Unified Contextual Query Rewriting

Yingxue Zhou[1], **Jie Hao**[1], **Mukund Rungta**[2]*, **Yang Liu**[1], **Eunah Cho**[1], **Xing Fan**[1], **Yanbin Lu**[1], **Vishal Vasudevan**[1], **Kellen Gillespie**[1], **Zeynab Raeesy**[1], **Wei Shen**[1], **Chenlei Guo**[1], **Gokhan Tur**[1]

[1] Amazon Alexa AI, [2] Georgia Institute of Technology

[1]{zyingxue,jieha,yangliud,eunahch,fanxing,luyanbin,vasuvish,kelleng raeesyzr,sawyersw, guochenl,gokhatur}@amazon.com, [2]mrungta8@gatech.edu

Abstract

Query rewriting (QR) is an important technique for user friction reduction (i.e. recovering ASR error or system error) and contextual carryover (i.e. ellipsis and co-reference) in conversational AI systems. Recently, generation-based QR models have achieved promising results on these two tasks separately. Although these two tasks have many similarities such as they both use the previous dialogue along with the current request as model input, there is no unified model to solve them jointly. To this end, we propose a unified contextual query rewriting model that unifies QR for both reducing friction and contextual carryover purpose. Moreover, we involve multiple auxiliary tasks such as trigger prediction and NLU interpretation tasks to boost the performance of the rewrite. We leverage the text-to-text unified framework which uses independent tasks with weighted loss to account for task importance. Then we propose new unified multitask learning strategies including a sequential model which outputs one sentence for multi-tasks, and a hybrid model where some tasks are independent and some tasks are sequentially generated. Our experimental results demonstrate the effectiveness of the proposed unified learning methods.

1 Introduction

Large-scale conversational AI agents such as Alexa, Siri, and Google Assistant, are becoming increasingly popular in real-world applications to assist users in daily life. However, some of the user interactions lead to dissatisfied experiences, where users do not get what they requested or the assistant has to engage with the user again to clarify the user request. These user frictions arise from errors in the system, including Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU), as well as user ambiguity and background noise. The goal of QR (Hao et al., 2022; Cho et al., 2021) is to identify the queries that lead to friction and rewrite them to queries without changing the users' intention, in order to mitigate defective interactions. Besides, in a multi-turn dialogue session with agent, users sometimes tend to use incomplete utterances which usually omit or refer back to entities or concepts that appeared in the previous dialogue, namely ellipsis, and co-reference. Thus, we also always rely on contextual carryover (Elgohary et al., 2019; Liu et al., 2020) to rewrite the incomplete query into a context-dependent and self-contained query.

Although query rewriting has received a lot of attention in recent years, it has always been studied in two separate directions, i.e., reduce defects and contextual carryover. Recent works integrate all functionalities into pre-trained Sequence-to-Sequence (Seq2Seq) language models and report impressive results (Su et al., 2021; Raffel et al., 2020). This synergy has resulted in a great deal of recent work developing transfer learning methodology for NLP. Inspired by that, we propose a unified contextual query rewriting framework, that can utilize one model to perform the rewrite for both friction reduction and contextual carryover. Moreover, we involve two additional tasks in the unified model: NLU interpretation of rewrite and rewrite trigger prediction. NLU interpretation of rewrite is a task to predict the domain, intent, and entity slots information. Trigger prediction is a task that enables the model to decide to trigger a rewrite or not. These tasks can not only be used for downstream modules but also serve as auxiliary tasks to boost the primary task query rewriting performance.

Specifically, we leverage the BART model (Lewis et al., 2020) which is a large-scale Seq2Seq framework. We unify defect reduction and contextual carryover as one QR task, where the model input is a dialogue context along with the current request, and the output is

*Work done while Mukund Rungta was interning at Amazon.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 608–615

the rewrite. For NLU interpretation and trigger prediction tasks, we also cast them as a text-to-text generation task, where the model output the trigger decision and the NLU interpretation for the current request. Motivated by the concept of in-context learning (Brown et al., 2020), to steer the model to solve different sub-task, we plug a task-specific prompt, into the model input. This way, the generations of different sub-tasks are decoupled, leading to better flexibility of the model regarding generation for each sub-tasks. Besides the traditional text-to-text unified learning approach (Raffel et al., 2020) in which each task's prediction is generated independently, we explored variants of unified learning approach, including sequential unified learning where one sequence is used to generate multi-task results using target prompts and hybrid unified learning where some tasks are independent and some tasks are sequentially generated. We conducted extensive offline experiments to study the proposed unified learning approaches. Our experimental results demonstrate the effectiveness of the proposed approaches. Our production simulation validates the positive impact of the proposed model which indeed generates rewrites of better quality.

