CGF: Constrained Generation Framework for Query Rewriting in Conversational AI

Anonymous ACL submission

Abstract

In conversational AI agents, Query Rewriting (QR) plays a crucial role in reducing users frictions and satisfying their daily demands. Users frictions are caused by various reasons, such as errors in the spoken dialogue system, users' accent or their abridged language. In this work, we present a novel Constrained Generation Framework (CGF) for query rewriting at both global and personalized level. The proposed framework is based on the encoder-decoder framework and consists of a context-enhanced encoding and constrained generation decoding phrases. The model takes the query and its previous dialogue context information as the encoder input, then the decoder relies on the pre-defined global or personalized constrained decoding space to generate the rewrites. Extensive offline and online A/B experimental results show that the proposed CGF significantly boosts the query rewriting performance.

1 Introduction

Large-scale conversational AI agents such as Alexa, Siri and Google Assistant help millions of users to perform a lot of tasks, such as playing music, controlling light devices at home, etc. In general, such conversational AI agents have two components: automatic speech recognition (ASR) and natural language understanding (NLU). ASR is responsible for converting speech signals of user query (e.g. "play Michael Jackson music") to a text transcript. Following this, NLU provides domain/intent classification (e.g. domain: Music, intent: PlayMusic) and entity labelling (e.g. ArtistName: Michael Jackson), which are used to fulfill the user's request.

However, users sometimes suffer friction due to errors occurred in the speech recognition. In detail, ASR module may mis-recognize utterance due to background noise or users' accent. For example, ASR error led to a erroneous transcript "*play alien bridges*", when the user actually meant "*play leon bridges*". Due to such errors, the downstream NLU system is affected, capturing a wrong entity "*alien bridges*" for the slot "*ArtistName*". This leads to a fractured user experience and they may need to rephrase their query. Moreover, the friction might happen due to the NLU cannot handle the users requests. For example, "tv to input three" cannot handled by NLU instead of user's intended "turn tv to h.d.m.i. three". Thus, in order to reduce the friction and make the dialog system more robust, query rewriting (QR) (Ponnusamy et al., 2019; Chen et al., 2020) becomes an increasingly important technique in the conversational AI agents.

Many existing QR systems in conversational AI described in the literature commonly involve complex search-based pipelines for either global-wise query rewriting (Fan et al., 2021; Chen et al., 2020) or personalized query rewriting (Cho et al., 2021). A search system mainly comprises of two stacks operating sequentially: retrieval and ranking. The global or personalized indexer constructs the global or personalized index by using the users historical defect-free interactions with agent. Whenever a new request arrives, the system compares it to existing utterances in the index using a series of dual encoder retrieval model with FAISS search (Johnson et al., 2017) and retrieves top N candidates from the index. Then the retrieved candidates are ranked by a ranking model with both neural semantic and IR features as input. The system picks the top 1 ranked candidates as the final rewrite. Such search-based system is widely used in the large scale conversational AI agents as it can easily and effectively control the output by index and thus risk averse.

However, we cannot ignore the limitations posed by (1) the error accumulation in multi-stage system; (2) lacking of fine-grained interactions in current search-based models; (3) a large memory footprint is needed to store dense representations when considering large index in retrieval layer. Also, few

research work targets query rewriting with considering the previous context information, although they admit the importance and the context information has been proven useful in many other NLP tasks (Wang et al., 2017; Wu et al., 2018).

In this work, we propose to solve query rewriting task by leveraging generation-based models under Constrained Generation Framework (CGF), which is to generate the rewrites left to right, token-by-token in an autoregressive fashion and conditioned on the previous context. Specifically, instead of input the query for the model encoder, we input the previous context including users requests and agent response to encoder. Then, we deploy the constrained decoding when inference to force the generated rewrite in a predefined candidate set. The proposed CGF enables us to mitigate the aforementioned shortcomings from search-based system as the autoregressive formulation allows the model to directly capture relations between contextual input and target rewrites and thus effectively cross encode both. Moreover, the memory footprint is greatly reduced because the parameters of our encoder-decoder architecture scale with vocabulary size, not index count. Also, neural language generation approaches are known to hallucinate content, the constrained decoding with a predefined candidate set helps generation model to be faithful to the model input and avoid the potential hallucinations or bad rewrites.

