Learning to Retrieve Engaging Follow-Up Queries

Christopher Richardson [‡*] **Sudipta Kar** [†] **Anjishnu Kumar** [†] **Anand Ramachandran** [†]
Omar Zia Khan [†] **Zeynab Raeesy** [†] **Abhinav Sethy** [†]
[‡] Georgia Institute of Technology
[†] Amazon Alexa AI
crichardson332@gmail.com
{sudipkar,anjikum,anramac,ozkhan,raeesyzr,sethya}@amazon.com

Abstract

Open domain conversational agents can answer a broad range of targeted queries. However, the sequential nature of interaction with these systems makes knowledge exploration a lengthy task which burdens the user with asking a chain of well phrased questions. In this paper, we present a retrieval based system and associated dataset for predicting the next questions that the user might have. Such a system can proactively assist users in knowledge exploration leading to a more engaging dialog. The retrieval system is trained on a dataset called the Follow-up Query Bank (FQ-Bank). FQ-Bank contains ≈14K multi-turn information-seeking conversations with a valid follow-up question and a set of invalid candidates. The invalid candidates are generated to simulate various syntactic and semantic confounders such as paraphrases, partial entity match, irrelevant entity, and ASR errors. We use confounder specific techniques to simulate these negative examples on the OR-QuAC dataset. Then, we train ranking models on FQ-Bank and present results comparing supervised and unsupervised approaches. The results suggest that we can retrieve the valid follow-ups by ranking them in higher positions compared to confounders, but further knowledge grounding can improve ranking performance. FQ-Bank is publicly available at https://github.com/amazon-science/fq-bank.

1 Introduction

State of the art open domain conversational voice assistants can help users accomplish a wide range of tasks, including: factoid question answering, playing music, adding items to personal lists, controlling smart home appliances, and booking transportation. However, the linear nature of dialog with existing voice assistant technology makes it challenging for users to discover and fully utilize the full range of these capabilities. In addition, successful utilization often requires exact formulation of the request, which further hinders the experience. One recent approach to addressing these issues in the voice assistant domain involves predicting relevant follow-up queries in order to assist the user with accomplishing their latent goals.[1]

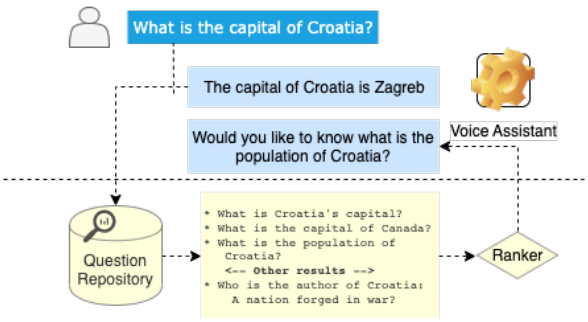

Figure 1: An overview of the follow-up question (FQ) retrieval system.

Relevant follow-up queries (FQs) for typical voice assistant scenarios can range from specific command and control tasks such as "*What is the temperature in New York*" followed by "*What is the chance of rain in New York?*", to more open-ended knowledge exploration, e.g. "*What is the capital of Croatia*" followed by *"What is the population of Croatia?"*. Once a valid FQ has been identified, the system can proactively recommend it to reduce user's cognitive load, e.g. "*Would you like to know what the population of Croatia is?*". This exchange is illustrated in Figure 1. The user can then be engaged in follow-up dialogue without the need to ask redundant questions.

Follow-up queries can be identified by retrieving and ranking candidates from a question repository. In this approach, we are given dialog of one or

[*] Work done during internship.