2 Related Work

Query Rewriting In dialogue systems, query rewriting benefits dialogue state tracking especially co-reference resolution (Rastogi et al., 2019; Hao et al., 2021), and reducing users' friction by replacing the users' utterance (Wang et al., 2021; Fan et al., 2021; Chen et al., 2023). Fan et al. (2021) and Cho et al. (2021) propose to leverage the search-based model, which consists of a DSSM based retrieval layer and a tree ranking layer, to handle global and personalized query rewriting. Su et al. (2019) use generation-based approaches to tackle the co-reference and omission-specific scenarios. Hao et al. (2022) propose a constrained generation based Seq2Seq model for query rewriting.

Contextual Carryover Contextual carryover has been an important component in dialogue systems for resolving co-reference and omission. Naik et al. (2018) leverage an encoder-decoder architecture for making independent carryover decision for each slot in the context. Later, Chen et al. (2019) propose a framework to jointly predict whether a subset of related slots should be carried over from dialogue history. Rastogi et al. (2019) formulate the contextual carryover problem as a contextual *query rewriting problem* (CQR). Yu et al. (2020) present a few-shot generative approach to conversational query rewriting. In this work, we unify the query rewiring task with CQR by one text-to-text generation model, with generated rewrite handling contextual slot carryover cases.

Unified Learning Transfer learning in natural language processing (NLP) has gained popularity due to its demonstrated effectiveness. This approach involves pre-training a model on a data-rich task and fine-tuning it for a specific task (Dong et al., 2019; Radford et al., 2018; Lewis et al., 2020). Later, the efficacy of transfer learning has been further improved by a unified framework that converts all text-based language problems into a text-to-text format presented through the T5 model (Raffel et al., 2020). In this paradigm, instead of adapting the pre-trained language model (LM) to downstream tasks via objective engineering, downstream tasks are reformulated to look more like those solved during the original LM training with the help of a textual prompt.

3 Methodology

Before introducing the unified learning model, we first establish the baseline for the query rewriting. We formulate the query rewriting as a sequence-to-sequence (Seq2Seq) task and fine-tune a pre-trained BART model for the rewrite generation. As shown in Figure 1, the model takes the current turn and its previous dialog context as input and generates the target text autoregressively. The Seq2Seq architecture comprises a bidirectional encoder that takes the context and current request as input, and an autoregressive decoder that performs constrained decoding to generate the target rewrite.

3.1 Generation based query rewrite

Query rewrite with contextual carryover. In addition to rewriting defective queries such as "play night talk by drake" to "play knife talk by drake", we consider contextual carryover task as a query rewriting task as well. For example, in a multi-turn dialog, we have "[USER] what's the current temperature at Colorado Springs [AGENT] Right now, it's 46 degrees Fahrenheit. Today, expect a high of 75 degrees. [USER] what's the air quality", where location slot "Colorado Springs" needs to be a carryover to current turn "[USER] what's the

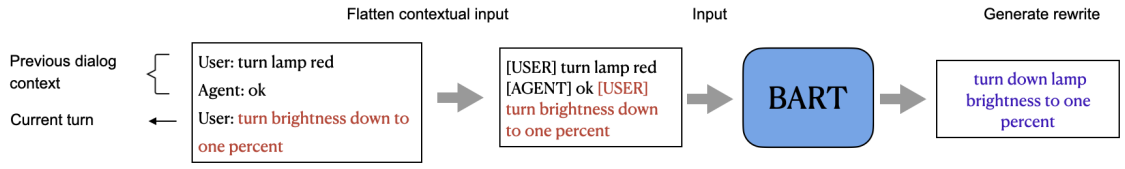

Figure 1: Illustration of the seq2seq model query rewriting model. When a new utterance arrives, the model takes the flattened contextual input and outputs as the final rewrite. We use [USER] as a special symbol added in front of the user turn, and [AGENT] as a special symbol added in front of the agent turn.

air quality". In such cases, we directly address the carryover problem by formulating it as a rewriting task, with the goal of generating the rewrite "What's the air quality in Colorado Springs."

