Consistency Training by Synthetic Question Generation for Conversational Question Answering

Hamed Hematian Hemati and **Hamid Beigy**
AI Group, Computer Engineering Department, Sharif University of Technology
hamedhematian@ce.sharif.edu, beigy@sharif.edu

Abstract

Efficiently modeling historical information is a critical component in addressing user queries within a conversational question-answering (QA) context, as historical context plays a vital role in clarifying the user's questions. However, irrelevant history induces noise in the reasoning process, especially for those questions with a considerable historical context. In our novel model-agnostic approach, referred to as **CoTaH** (**Co**nsistency-**T**rained **a**ugmented **H**istory), we augment the historical information with synthetic questions and subsequently employ consistency training to train a model that utilizes both real and augmented historical data to implicitly make the reasoning robust to irrelevant history. To the best of our knowledge, this is the first instance of research using synthetic question generation as a form of data augmentation to model conversational QA settings. By citing a common modeling error prevalent in previous research, we introduce a new baseline and compare our model's performance against it, demonstrating an improvement in results, particularly in later turns of the conversation, when dealing with questions that include a large historical context.

1 Introduction

Humans often seek data through an information-seeking process in which users engage in multiple interactions with machines to acquire information about a particular concept. A prominent example of this phenomenon is the introduction of ChatGPT (OpenAI, 2023). Conversational Question-Answering (CQA) systems address user questions within the context of information-seeking interactions. In CQA, unlike conventional question answering, questions are interconnected, relying on previous questions and their corresponding answers (history) to be fully understood without ambiguities. Qiu et al. (2021) showed that filtering irrelevant history can boost the model's accuracy. However, it utilizes the gold answers of history instead of the predicted ones, like many previous methods. This setting deviates from the real-world scenario, where models have to rely on their own predictions for previous questions to answer the current question. Our work aligns with the framework of addressing irrelevant history. However, unlike Qiu et al. (2021), our method abstains from utilizing the gold answers of history. Moreover, unlike Qiu et al. (2021), which requires an iterative process to select relevant history, we utilize only one transformer (Vaswani et al., 2017) during prediction, resulting in reduced time and memory. We augment the history of questions in the training set with synthetic questions. Our underlying idea is to maintain the model's consistency in its reasoning, whether utilizing the original historical data or the augmented version. Baselines like BERT-HAE (Qu et al., 2019a), HAM (Qu et al., 2019b), and GraphFlow (Chen et al., 2020) leverage the gold answers of history in their modeling. Siblini et al. (2021) conducted a re-implementation of BERT-HAE and HAM, and Li et al. (2022) conducted a re-implementation of HAM and GraphFlow using predicted history answers, which resulted in a significant performance decrease. As a result, in this paper, we employ the base transformer of our method as the baseline, as its performance surpasses the re-implementation of the mentioned methods. Our method results in a 1.8% upgrade in overall F1 score compared to this baseline, causing a significant improvement in the scores of questions in the later turns (questions with large historical context). Furthermore, our method introduces a substantial improvement in detecting unanswerable questions compared to the introduced baseline.

2 Related Works

The task of CQA has been introduced to extend question answering to a conversational setting.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 630–639

CoQA (Reddy et al., 2019) and QuAC (Choi et al., 2018) have been proposed as two extractive datasets in the CQA task. BERT-HAE (Qu et al., 2019a) employs a manually defined embedding layer to annotate tokens from previous answers within the document, and Qu et al. (2019b) extends this approach by introducing an ordering to these annotations. GraphFlow (Chen et al., 2020) utilizes a graph made out of document tokens to tackle the problem. FlowQA (Huang et al., 2019) utilizes multiple blocks of Flow and Context Integration to facilitate the transfer of information between the context, the question, and the history. ExCorD (Kim et al., 2021) uses consistency regularization (Laine and Aila, 2017; Xie et al., 2020) to regularize the training by leveraging re-written questions. Qiu et al. (2021) introduces the idea of irrelevant history and its effect on degrading performance, proposing a policy network to select the relevant history before reasoning. However, the mentioned models employ the gold answers from history in their modeling. This approach deviates from real-world scenarios, where systems should rely on their previous predictions to answer current questions (Siblini et al., 2021). Siblini et al. (2021) re-implements BERT-HAE and HAM, and Li et al. (2022) re-implements HAM, GraphFlow, and ExCorD using the model's predictions, reporting a sharp decrease in performance. FlowQA experiences a performance drop from 64.6% to 59.0% on the development set when gold answers in history are not used (Huang et al., 2019).