[1] https://www.amazon.science/blog/alexa-gets-better-at-predicting-customers-goals

Dialog History
Where was Kurt Gödel born?
Brunn, Austria-Hungary
When was Kurt Gödel born?
April 28, 1906
What was Kurt Gödel's home life like?
ethnic German family
Where did Kurt Gödel go to school?
Godel attended the Evangelische Volksschule in Brunn
Valid Follow up
What were Kurt Gödel's interests?
Generated Negative Examples with Types
Where was Kurt Gödel born?
Duplicate of dialog history
Which school did Kurt Gödel attend?
Paraphrase
Where was Cristiano Ronaldo born?
Irrelevant Entity
Where did Curt Gödel go to school?
ASR Error
When did Cristiano Ronaldo join Juventus?
Random Question
When did Kurt Gödel join Juventus?
Irrelevant context

Table 1: An example showing the dialog history, current turn, valid utterance, and a set of negative utterance candidates from the generated dataset.

more turns between a user and a voice assistant. The system first uses a search engine to retrieve a set of relevant questions by searching against a question repository comprising of historical queries and questions generated from a knowledge base. To create questions from a knowledge base, we can use templates or use a few-shot Natural Language Generation (NLG) model such as T5 (Raffel et al., 2020). For example, we can take the tuple `{entity, place of birth}` and construct a question template like "*what is the birthplace of {entity}*".

Through a preliminary study of this retrieval approach, we found a basic lexical similarity-based search engine to be ineffective and often returns invalid follow-up queries. Often these top search results included paraphrases of the original query (*When was Cristiano Ronaldo born* → *What year was Cristiano Ronaldo born*), as well as similar questions for unrelated entities (*When was Cristiano Ronaldo born* → *When was Christian Bale born*). Therefore, an additional ranking module is needed in order to re-rank the search results based on their quality as follow-up queries. To the best of our knowledge, there exists no dataset focused on information-seeking follow-up queries, given a dialog context and a set of valid and invalid follow-up candidates. This problem differs from traditional recommendation systems in that 1) a voice assistant can only recommend one follow-up at a time, and 2) the follow-up query must be highly precise, contextually relevant, and coherent to ensure a positive user experience. This technique can be extended beyond the domain of virtual assistants, for example to chatbots, search engines, and any other smart interaction scenario where contextual coherence and precision is necessary. Therefore, in this paper, we created the FQ-Bank dataset addressing this problem and explored different modeling techniques to develop a ranking model to retrieve relevant follow-up queries (FQ). The main contributions of this paper can be summarized as follows:

1. For the scenario of a retrieval-based follow-up question selection, we identify a typology of confounders based on preliminary results from a search engine.

2. We propose techniques to synthetically generate confounders according to this typology, based on the publicly available conversation dataset OR-QuAC (Qu et al., 2020), and created the Follow-up Query Bank (FQ-Bank) dataset. FQ-Bank is publicly available and can be used to develop and test machine learning systems for identifying contextually relevant and meaningful follow-up queries from search results. Additionally, the confounder creation techniques can be applied in data augmentation for similar problems. Table 1 shows an example from the generated dataset.

3. We adopt a pre-trained language model based approach to develop a benchmark model for ranking a set of candidate follow-up queries for a given factoid utterance and dialog history. We explore the effectiveness of this technique and identify gaps and future directions.

2 Related Works

Previous studies on proactivity in conversational AI mostly focuses on response generation. Follow-up question identification and generation approaches have been explored from different perspectives. For example, Kundu et al. (2020) explored the task of identifying if the latest user utterance is a follow-up of the previous questions or it has a different new

context. This is helpful for understanding the question context properly and give the correct response. They also derived a new dataset called LIF from the QuAC dataset (Choi et al., 2018), where each data point contains a conversation history, a new utterance, a passage used to answer the previous questions, one valid follow-up, and one or two invalid follow-ups. However, this dataset is focused on passage-based question answering, and the confounder typology does not address issues found in the search engine based FQ retrieval scenario (e.g. paraphrases, irrelevant entity substitution, etc).

Other works have focused on generating follow-up queries for extracting information from users. For example, Ge et al. (2022) proposed a knowledge-driven system for generating follow-up queries, but it targeted the generation of follow-up survey questions to extract information from humans. Su et al. (2018) and B et al. (2020) explored systems for asking follow-up queries to interview candidates to extract more relevant information.