We adopt the pre-trained BART (Lewis et al., 2020) which has the same model architecture as the widely-used Transformer model (Vaswani et al., 2017) and is pre-trained with a denoising way (Devlin et al., 2018). As illustrated in Figure 1, we flatten the previous dialogue turns (including both user requests and agent responses) and the current user request into a single sequence for input to the encoder. Then, we fine-tune BART for our task.

Formally, given a contextual request sequence $\mathbf{q} = \{q_1, ..., q_M\}$, where q_i for $i \in \{1, ..., M\}$ denotes a token in the sequence, and the corresponding rewrite $\mathbf{r} = \{r_1, ..., r_N\}$. The encoder is responsible for reading the input request and its previous dialogue turns, and the decoder autoregressively generates the rewrites.

The ultimate goal of the rewrite generation problem is to learn a probability distribution $p_\theta(\mathbf{r})$ over the variable-length text sequence $\mathbf{r}$, where θ is the parameter of the BART. Typically, the maximum likelihood estimation (MLE) objective is used to train the language model which is defined as

$$\mathcal{L}_\theta(\mathbf{q}, \mathbf{r}) = -\frac{1}{|\mathbf{r}|} \sum_{j=i}^{|\mathbf{r}|} \log p_\theta(r_j | \mathbf{r}_{<j}) \ .$$

Typically, given finite training examples, i.e., T pairs of contextual query and rewrite $S = \{\mathbf{q}_t, \mathbf{r}_t\}_{t=1}^{T}$, the model is trained by minimizing the empirical finite sample objective loss function $\mathcal{L}_\theta(S) = \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_\theta(\mathbf{q}_t, \mathbf{r}_t)$.

3.2 Other tasks

We first introduce the tasks we consider in this paper as follows.

NLU hypothesis generation task. In conversational AI systems, interpreting the user's query, such as their intent and domain, helps downstream modules to respond more effectively to the user's request. Integrating this interpretation task into the model can improve its understanding of the user's request and context. The results of the NLU interpretation can not only be utilized by downstream modules but also act as regularizers for the primary query rewriting task. To integrate the NLU interpretation task, we let the model generate such interpretation as an NLU hypothesis that takes the form of "domain | intent | slot_type:slot_value". For example, given the query "play bad blood by taylor swift", the corresponding NLU hypothesis would be "Music | PlayMusic | SongName:bad blood | ArtistName:taylor swift". The NLU hypothesis generation task takes the query as input and generates the corresponding NLU hypothesis.

Trigger prediction task. The trigger task allows the model to predict whether a rewrite (or contextual carryover) is necessary for the incoming query. For example, if the query "play bad blood by Taylor Swift" is not defective, the model should not trigger a rewrite. Typically, separate and independent models are used to make this binary decision. However, in our unified model, we formulate this binary prediction problem as a text generation problem. Queries that do not require a rewrite have a target output of "no trigger", while defective queries have a target output of "trigger". Integrating trigger tasks in the unified generation model can save the resources for having separate trigger models.

3.3 Parallel unified learning model

Figure 2 illustrates the design of parallel unified learning which is similar to T5 (Raffel et al., 2020). We fine-tune the BART model on the above three tasks. The rationale behind unifying multi-tasks in training is that by successfully predicting the system's interpretation of a request (i.e., domain, intent, and slots in the NLU hypothesis) and its trigger decision, the model can improve its prediction of the trigger task and subsequent rewrite. As shown in Figure 2, we add the prompt "predict trigger:" to the contextual query as input: "***predict trigger:*** *turn brightness down to one percent"* and