3 Problem Definition

To model a CQA setting, at dialog turn k, a model receives a question (q_k), a document containing the answer (D), and the history of the question (H_k), which is represented as a set of tuples, such as $H_k = \{(q_0, a_0^{pred}), \cdots, (q_{k-1}, a_{k-1}^{pred})\}$, where a_j^{pred} is the model's prediction for q_j. It's important to note that the model may utilize only some of this information. For instance, we only employ history questions while excluding history answers. The objective is to predict the answer a_k^{pred} for q_k.

$$a_k^{pred} = \arg\max_{a_k} P(a_k|q_k, H_k, D) \quad (1)$$

4 Methodology

We seek to make the reasoning robust to irrelevant history implicitly by augmenting the dataset. To this end, for question q_k, we augment its history by injecting some synthetic questions. Let $H_k^\star$ be the augmented history. The intuition is that irrespective of whether the reasoning is performed with H_k or $H_k^\star$, the result should be the same. In other words:

$$P(a_k|q_k, H_k, D) = P(a_k|q_k, H_k^\star, D) \quad (2)$$

To achieve this goal, we establish a two-stage pipeline. Our pipeline consists of a history augmentation module, whose goal is to augment the history and a question-answering module, whose objective is to consistently train a QA network so that the reasoning is consistent. The overall architecture of our model is depicted in Figure 1.

4.1 History Augmentation Module

This module includes a conversational question generator, denoted as CQG_θ, where θ represents the parameter set of the generator, and a question selector, denoted as QS, which is responsible for choosing a set of S synthetic questions generated to augment the history.

Training The first step involves training CQG_θ. While there has been research aimed at generating conversational questions (Gu et al., 2021; Pan et al., 2019), for the sake of simplifying the implementation, we employ a straightforward generative transformer for this task. To train this network, we input D, H_k, and a_k into the network, intending to generate q_k. We train this network using cross-entropy loss in an auto-regressive manner.

Question Generation After training CQG_θ, we aim to generate synthetic conversational questions for the training set. Suppose that we want to generate synthetic conversational questions for q_k. We iteratively generate synthetic questions between q_j and q_{j+1} for $1 \leq j \leq k-1$. Suppose that a_j is located in the i-th sentence of the document. We extract noun phrases from sentences $i-1$, i, and $i+1$ as potential answers. We make this choice because we want these answers to be similar to the flow of conversation, and if these answers are extracted from local regions, the likelihood increases. Let one of these answers be called a^{syn}. We feed D, H_{j+1} (all the questions and answers before a^{syn}), and a^{syn} to CQG_θ to obtain the synthetic question of q^{syn}. We refer to all generated synthetic questions and real questions of history as the pool of questions (P_k) for q_k.