Our proposed method is focused not on information extraction from users, but rather providing highly relevant additional information to the user.

3 Follow-up Query Bank

For our initial study on identifying FQs, we created a search index of information-seeking questions regarding public facts, spoken by users of a commercial voice assistant. We then queried the search engine with different types of information-seeking questions and analyzed the top negative (i.e., not a suitable follow-up) search results and categorized them into a typology of confounders. Using this typology, we set out the task of simulating a similar scenario on a public conversation dataset and did not use any voice assistant data anymore. We selected OR-QuAC as the seed dataset as it provides multi-turn information-seeking dialogs on a particular topic. In the rest of this section, we will provide a brief overview of the confounders and OR-QuAC dataset, followed by an overview of the simulation methodology of the confounders using OR-QuAC.

3.1 Confounder Identification

We created a search index with an open-source search engine for a set of de-identified user interactions with a commercial voice assistant during a period of time. Then, we carefully selected a set of questions that are different from each other in aspects such as the intent, entity in context, entity's gender and topic domain. We searched the index against each of these questions and inspected the relevance of the top 20 search results as a follow-up question. We found that most of ($\approx 95\%$) the top search results are not suitable candidates for a follow-up. We analyzed the top irrelevant results and categorized them as the following confounders that should rank low in their relevance as follow-ups.

- **Paraphrase** We observed a large segment of irrelevant candidates that are semantic equivalents of the query question. This happens because people can ask the same question in different ways. For example, "*How old is Joe Biden*" and "*Joe Biden age*" are lexically different but semantically equivalent.

- **Irrelevant Entities** Often, the top search results are about entities different from the query question, but the questions have a similar carrier phrase. For example, "*What team does Ronaldo play for*" retrieves questions like "*What team does Tom Brady play for*". It is true that some user may find such questions as relevant follow-ups, but this is highly subjective. Tom Brady will be completely irrelevant to a user who does not follow the National Football League (NFL). As a result, we considered that such scenarios are irrelevant for now.

- **Partial Entity Match** This is a variation of the previous confounder. Here, not only is the carrier phrase similar, but also, the entities share one or more tokens. For example, "*How old is the University of Washington*" can retrieve questions like "*How old is the University of Houston*". Here we observe partial entity match "*university of*" in addition to the identical carrier phrase "*how old is the*".

- **Irrelevant Context** Some top search results share the correct entity with the query question, but the retrieval can be a non-sequitur. For example, "*What is the capital of France*" can retrieve questions like "*Where is France*". Even though contextually it is a relevant questions, asking back "*Would you like to know where is France*" does not make a good experience for the user as the chances are high that

	Train	Validation	Test
Dialog	13,480	1,445	2,132
Turns (utterance, response pair)	52,712	5,534	8,195
Turns per dialog	3.91	3.86	3.84
Tokens per utterance	10.39	10.15	10.12
Tokens per response	16.76	17.00	16.90
Confounders			
Paraphrase	30,707	3,033	4,676
- *per dialog*	2.28	2.11	2.19
Irrelevant entity	91,490	9,660	13,736
- *per dialog*	6.79	6.74	6.44
Irrelevant context	51,342	5,479	8,153
- *per dialog*	3.81	3.82	3.82
ASR Error	85,136	8,906	11,007
- *per dialog*	6.32	6.21	5.16
Random utterance	40,440	4,302	6,396
- *per dialog*	3	3	3
Duplication of dialog history	52,712	5,534	8,195
- *per dialog*	3.91	3.86	3.84
Total	3,51,827	36,914	52,163
- *per dialog*	26.10	25.74	24.47

Table 2: Statistics of the created dataset with each category of the negative examples.

the user already have an idea on the geographical position of France.