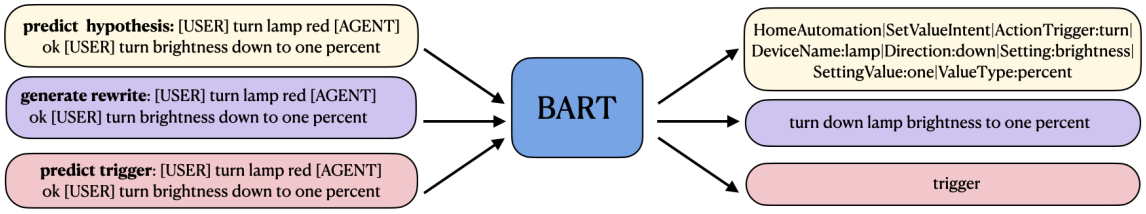

Figure 2: Illustration of the parallel multi-task unified learning model. For each task, the model takes the current turn with its previous dialog context and the task-specific prompt as input to generate the corresponding target text, i.e., prompt *predict hypothesis:* for NLU task, *predict trigger* for trigger task, and *generate rewrite* used for rewrite task.

the target output of trigger prediction task is *"trigger"*. For the NLU interpretation task, the model takes "***predict hypothesis:*** *turn brightness down to one percent"* as input and generates the corresponding hypothesis. For the rewriting task, we add prompt "**generate rewrite**" to the input. To account for the varying importance of each task, we incorporate a weighted loss in our multi-task unified training approach. Given the incoming contextual query $\mathbf{q}$, and target rewrite $\mathbf{r}$, target NLU hypothesis $\mathbf{h}$ and target trigger prediction $\mathbf{g}$, we use weighted loss as follows

$$\mathcal{L}_\theta(\mathbf{q},\mathbf{r},\mathbf{h},\mathbf{g}) = \lambda_1\, \mathcal{L}_\theta(\mathbf{q},\mathbf{r}) + \lambda_2\, \mathcal{L}_\theta(\mathbf{q},\mathbf{h}) + \lambda_3\, \mathcal{L}_\theta(\mathbf{q},\mathbf{g})\ ,$$

where λ_1, λ_2, and λ_3 are the weights for rewrite, NLU, and trigger tasks separately.

3.4 Sequential unified learning model

In this section, we propose a novel unified learning approach by leveraging a single (i.e. text generation) task of text generation with multiple prompts. Our model encodes the current request and its previous dialog context and then generates a sequential output that predicts various tasks. To guide the predictions, we use markup tokens such as "[rewrite]", "[trigger]", and "[hypothesis]" to prompt the prediction. This approach leverages the benefits of conditional generation, allowing the model to consider the previous task's prediction when performing the next task's prediction. As a result, the order of task generation is important. We also consider the same three tasks for training this model: NLU hypothesis task, trigger task, and rewrite generation task. The model generates the prediction for each task in a single sequence with the order: rewrite $\rightarrow$ trigger $\rightarrow$ hypothesis. Figure 3 illustrates the idea of this sequential multi-task unified learning model.

3.5 Hybrid unified learning model

Parallel and sequential unified learning approaches both have advantages and disadvantages. The parallel multi-task approach trains each task independently for a given query, which requires duplicating training data by the number of tasks and leads to a longer training cycle. As the number of tasks increases, the size of the training data also increases, making it challenging to add more tasks. Besides, each task is trained independently, the model cannot leverage the correlation between tasks. On the other hand, the sequential unified approach does not require duplicate data since the model nests multitasks into one sequential output, reducing the training cost. This model has the potential to learn and leverage the correlation between tasks. However, the sequential model has higher latency due to the longer decoding length compared with the parallel multi-task model. In addition, it is hard to apply weighted loss for sequential multi-tasks.

To address these issues and combine the benefits of both models, we proposed a hybrid unified learning model as shown in Figure 4. The hybrid model considers the rewrite generation and trigger prediction as a nested sequential task, with the prompt "generate_rewrite_trigger:". The NLU hypothesis generation is treated as an independent parallel task, with the prompt "predict hypothesis:". This hybrid model reduces the duplication of training data, reduces the training cost, and leverages the correlation between tasks. We also apply weighted loss training loss to reflect the importance of tasks:

$$\mathcal{L}_\theta(\mathbf{q},\mathbf{r},\mathbf{h},\mathbf{g}) = \lambda_1\, \mathcal{L}_\theta(\mathbf{q},\{\mathbf{r},\mathbf{g}\}) + \lambda_2\, \mathcal{L}_\theta(\mathbf{q},\mathbf{h})$$

where λ_1, λ_2 are the weights for nested rewrite_trigger task and NLU tasks separately.