Finally, we conduct extensive offline experiments for both global query rewriting and personalized query rewriting to show the effectiveness of the proposed approach. Online experiments and case studies reveal that the proposed CGF indeed generates rewrites of better quality and less risks.

The main contributions of this work are as follows:

- We introduced CGF, which consists context-enhanced encoding and constrained decoding.
- CGF enables generative models to perform personalized query rewriting for the first time.
- We provide both offline and online experiments to validate the effectiveness of the proposed CGF approach.

2 Related Work

2.1 Query Rewriting

In dialogue system, on the one hand, the query rewriting serves for the dialogue state tracking especially for the reference resolution (Rastogi et al., 2019; Vakulenko et al., 2020). On the other hand, query rewriting can seamlessly replace the use's utterance in order to remove friction and unsatisfactory experience to users (Ponnusamy et al., 2019). To do this, (Ponnusamy et al., 2019) propose to reformulate the queries with Markov Chain. Chen et al. (2020) propose a retrieval-based model with pre-training method to reduce the customer's friction. Fan et al. (2021) and Cho et al. (2021) leverage multi-stage search-based system to perform global and personalized query rewriting. In this work, we propose CGF based on Seq2Seq model to generate a rewrite of the initial query of user.

Another thread of work which is very like the query rewriting is Grammatical Error Correction (GEC) task. GEC is the task of correcting different kinds of grammatical errors in text such as spelling, punctuation, grammatical, and word choice errors. Recently, Seq2Seq TRANSFORMER has become state-of-the-art approach for GEC (Zhao et al., 2019; Wang et al., 2019; Kaneko et al., 2020), in which the model aims to corrects an ungrammatical sentence to a grammatical sentence. Therefore, the main difference between GEC and our query rewriting is that this task is more concerned with grammatical corrections, and we focus on the errors from usrs, ASR or NLU system to reduce the friction.

2.2 Constrained Generation

Constrained generation has been applied in many tasks like machine translation and web search. Hokamp and Liu (2017) introduce grid beam search to allow the inclusion of pre-specified lexical constraints. Mohankumar et al. (2021) apply constrained decoding with diverse sibling search algorithm for search advertising. To the best of our knowledge, ours is the first work which introduce the constrained decoding into the query rewriting for conversational AI agents. Moreover, we extend the approach for personalized rewriting so that it takes full advantage of the constrained generation.

3 CGF for Query Rewriting

As shown in Figure 1, we introduce the sequence-to-sequence (Seq2Seq) model to generate the