Additionally, we listed the following confounders that were not seen in our limited data analysis but can appear in a larger system:

- **Automatic Speech Recognition (ASR) Error** ASR failures can sometimes replace an entity with a similar sounding word. For example, *Kurt* can be replaced with *Curt* in "*Where did Kurt Gödel go to school*". High lexical overlap can rank such irrelevant entities highly.

- **Duplication of Dialog History** Sometimes the information provided by a candidate follow-up question can already be present in a multi-turn dialog history. In such a case, it is important to identify and get rid of those questions by modeling the dialog history.

After identifying these confounder categories, we selected OR-QuAC as the starting dataset, and used different techniques to generate these confounders and simulate the retrieval scenario for the follow-up selection system.

3.2 OR-QuAC Dataset

Open-Retrieval Conversational Question Answering (OR-QuAC) consists of ≈6K multi-turn information-seeking dialogues between two humans, one posing as student (asks knowledge-seeking questions) and the other as teacher (answers the questions using Wikipedia as the knowledge source). It draws from the popular QA dataset QuAC (Choi et al., 2018) as well as CANARD (Ghoneim and Peskov, 2019), which provides context-independent rewrites of initial questions written by human annotators.

This dataset is well-suited for our purposes as: i) the conversations aim at exploring knowledge about entities or topics, ii) multi-turn conversations enable us simulating a dialog history (one or more question-answering turns between two people), iii) query rewrites are helpful to get rid of anaphoric references which can make candidate questions ambiguous about entities (e.g., "*How many kids Kamala Harris has*" removes the ambiguity from "*How many kids she has*").

For each information-seeking question in the OR-QuAC dataset, we chose the rewritten version as the current question, the previous turns as the dialog history, and the immediate next turn as the valid follow-up question. Then, we used different techniques to generate the confounders that we will explain in the next section.

3.3 Data Sample Generation

For a conversation in the OR-QuAC dataset of T turns (question−answer pairs), we have sampled $T-1$ data points $\{x, y\}$. Each generated data sample contains a dialogue context, x, of length $\mathcal{L}$ ($1 \leq \mathcal{L} \leq T-1$). x contains a dialog history $\{(q_1, a_1), ..., (q_{\mathcal{L}-1}, a_{\mathcal{L}-1})\}$ of length $\mathcal{L}-1$ (i.e., $\mathcal{L}-1$ question (q)−answer (a) pairs), and a current question ($q_{\mathcal{L}}$) and the answer ($a_{\mathcal{L}}$). Hence, $x = \{(q_1, a_1), ..., (q_{\mathcal{L}-1}, a_{\mathcal{L}-1}), (q_{\mathcal{L}}, a_{\mathcal{L}})\}$.

Each data sample also contains a set of positive and negative follow-up queries, $y = \{y^+\} \cup y^-$. y^+ is a single positive follow-up question, and y^- is a set of negative follow-ups ($y^- = \{y_1^-, y_2^-, ...\}$) that we have created based on the identified confounders.

Valid Examples Given a dialog history $x_{1:T-1}$ of length $T-1$ and a current turn x_T, we consider the consecutive question (x_{T+1}) in the OR-QuAC dataset as a positive follow-up question.

Adversarial Examples We used the following methods to populate the candidate question space for a turn with negative examples based on the confounders we have listed in Section 3.1:

- **Paraphrase:** We used a pre-trained BART model (Lewis et al., 2019) that was fine-tuned on several paraphrase datasets[2]. For the last user turn in a dialog history, we used this model generated paraphrase as a confounder.
- **Irrelevant Entities and Partial Entity Match:** We first used the SpaCy[3] library to identify the named entities in the current question in a turn. Then we generate a negative example by replacing the entity with an entity of a similar type from a catalog generated from WikiData. For entities with multiple word tokens, we replace a token (e.g., first name or last name) with a random first name or last name token. For a dialog, we created multiple such examples.
- **Irrelevant Context:** We randomly sampled one question from the rest of the dataset that has a similar entity type and replace that with the entity in the context of a current question. That means, for an entity we swap the original question with a random one.
- **Random question:** We added three random questions from the dataset as a negative examples for a dialog.
- **ASR Error:** For an entity in a question, we generated a similar sounding entity using the Datamuse API [4] and replaced the original entity with the generated homophone. For entities with multiple word tokens, we created multiple examples like this by replacing one token with a homophone at a time.
- **Duplication of Dialog History:** We added a question from the dialog history in the candidate set.