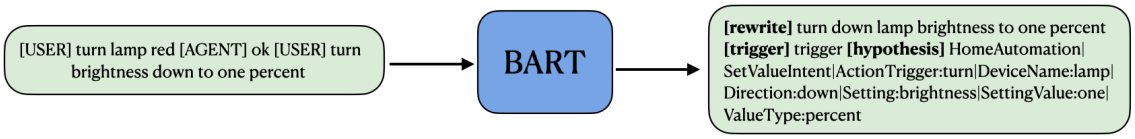

Figure 3: Illustration of the sequential unified learning model. The model sequentially generate a single output of different tasks. We have use special tokens (e.g., "[rewrite]", "[trigger]", "[hypothesis]") to prompt the prediction for different tasks.

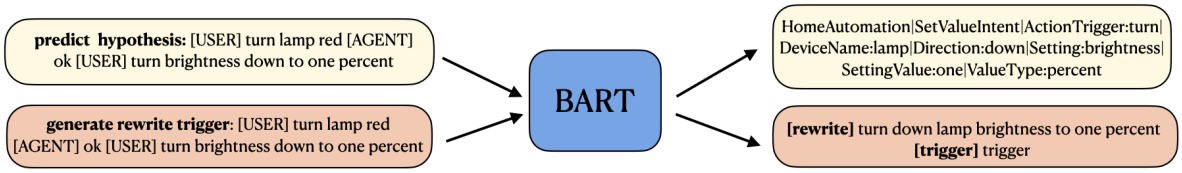

Figure 4: Illustration of the hybrid multi-task unified learning model. The model uses task-specific prompt to generate the task-specific target text, i.e., prompt "predict hypothesis" for NLU hypothesis task, "generate_rewrite_trigger:" for trigger task and rewrite task.

4 Experiments

We conduct two sets of experiments to evaluate the proposed unified learning models. The first set of experiments compares the three proposed unified models in terms of their effectiveness on the query rewrite, trigger prediction, and NLU hypothesis generation. The second set of experiments evaluates the benefit of integrating contextual carryover into the query rewriting task.

4.1 Experimental Setup

Datasets We use two datasets for our experiments: the query rewrite dataset (QueryR) and the contextual carryover rewrite dataset (CarryoverR). The QueryR dataset is weakly annotated by a defect detection model to identify consecutive user utterances where the first turn was defective but the second turn was successful. We also collect 1M non-defective quires which do not need to be rewritten/triggered (i.e., trigger task label is "no trigger"). The ContextCarryoverR dataset is human-annotated for contextual carryover queries. In this dataset, we have 1M queries need carryover and 1M queries do not need carryover (i.e., trigger task label is "no trigger"). We collect 1-month period data for training and validation (randomly split by 9:1) and subsequent 1-week period data testing. Table 1 provides the information of each dataset. Note there is no overlap between QueryR and CarryoverR, i.e., QueryR does not have contextual carryover queries and CarryoverR only has contextual carryover queries. Note that all the data has been de-identified.

Datasets	Trigger label	Train	Test
QueryR	trigger	7M	200k
	no trigger	1M	200k
CarryoverR	trigger	1M	908
	no trigger	1M	4340

Table 1: Statistics of the query rewriting data sets.

Model Setup In the first set of experiments, we only focus on the QueryR dataset which does not have any contextual carryover queries. We train the *parallel*, *sequential*, and *hybrid* unified models on QueryR dataset by fine-tuning the BART-base model, which has 140M parameters, following must-task learning approaches in Sections 3.3, 3.4 and 3.5. We compared the proposed unified learning with the baseline **CGF** (Hao et al., 2022) which only consider query rewrite task.

In the second set of experiments, explore the advantages of unifying query rewrite with contextual carryover. Thus we train the BART-base trained on CarryoverR dataset for rewrite generation as the baseline (name this baseline as BART_CR). We also have another baseline that we combine CarryoverR and QueryR datasets and train the BART-base on the combined dataset for rewrite generation (name this baseline as BART_CR_QR). For the proposed unified method, we train the hybrid unified model on the combined dataset using rewrite, trigger, and NLU tasks (name this unified model as Hybrid_CR_QR).