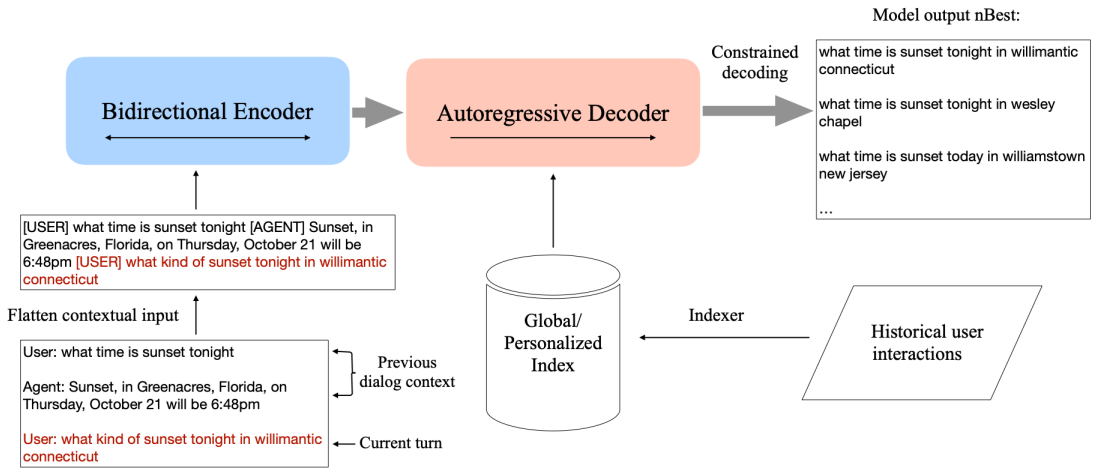


Figure 1: Illustration of the Constrained Generation Framework (CGF) for query rewriting. When a new utterance arrives, the model performs the contextual encoding and constrained decoding and outputs the final rewrites. "Model output nBest" denotes the candidates in a beam from beam search.

rewrite, where a bidirectional encoder takes the context information and request as input, an autoregressive decoder relies on the pre-defined index to perform the constrained decoding in order to generate the target rewrite.

3.1 Context-enhanced Encoding

For the generative query rewriting, we adopt the Seq2Seq pre-trained model BART (Lewis et al., 2020). BART has the same model architecture with the widely-used Seq2Seq Transformer model (Vaswani et al., 2017) and it is pre-trained with a denoising way (Devlin et al., 2019).

In this work, we directly fine-tune the BART. Instead of training BART to maximize the conditional distribution of the (request, rewrite) pairs, we flatten the previous dialogue turns (including both user request and agent response) and the current request into a single sequence for the encoder input, as shown in the Figure 1.

Formally, given a pair of context-enhanced request $\mathbf{Q} = \{q_1, ..., q_M\}$, the conditional probability of its corresponding rewrite $\mathbf{R} = \{r_1, ..., r_N\}$ is defined as:

$$\mathbf{P}(\mathbf{R}|\mathbf{Q}) = \prod_{n=1}^{N} \mathbf{P}(\mathbf{r_n}|\mathbf{R_{<n}}, \mathbf{Q}; \theta), \quad (1)$$

where r_n denotes the n-th target token. θ denotes the parameters of the BART model, which are optimized to minimize the following loss function over the training corpus D:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{Q},\mathbf{R})\sim D}[-logP(\mathbf{R}|\mathbf{Q}; \theta)] \quad (2)$$

The auto-regressive generation process is basically achieved upon the encoder-decoder framework. The encoder is responsible for reading the input request and its previous dialogue information, the decoder auto-regressively generates the rewrites. Given the embeddings of the context-enhanced request and rewrite by equations (3) and (4), the conditional probability of the n-th target word $\mathbf{r_n}$ is calculated as following:

$$\mathbf{H}_{Enc} = \text{ENC}_{BART}(\mathbf{Q}^0), \quad (3)$$

$$\mathbf{H}_{Dec} = \text{DEC}_{BART}(\mathbf{R}^0, \mathbf{H}_{Enc}) \quad (4)$$

$$\mathbf{P}(\mathbf{r_n}|\mathbf{R_{<n}}, \mathbf{Q}; \theta) = Softmax(Proj(\mathbf{h_n})) \quad (5)$$

where ENC_{BART} is the BART encoder to read the context-enhanced request, and DEC_{BART} is the BART decoder to read the target input rewrite and conduct the cross attention on the encoder output. $\mathbf{h_n}$ is the n-th hidden representation of $\mathbf{H}_{Dec}$. $Proj()$ and the $Softmax()$ are two transformation functions in the output layer of the decoder (Vaswani et al., 2017).