We maintained the standard training, validation, and test splits from OR-QuAC while generating the dataset. Table 1 shows an example of generated data and Table 2 shows statistics of the dataset. As we maintained the original data split, distribution is similar across all the splits. For each dialog, there are ≈25 negative examples with one positive example. That means, a model needs to learn contextual relevancy for being able to identify the correct follow-up.

[2] https://huggingface.co/eugenesiow/bart-paraphrase

[3] http://spacy.io

[4] https://www.datamuse.com/api

4 Learning to Identify Relevant Follow-up

Task Formulation: Given a dialog $x = \{(q_1, a_1), ..., (q_{\mathcal{L}-1}, a_{\mathcal{L}-1}), (q_{\mathcal{L}}, a_{\mathcal{L}})\}$ of $\mathcal{L}$ turns and a randomly organized set of n candidate follow-up queries $y = \{y^+\} \cup \{y_1^-, y_2^-, ..., y_{n-1}^-\}$, the task is to model $P(i|x), i \in y$, such that $\text{argmax}_i\, P(i|x) = y^+$.

Here, q is a question, a is an answer, y^+ is a positive follow-up example, and y^- is a set of negative examples. In order to develop a follow-up question candidate ranker, we experiment with different unsupervised and supervised approaches as described below.

4.1 Unsupervised

We experimented with Glove (Pennington et al., 2014) word embeddings, pre-trained Sentence-BERT (Reimers and Gurevych, 2019) model in the unsupervised direction. For a given dialog x and candidate utterance set $y = \{y^+, y^-\}$, we use the Glove or SentenceBERT to generate a high-level vector representation $\bar{x}$ from the dialog and do the same for each of the candidate utterances $y_i \in y$. With Glove, we compute the mean of the $300d$ embedding vectors for all the word tokens in a dialog x and represent the out-of-vocabulary (OOV) words with zero vectors. The vocabulary coverage of Glove is ≈99% for the dataset. For SentenceBERT, we feed the entire input texts (concatenation of multiple turns in x) for x and $y_i \in y$ to generate $\bar{x}$ and $\overline{y_i}$, respectively. Then, we compute the cosine similarity $\alpha = \cos(\bar{x}, \overline{y_i})$ between $\bar{x}$ and $y_i \in y$ and rearrange y in descending order based on α.

4.2 Supervised

For the supervised experiments, we fine-tune a pre-trained language model by translating the problem as a binary classification task. In other words, for a given dialog $x = \{(q_1, a_1), ..., (q_{\mathcal{L}-1}, a_{\mathcal{L}-1}), (q_T, a_T)\}$ and a candidate set $y = \{y^+\} \cup y^-$, we train a model θ to predict $\hat{y} = P(i \mid x), i \in y$, where $\hat{y} \rightarrow \mathbb{R} : [0, 1]$.

We format the input by concatenating the dialog history turns and a candidate utterance with a `[SEP]` token and a single output node outputs a continuous value between 0 and 1. As the starter pre-trained language model we experiment with BERT (Devlin et al., 2019) and RoBERTa (Liu

	Validation	Test
Unsupervised		
Glove	0.142	0.141
SentenceBERT	0.133	0.141
Supervised		
BERT	0.842	0.805
RoBERTa	0.838	0.808
Hit Ratio@1/ Hit Ratio@3		
BERT	72.0/ 89.3	68.5/ 88.1
RoBERTa	71.7/ 88.7	68.2/ 89.5

Table 3: Ranking performance in MRR for different unsupervised and supervised methods. The last two rows show the Hit Ratio at the first and third position for BERT and RoBERTa.

et al., 2020). We use the bert-base-cased[5] and roberta-base[6] variations of these models. We fine-tune the models for 20 epochs with an early stopping patience of three epochs with a learning rate of $2e^{-5}$ and batch size of 64. We use cross-entropy loss to optimize the model with AdamW optimizer. During inference, we use the model predicted score to rearrange the candidate set for a dialog in descending order.

