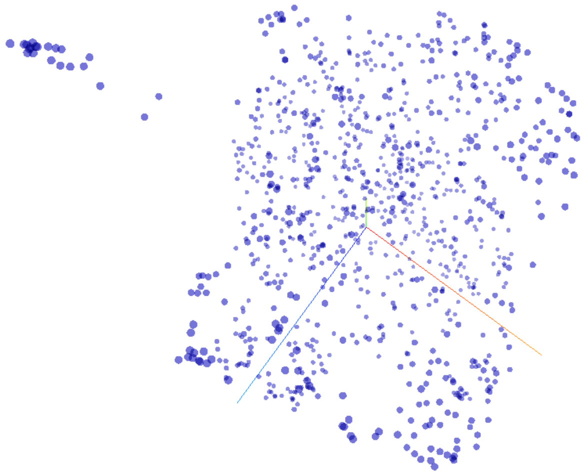

Figure 7: Prompt vectors in the embedding space using UMAP