3.2 Constrained Decoding

Neural language generation approaches are known to hallucinate content, resulting in generated text that conveys information that did not appear in the input. In general, one of the typical types of hallucination is factual inconsistency generation. In query rewriting, both of them will generate the defective utterance and hurt the user experience. For example, if a user has a request "*play broadway girls by morgan wallen*", the model with free-style generation can have the generated rewrite "*play*

broadway girls by morgan wade". It's a wrong fact generation actually as the "*morgan wade*" never sings the song "*broadway girls*". However, general generative model leverage the beam search over all the whole vocabulary and thus there is a good chance of generating fluent but factually wrong sentences. Thus, the inability to effectively control the generated text have become one of the biggest obstacles for letting generative models for query rewriting in conversational AI. In this work, we consider to have the constrained decoding for generative models to reduce the potential bad rewrites.

The beam search is widely used in Seq2Seq models during inference to improve the search quality. The standard beam search consists of selecting the top B hypothesis with the maximum sum of log probability $S(R,Q) = S(r_{<t}|Q) + logP(r_t|r_{<t},Q)$ at each time step t. Since we want to output the rewrite from U (we can regard it as a pre-defined rewrite candidate set), we cannot use traditional Beam Search while decoding. Allowing to generate any token from the vocabulary at every decoding step might lead the model to generate output strings that are not valid (i.e. bad rewrite). Hence, we resort to constrained beam search, forcing to only decode valid rewrite from a predefined candidate set. Beam Search only considers one step a head during decoding so we can only constrain the generation of a single next token conditioned on the previous ones. Thus, we define our constrain in terms of a prefix trie T, where nodes are annotated with tokens from the vocabulary. For each node $t \in T$, its children indicate all the allowed continuations from the prefix defined traversing the trie from the root to t. More formally, when decoding the token r_t at time step t, the constrained probability distribution is calculated as:

$$\widetilde{P} = \begin{cases} P(r_t = r|r_{<t}, Q), & \text{if } r \in \text{suffix}_T(r_{<t}) \\ 0, & \text{otherwise} \end{cases}$$

where we remove all the tokens r which are not a suffix of the already generated sequence $r_{<t}$ in the predefined trie. In this way, we can ensure that the model is only allowed to generate the rewrites from predefined candidates set. In the trie showed in Figure 2, each path from the root node to the leaf node (e.g. [BOS] $\rightarrow$ *play* $\rightarrow$ *staring* $\rightarrow$ *at* $\rightarrow$ *it* $\rightarrow$ [EOS]) represents an utterance that we allow the model to generate. "[BOS]" is the special token for the model to indicate the begin of sequence. Similarly, "[EOS]" denotes the end of sequence.

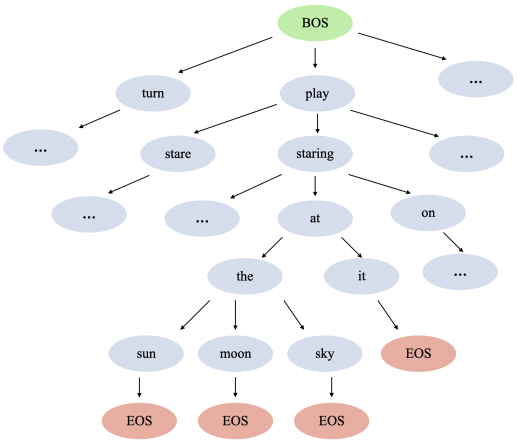


Figure 2: A snapshot of the utterance trie we construct based on global index. When the model has generated a sequence "[BOS] *play staring at*" during the decoding process, in the next step, with the pre-defined trie, the model is only allowed to generate either "*the*" or "*it*". Then, if the model generates "*the*" in the next step, it is only allowed to generate one of the three words "*sun*", "*moon*" or "*sky*" in next next step.

In this work, similar to how to build index for search-based QR models (Fan et al., 2021), we build a trie containing all the defect-free utterances from the user historical interactions of the conversational agent. However, as constrained decoding with trie doesn't require to store dense vectors of index, we can reduce the memory footprint greatly and thus enlarge the trie a lot comparing to the index of search-based models in real online system.