5 Experiments and Results

Evaluation Metric: As the task is to rank the valid follow-up question higher than a set of invalid confounders, we evaluate the performance using Mean Reciprocal Rank (MRR), given as:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i} \quad (1)$$

where rank_i is the rank position of the valid candidate for the ith datapoint.

Additionally, we compute Hit Ratio@1 and Hit Ratio@3 for the top performing methods to analyze the percentage of samples for which the ranking method ranked the correct candidate as the first item and within the first three items.

Quantitative Results: In Table 3 we report results comparing the methods proposed in Section 4 on the adversarial dataset described in Section 3. The unsupervised methods performed poorly for the ranking task resulting into MRR scores of 0.141 for both Glove and SentenceBERT based embeddings, and the trend is similar for all the data splits. This is not surprising as cosine similarity is expected to be high for paraphrases. As discussed in section 3.1, paraphrases are not good candidates for FQs as they don't provide any value to the user.

We observe a large improvement when we fine-tune pre-trained language models like BERT and RoBERTa to simply classify each candidates as relevant or irrelevant. The MRR is ≈ 0.8 when we treat the models' confidence score for relevancy as the basis for ranking the candidates.

The Hit Ratio@1 and Hit Ratio@3 metrics show that both BERT and RoBERTa ranked the correct FQ at the first rank for ≈68% cases. Both the models ranked the correct FQ within the first three items for ≈88-89% cases. This shows promise in using such methods to retrieve a relevant follow-up

Dialog context

- Where was Michael Bennett born?
- Who are Michael Bennett's parents?
- When did Michael Bennett's career begin?
- What show did Michael Bennett begin his career?

Irrelevant Context Candidate
When did Michael Bennett move to Alaska? (Model score: 0.3)
Valid Candidate
What was Michael Bennett's role in the "Here's Love" and "Bajour"? (Model score: 0.2)

Dialog context

- What happened to Sachin Tendulkar during the tour of Australia?
- How did Sachin Tendulkar do in the 2003 Tour of Australia?
- How many games did Sachin Tendulkar win during 2003?
- Did Sachin Tendulkar win any awards?

Irrelevant Context Candidate
How many hits did Sachin Tendulkar have? (Model score: 0.21)
Valid Candidate
Was there any controversies for Sachin Tendulkar? (Model score: 0.35)

Table 4: Examples where the model predicted scores do not match with the category of the follow-ups.

[5] https://huggingface.co/bert-base-cased
[6] https://huggingface.co/roberta-base

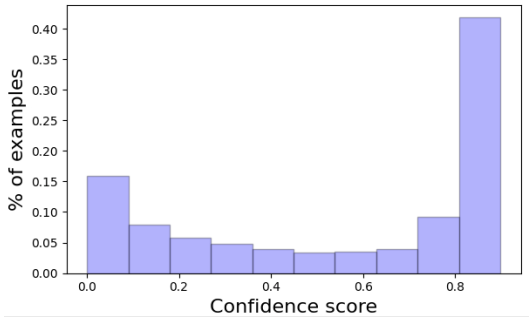

Figure 2: Histograms of model predictions for valid follow-up queries.

question.