Evaluation Metrics. In practice, the query rewriting system is not expected to rewrite or trig-

Task	Rewrite	Trigger	NLU
	precision	F1	precision
CGF	78.44%	NA	NA
Parallel	79.01%	0.89	68.59%
Sequential	65.67%	0.90	66.11%
Hybrid	79.98%	0.91	67.78%

Table 2: Compare parallel, sequential, and hybrid unified models with existing CGF query rewriting on QueryR test set. The Hybrid model achieves the best precision and F1 score.

Dataset	QueryR	CarryoverR
BART_CR	NA	68.51%
BART_CR_QR	78.62%	72.37%
Hyrbid_CR_QR	80.21%	78.45%

Table 3: Rewrite precision at 20% trigger rate of baselines and Hybrid unified model on QueryR and CarryoverR test sets. Hybrid unified model achieves much higher precision than baselines do.

ger every query from the users, taking into account cases where the query itself may not be defective or need a contextual carryover. Thus, To evaluate rewrite and NLU hypothesis quality, we use *utterance-level precision* at a fixed *trigger rate*, i.e., 20% trigger rate. The utterance level precision denotes how often the triggered rewrite matches the correct rewrite. We use the *F1* score as the trigger task evaluation metric. For CarryoverR test data, we also use *hallucination* as metrics. A rewrite is considered hallucinated if it contains entities that are not present in the target utterance. We also evaluate *intrinsic* hallucination (when the hallucinated entities are present in the input) and *extrinsic* hallucination (when the hallucinated entities are not present in the input).

4.2 Experimental Results

Unified models on QueryR For parallel model and hybrid model, we have explored the weights for different tasks. By conducting a grid search, we identify the optimal choice of weight for parallel model is 0.7, 0.1, 0.2 for rewrite, trigger and NLU tasks. For hybrid model the optimal choice of weight is 0.8, 0.2 for rewrite-trigger task and NLU task. Table 2 presents the results of the rewrite generation precision at a 20% trigger rate on the QueryR test set, as well as the F1 score for the trigger task. The Hybrid model outperforms the other models by achieving the best precision in the rewriting task. The sequential model has a performance regression due to its longer decoding sequence when adding the hypothesis task. In the hybrid model, the trigger task output is conditioned at the rewriting task, which can explain the higher F1 score for the Hybrid model compared with the Parallel model where the trigger task is learned independently. Overall, the Hybrid model is favored in terms of good rewrite performance, trigger performance, and shorter decoding time.

	Hallucination	Intrinsic	Extrinsic
BART_CR	46.23%	30.42%	20.79%
BART_CR_QR	44.69%	31.82%	18.14%
Hyrbid_CR_QR	39.71%	27.78%	17.11%

Table 4: Hallucination rate (intrinsic and extrinsic hallucination rates) of baselines and Hybrid model. The Hybrid model achieves the lowest hallucination rate.

Unified model on QueryR and CarryoverR Table 3 shows the precision at 20% trigger rate of the unified model on QueryR and CarryoverR test sets. The results of BART_CR and BART_CR_QR indicate that unifying the contextual carryover task with the query rewrite task can improve the carryover performance, even under a single-task training approach. The Hybrid model achieved the highest precision, demonstrating further improvement through multi-task learning. Table 4 displays the hallucination rates, including intrinsic and extrinsic hallucination rates, on the CarryoverR test set. The results indicate that the Hybrid model has the lowest hallucination rates.