3.3 Global and Personalized Query Rewriting

Constrained generation with pre-fixed decoding space can not only reduce the risky generated rewrites from the generation-based query rewriting models, but also open the door for the generation models to conduct the personalized query rewriting, as we can constrain the decoding space on the user level to reflect the users preference. In this subsection, we introduce how to conduce the global and personalized query rewriting with proposed CGF approach.

Global Query Rewriting Global query rewriting indicates that the rewrite for a certain request is applicable for all users. For example, the user's original query is "*tv to input three*", while the agent cannot well handle it and thus responses "*I'm not quite sure what went wrong.*". The ideal rewrite for this query is "*turn tv to h. d. m. i. three*" which can be applicable to all users who might say

this request. In the proposed CGF, we pre-define the global constrained decoding space where there are all the rewrite candidates that the model is allowed to generate. To achieve this, inspired by the approach to construct the global index in (Fan et al., 2021), we build the global trie which provides rewrite candidates extracted from all users' interactions. The global trie is generated from aggregated, anonymized historical interactions between the users and the agent within a period of time (e.g. 30 days). Specially, after collecting all the users historical interactions, we rely on a defect detection model (Gupta et al., 2021) to filter out the defective utterances. Moreover, We also make sure the impression of the utterance appears in the trie is at least 2. Finally, we got 27M unique utterances in global trie, which will be the constrained decoding space for the global query rewriting.

Personalized Query Rewriting A crucial nature of query rewriting is that often it needs to reflect personal preference or personalized error types to recover from the defect (Cho et al., 2021). For example, when a defective request "*turn on the moon*" from user A and user B comes in, the user A intended request should be "*turn on the moonlight sonata*". However, the user B might want to "*turn on the moon lamp*". Thus, the global query rewriting described above is not a optimal solution to handle such cases. It's necessary to have a personalized query rewriting system to fill this gap. The vanilla Seq2Seq models is not able to conduct personalized generation naturally. However, the proposed CGF can allow the generation models to perform the personalized query rewriting. Specifically, we build the personalized constrained decoding space for each user. For a request comes from a specific user, the model is only allowed to generate a rewrite from the pre-defined personalized decoding space. In order to support personalized rewrites, it's important to build the proper personalized constrained decoding space for each user. We follow Cho et al. (2021) to build the constrained decoding space for each user, leveraging individual interaction history. The utterance included in the constrained decoding space reflects satisfied experiences for each user with in past 30 days of time window.

Data Type	Machine			Human
	Train	Valid	Test	Test
Global QR	1083.0x	66.0x	66.0x	1.0x
Personalized QR	-	-	-	1.0x

Table 1: Query rewriting data sets summary. "Machine" denotes the Machine-Annotated data. "Human" denotes the Human-Annotated data. We report relative size with respect to the global human test set. The training data was extract one month traffic from a large-scale conversational AI agent. The valid/test data are extracted from latter one week traffic.

4 Offline Experiments

4.1 Data

We train and evaluate our proposed method with both weak-labeled data annotated by the model (**Machine-Annotated data**). Specifically, we first leverage a defect detection model such as Gupta et al. (2021) to find two consecutive user utterances, where the first turn was defect, but the second turn was successful. Then, we further filter out the two consecutive user utterances whose time gap is larger than 35 seconds and edit distance is larger than 5, in order to reduce the potential noise in the data. We also evaluate our model with a human-annotated test data (**Human-Annotated data**). For both global and personalized test set, we make sure the target rewrites are in the global/personalized constrained decoding space. Table 1 gives the statistics of the data set. Note all the data has been de-identified.