Error Analysis Analysing the fine-tuned models' predicted scores for different confounder types, we have found that the models can identify most of the confounder types easily. For example, the model predicted score is below 0.1 for $\approx 99\%$ candidates from duplication of dialog history, ASR errors, and random utterance confounder. However, the models often predicted higher scores for the candidates from the irrelevant context category (score is < 0.1 for 83% candidates). For 8% of these candidates, the score is higher than 0.4, which is not the case with other categories of confounders.

Inspecting some examples like the ones presented in Table 4, we found that without having factual information about the entities or topics, such irrelevant contexts are often difficult to identify for humans as well. They can look linguistically plausible but have factual errors. For example, the question "*How many hits did Sachin Tendulkar have*" sounds plausible to some humans, but is actually invalid. Tendulkar is a cricket player, and 'hits' is not a statistic in cricket. However, it is a real statistic in baseball, so specific domain knowledge is needed to rule this out as a valid FQ. This example illustrates how integrating information about entities from knowledge bases can be helpful for the system, ans this method can be explored in the future. Although scores like 0.2 do not look very high in general for a scale of 0 to 1, Figure 2 shows that the model assigns such scores to a large portion of valid follow-up queries.

Observing the model's overall performance (MRR of $\approx$0.8) in ranking the valid candidates at a better position than the invalid ones shows promise in using such a system can be a good starting point for developing a follow-up question retrieval system. A large advantage of the proposed adversarial example generation methods and the proposed dataset is that these can help to bypass the need for exhaustive data annotation need for developing a follow-up generation system. Additionally, the trained model using this dataset can be further fine-tuned by annotating a small number of case specific examples, which would help to improve the model accuracy and adapt in different use cases, as well as reach a higher accuracy in identifying suitable follow-up queries.

6 Conclusions

In this paper, we sought to address the problem of identifying valid and engaging follow-up queries for a user interacting with a conversational assistant. We experimented with a retrieval and ranking based framework to achieve this using a search engine and a database of past queries. In doing so, we identified a typology of confounders returned by the search. In order to train a ranking model to identify valid follow-up queries, we synthetically generated confounders based on a publicly available conversation dataset. We showed that our approach of ranking retrieved candidates based on their validity as follow-up queries achieved reasonable performance, but also that integrating external knowledge on entities or topics could improve follow-up selection. We have made the dataset publicly available to enable further research in this direction.

7 Limitations

The first limitation of this work is that we are attempting to mimic conversational interactions with publicly available human-annotated data based on Wikipedia. Thus in some cases the generated dataset can contain dialogues unrealistic to the voice assistant scenario. Additionally, despite our typology of confounders being based on results from a search-based approach using real data, there are inevitably additional types of potential confounders not fully covered by our approach.

Second, we only focused on contextual relevance and coherence through the lens of language. But, in practice, there are external factors like user preference, time of the day, repetition in a longer period (e.g., a user may have asked the question in the follow-up a couple of days ago and it does not make any sense to ask the same question as a follow-up). More comprehensive methods would be needed to address these concerns.

Finally, this dataset is limited to knowledge-seeking queries. Other types of valid follow-up actions (e.g. setting a timer, booking a ride) are not included in this dataset.

References

Pooja Rao S B, Manish Agnihotri, and Dinesh Babu Jayagopi. 2020. Automatic follow-up question generation for asynchronous interviews. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 10–20, Santiago de Compostela, Spain. Association for Computational Lingustics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yubin Ge, Ziang Xiao, Jana Diesner, Heng Ji, Karrie Karahalios, and Hari Sundaram. 2022. What should i ask: A knowledge-driven approach for follow-up questions generation in conversational surveys. *arXiv preprint arXiv:2205.10977*.

Ahmed Elgohary Ghoneim and Denis Peskov. 2019. Canard: A dataset for question-in-context rewriting.

Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. Learning to identify follow-up questions in conversational question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 959–968, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. RoBERTa: A robustly optimized BERT pretraining approach.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 539–548.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Huai-Hung Huang. 2018. Follow-up question generation using pattern-based seq2seq with a small corpus for interview coaching. In *INTERSPEECH*.