Production simulation We also conduct the production simulation of the proposed Hybrid model (Hybrid_CR_QR). We gather one-week live traffic data from our production system and input the data into the model. We compare the proposed model with the no-unified model rewrites within the English-speaking user's environment. We use one primary metric to evaluate the rewrite performance: defect rate, which is calculated as the number of defective rewritten utterances, divided by the total number of rewritten utterances. We use the defect detection model in Gupta et al. (2021) to measure if an utterance is defective. In the analysis, we observe a 16.65% reduction in the defect rate and an increase of millions of new rewrites per week. Table 5 provides examples showing the

defect reduction case
USER: put satellite ho by monica AGENT: I couldn't find satellite ho by Monica, but here is other music by Monica . USER: i said sideline hope by monica Baseline rewrite: play sideline hope by monica Unified model rewrite: play sideline ho by monica
contextual carryover case
USER: who's the tallest man in the world AGENT: Sultan Kosen is the tallest man alive. The tallest man across history is Robert Wadlowski. USER: how tall is it Baseline rewrite: how tall is sultan kosen Unified model rewrite: how tall is robert wadlowski

Table 5: Production examples of Hybrid Unified model and baseline.

effusiveness of the unified model.

4.3 Limitations

We acknowledge that there are certain limitations of this framework. First, generation-based models have latency issue due to the autoregressive generation. Thus, we will explore Non-autoregressive and semi-autoregressive methods in a future study. Second, the knowledge is only stored in model parameters which limits the capacity of the model to make the smarter trigger decision through fact-checking and generate a valid rewrite. To this end, we intend to consider a retrieval-augmented generation to incorporate external knowledge to improve performance as well as incorporating more contextual (e.g. if the user is listening music) and personalized (e.g. user preference) signals into the model. Moreover, generative models can also pose quality control challenges, such as hallucinations. To mitigate this issue, we will add constrained decoding (Hao et al., 2022) to control hallucinations.

5 Conclusion

In this work, we propose unified learning approaches for QR. The proposed approach unifies several tasks into one text-to-text model. Besides, the proposed approach unifies general rewrite tasks with contextual carryover tasks. We explored multiple unified learning scenarios such as parallel multi-task learning, sequential multi-task learning, and hybrid multi-task learning. Our experimental results and production simulation demonstrated the superiority of the unified learning model.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. volume 33, pages 1877–1901.

Tongfei Chen, Chetan Naik, Hua He, Pushpendre Rastogi, and Lambert Mathias. 2019. Improving long distance slot carryover in spoken dialogue systems. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 96–105.

Zheng Chen, Ziyan Jiang, and Fan Yang. 2023. Graph meets llm: A novel approach to collaborative filtering for robust conversational understanding. *arXiv:2305.14449*.

Eunah Cho, Ziyan Jiang, Jie Hao, Zheng Chen, Saurabh Gupta, Xing Fan, and Chenlei Guo. 2021. Personalized search-based query rewrite system for conversational ai. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 179–188.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *EMNLP*.

Xing Fan, Eunah Cho, Xiaojiang Huang, and Chenlei Guo. 2021. Search based self-learning query rewrite system in conversational ai. In *2nd International Workshop on Data-Efficient Machine Learning (DeMaL)*.

Saurabh Gupta, Xing Fan, Derek Liu, Benjamin Yao, Yuan Ling, Kun Zhou, Tuan-Hung Pham, and Chenlei Guo. 2021. Robertaiq: An efficient framework for automatic interaction quality estimation of dialogue systems. In *2nd International Workshop on Data-Efficient Machine Learning (DeMaL)*.

Jie Hao, Yang Liu, Xing Fan, Saurabh Gupta, Saleh Soltan, Rakesh Chada, Pradeep Natarajan, Chenlei Guo, and Gökhan Tür. 2022. Cgf: Constrained generation framework for query rewriting in conversational ai. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 475–483.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2021. Rast: Domain-robust dialogue rewriting as sequence tagging. In *Proceedings*

of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4913–4924.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. Incomplete utterance rewriting as semantic segmentation. In *EMNLP*.

Chetan Naik, Arpit Gupta, Hancheng Ge, Mathias Lambert, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. *Proc. Interspeech 2018*, pages 596–600.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pushpendre Rastogi, AI Alexa, Arpit Gupta, and Tongfei Chen. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In *Proceedings of NAACL-HLT*, pages 97–105.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. *arXiv preprint arXiv:1906.07004*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhuoyi Wang, Saurabh Gupta, Jie Hao, Xing Fan, Dingcheng Li, Alexander Hanbo Li, and Chenlei Guo. 2021. Contextual rephrase detection for reducing friction in dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1899–1905.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1933–1936.