4.2 Model Setup

In this work, we adopt the pre-trained BART large model provided by (Lewis et al., 2020). Then, we fine-tune the BART model on our query rewriting data sets. We use a batch size of 2048 tokens, dropout rate of 0.1 and adam optimizer. The learning rate is $3e^{-5}$ and linearly warms up over the first 5% steps, then decreases proportionally to the inverse square root of the step number. All the models are trained on eight NVIDIA Tesla V100 GPU.

To evaluate the query rewriting performance, we compare our proposed model with a few baselines. For global query rewriting task, we compare proposed approach with Fan et al. (2021). For personalized task, we have Cho et al. (2021) as the baseline. As trie used in our system is fast to query and small enough to fit in memory, we can enlarge the size of the trie easily. In this work, we follow

System	Machine test set		Human test set	
	Precision	Trigger Rt	Precision	Trigger Rt
UFS-QR (Fan et al., 2021)	59.63	14.52	63.34	15.03
CGF	77.84	51.85	78.24	47.67
Ablations				
CGF w/o context-enhanced encoding	75.03	49.55	76.99	47.17
CGF w/o constrained decoding	75.00	50.34	77.30	47.46
CGF w/o both	72.75	46.55	76.02	46.10

Table 2: Global query rewriting evaluation. We compare our proposed CGF with the existing search-based query rewriting systems on both machine annotated and human annotated test sets.

Fan et al. (2021) to build global trie which contains 27M unique utterances and Cho et al. (2021) to build the personalized trie for each user which contains at most 100 utterances to make the fair comparison. For personalized query rewriting evaluation, we just utilize the model trained with global training data and apply the personalized trie on it for personalized rewriting. We employed the constraints masking the log-probabilities of the invalid tokens and we do not re-normalize the probability over the vocabulary.

4.3 Evaluation Metrics

We perform the evaluation on the utterance level with precision and trigger rate. The precision denotes how often the triggered rewrite matches the correct rewrite. The trigger rate is the fraction of instances on which the model makes a prediction with the final beam score above a threshold [1]. We set the threshold to -0.2 for all the proposed CGF models.

4.4 Global Query Rewriting

Table 2 shows the CGF results on both machine-annotated and human-annotated test sets when incrementally applying the proposed techniques within CGF, i.e. context-enhanced encoding and constrained generation. The experimental result is quite consistent with our intuition. CGF with context-enhanced encoding and constrained decoding gets the best performance on precision and trigger rate across two test sets. Our approach outperforms the search-based UFS-QR system by more than 14% absolute precision for both Machine-Annotated and Human-Annotated test sets. Moreover, the proposed approach can confidently trigger more cases, which is super beneficial for the production rewrite traffic volume increase.

[1]The final beam score is formalized as $P_\theta(y|x) = \Pi_{i=1}^{N} p_\theta(y_i|y_{<i}, x)$, where θ is the model parameters and x is the model input.

System	Precision	Trigger Rt
Personalized-QR (Cho et al., 2021)	71.38	88.80
CGF	73.04	90.33

Table 3: Personalized query rewriting evaluation.

4.5 Ablation Study

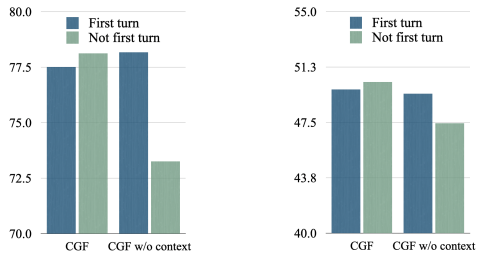
Table 2 also lists the ablation study results for the CGF with global query rewriting task. "w/o both" denotes the CGF without context-enhanced encoding and constrained decoding, in which the model takes the query only as the encoder input and conduct the unconstrained generation. Eventually, using context-enhanced encoding or constrained decoding proved useful. Combining them together is better to get the high precision and at the same time higher trigger rate.

4.6 Personalized Query Rewriting

We here discuss the personalized query rewriting evaluation results on Human-Annotated test set for our proposed CGF showed in Table 3. We use the same model as the global query rewiring task. The only difference is that the constrained decoding space is changed to personalized-level based on each user's historical interactions with agent and thus diverse for each user. As seen, the results of both search-based and our CGF models are in the relatively high range, as the search space for personalized query rewriting is much smaller comparing to global-wise. However, the CGF can still outperforms search-based system by 1.66% absolute precision score and 1.53% trigger rate meanwhile.

Dialog	Reference	CGF	CGF w/o CD	CGF w/o CE
USER: what time is sunset tonight AGENT: sunset, in greenacres, florida, on thursday, october 21 will be 6:48pm USER: what kind of sunset tonight in willimantic connecticut	what time is sunset tonight in willimantic connecticut	what time is sunset tonight in willimantic connecticut	what time is sunset tonight in willimantic connecticut	what is sunset tonight in willimantic connecticut
USER: play little yancy AGENT: Lil' Fancy from Apple Music.y USER: play little yankees praise part	play little yancy praise party	play little yancy praise party	play little yancy praise party	play little yankees praise party music
USER: play in jesus name by katie nicole	play in jesus name by katy nichole	play in jesus name by katy nichole	play in jesus name by kayla nicole	play in jesus name by katy nichole

Table 4: Examples of the generated rewrites. In the dialog session, the last turn from the user is the current request which is needed to be rewritten by the model. "CGF w/o CD" denotes the model CGF without constrained decoding, "CGF w/o CE" denotes the CGF without context-enhanced encoding.



(a) Precision (b) Trigger Rate

Figure 3: Global query rewriting evaluation on the first turn and not first turn subsets. "CGF w/o context" denotes the CGF without context-enhanced encoding.

4.7 Effect of Context-enhanced Encoding

We study the effect of the context-enhanced encoding in this subsection. As in some of test cases, there are not previous context available and the query the model will rewrite the first turn of the multi-turn dialogue session. We investigate if the proposed model is robust and effective for both of such cases. Thus, we split the global test set into the with previous context ("First turn") and without previous context cases ("Not first turn").

As shown in the Figure 3, the CGF gets significant improvement for both precision and trigger rate on the "Not first turn" test set comparing to CGF without context-enhanced encoding, which demonstrates the effect of the context information during the model training. Moreover, on "First turn" test set, surprisingly, when there is not previous context for the CGF model, the performance only decrease in a relatively small margin. This suggests that the model is good at generalization and robust for various test cases in the actual scenario.

4.8 Case Study

We here discuss the several cases that indicate the representative situations we find so that we can further understand the effect of the context-enhanced encoding and constrained decoding in CGF. As shown in the Table 4, the first example illustrates the cases when CGF w/o context-enhanced encoding gives a rewrite which changes the semantic meanings of the source request ("what kind of" -> "what is") and is not faithful. However, with consideration of previous context information, the CGF is able to understand the user intent and provide the accurate rewrite. Also, the second case corresponds to the situation of carrying over an correct entity from context and replace the wrong entity in current utterance, while as shown in the table, this is not hard for our context-enhanced encoding models. However, without consider the information from context, the model sometimes fails. The third case shows that without constrained decoding, the CGF has a factual inconsistency generation ("kayla nicole" is an artist but never sung "in jesus name"). This is a common situation for generation-based models, especially on unseen data samples. Conversely, this situation rarely happens with constrained decoding, as the generation is based on the predefined constrained decoding space and we will never have such factual inconsistency generation.

5 Online Experiments

To investigate the effectiveness of the introduced techniques in the real-world large-scale conversational AI agent, we leverage the proposed model CGF to generate the global rewrites and deploy them into the online production environment. We compare it with the no CGF rewrites within English speaking users environment. The data was collected for more than one week over a significant percentage of traffic via the A/B testing framework. We use one primary metric to evaluate the performance of our proposed CGF approach during A/B:

- Defect Rate: the total number of potential rewritten utterances that is defective divided by total number of potential rewritten utterances. We leverage the defect detection model

Original request (w/o context): USER: how much time is left on my **CGF rewrite:** how much time is left on my timer	**Response (before rewrite):** Sorry, I missed something. Can you say it again? **Response (after rewrite):** You have 2 minutes and 20 seconds left on your 7-minute timer.
Original request (w/o context): USER: play night talk by drake **CGF rewrite:** play knife talk by drake	**Response (before rewrite):** I couldn't find night talk **Response (after rewrite):** Here's Knife Talk , by Drake (feat. 21 Savage, Project Pat) , on Amazon Music.
Original request (w/ context): USER: play little yancy AGENT: Lil' Fancy from Apple Music.y USER: play little yankees praise part **CGF rewrite:** play little yancy praise party	**Response (before rewrite):** Sorry, I'm having trouble finding the song. **Response (after rewrite):** Here is little yancy praise party.

Table 5: Examples from the online experiment.

proposed by Gupta et al. (2021) to measure if an utterance is defective.

From A/B results, our proposed model contributed to a healthy decease of defect rate without compromising on the traffic volume of generated rewrites. We observed significant [2] reduction of defect rate 20.94% and millions of new rewrites generated by the proposed approach.

As a part of online monitoring, we monitor the number of new, previous unseen rewrites in online live traffic. The Table 5 shows the cases where the original request got unsatisfied response from the agent and after the rewrite, the friction was removed with satisfied response. In the first example, the original request "*how much time is left on my*" lacks the key entity "*timer*" due to user's accent or background noise. Even without context information available i.e. the request is the first turn, the propose CGF can successfully rewrite it and thus the user got the satisfied response from the agent finally. In the second case, due to an ASR error, the original request cannot well interpreted by agent and the agent cannot fulfill the user's need. The CGF can correct the ASR error from "*night*" to "*knife*" so as to meet the need of the user. The last example demonstrates that when user rephrase his/her request, CGF generated the good rewrite "*play little yancy praise party* " based on the helpful context information.

6 Conclusion

In this work, we propose CGF, a novel paradigm to conduct query rewriting: generate target rewrite autoregressively with context-enhanced encoding and constrained decoding. Our proposed CGF is a general framework for different query rewriting purposes where one can freely define the decoding space, (e.g. global, personalized or domain-specific space). Both offline and online experiments show that our approach consistently and significantly improves query rewriting performance and demonstrate the effectiveness and universality of the CGF.

Future directions include exploring advanced prompt approaches that can better reflect the learning abilities of large-scale language models, as well as extending the current CGF to handle the other tasks like entity correction, etc.

[2] p-value<0.0001

References

Zheng Chen, Xing Fan, and Yuan Ling. 2020. Pre-training for query rewriting in a spoken language understanding system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7969–7973. IEEE.

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. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

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)*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *ACL*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *ACL*.

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*.

Akash Kumar Mohankumar, Nikit Begwani, and Amit Singh. 2021. Diversity driven query rewriting in search advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3423–3431.

Pragaash Ponnusamy, Alireza Roshan Ghias, Chenlei Guo, and Ruhi Sarikaya. 2019. Feedback-based self-learning in large-scale conversational ai agents. *arXiv preprint arXiv:1911.02557*.

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Lambert Mathias. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In *NAACL*.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question rewriting for conversational question answering. *arXiv preprint arXiv:2004.14652*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NIPS*.

Liang Wang, Wei Zhao, Ruoyu Jia, Sujian Li, and Jingming Liu. 2019. Denoising based sequence-to-sequence pre-training for text generation. *arXiv preprint arXiv:1908.08206*.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *ACL*.

Xianchao Wu, Ander Martinez, and Momo Klyen. 2018. Dialog generation using multi-turn reasoning neural networks. In *NAACL*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *NAACL*.