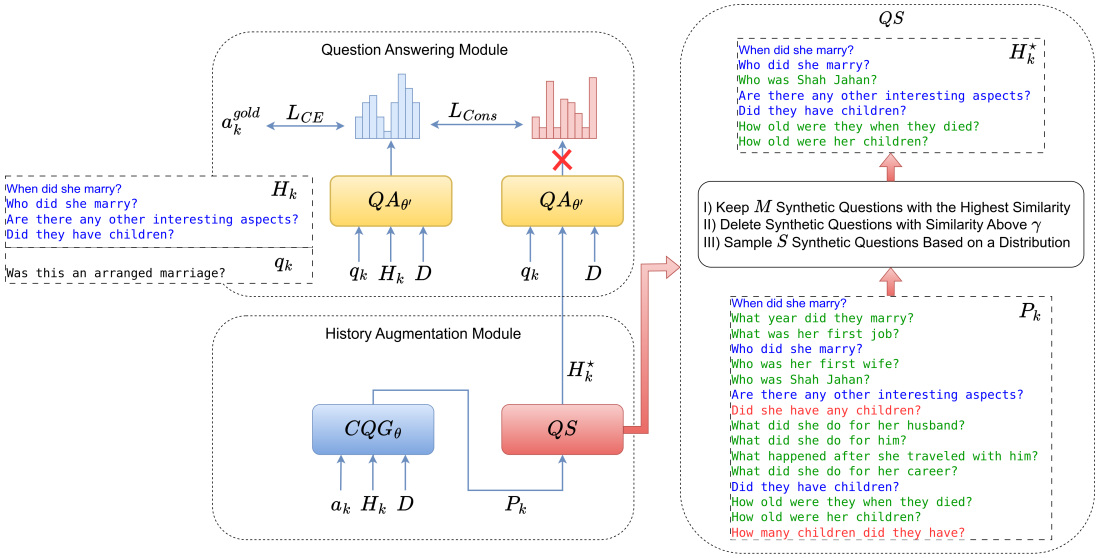

Figure 1: **Architecture of the Model**: For a given question q_k, the conversational question generator CQG_θ constructs a pool of questions denoted as P_k. Questions in H_k are shown in blue. The synthetic questions are depicted in red and green: those similar to H_k questions are in red, and the dissimilar ones are in green. The question selector QS selects M questions with the highest scores, discards red questions, and chooses $S = 3$ synthetic questions from the green questions according to uniform distribution, along with H_k questions, to create $H_k^\star$. The QA network $QA_{\theta'}$ computes its output using both H_k and $H_k^\star$ as input. The QA network is trained by minimizing the cross-entropy loss (L_{CE}) and consistency loss (L_{Cons}). q_k and H_k are from the QuAC dataset.

Question Filtering & Injection We could set P_k as $H_k^\star$; however, P_k contains a multitude of synthetic questions which induces too much noise. Additionally, in the consistency training setting, the noise (perturbation) should be small. Thus, we only select S of synthetic questions from P_k, where S is a hyperparameter. Not all synthetic questions are helpful, necessitating the need to filter out degenerate ones. We want our selected synthetic questions to be similar and relevant to the trend of the conversation. To this end, we compute a score for each synthetic question and only keep the top M synthetic questions with the highest score. To compute the score, each question (real or synthetic) is encoded with LaBSE (Feng et al., 2022). For each synthetic question q^{syn} which is located between history turns q_j and q_{j+1}, the score is computed as $Sim(h(q_j), h(q^{syn})) + Sim(h(q_{j+1}), h(q^{syn}))$, where Sim is the cosine similarity function and $h(\mathbf{x})$ is the LaBSE's encoding of the sentence $\mathbf{x}$. Additionally, sometimes, we generate questions that are too similar to previous or future questions, which are invaluable. Thus, we compare the similarity of the generated question q^{syn} with questions in $\{q_k\} \bigcup H_k$ and if the similarity is above γ, q^{syn} is discarded. This situation is depicted in Figure 1, where P_k contains real history questions, depicted in blue, and synthetic questions, depicted in red and green. Those synthetic questions that have high similarity with $\{q_k\} \bigcup H_k$ are depicted in red. As it can be seen, the two questions "Did she have any children" and "How many children did they have" have high similarity with the question "Did they have children", and thus, they're discarded. In addition, we need to set a distribution to guide the selection of S number of generated questions. We conduct experiments using two distributions: uniform and linear. In the uniform setting, the generated questions are selected with the same probability. For the linear, if q^{syn} is located between q_j and q_{j+1}, its probability of being selected ($P(q^{syn})$) is $P(q^{syn}) \propto j$. We opt for the linear distribution, as we believe that closer synthetic questions to the original question might contribute to greater robustness, as questions that are further away are likely less relevant.

4.2 Question Answering Module

For each question q_k, as illustrated in Figure 1, we feed q_k, H_k, and D to the QA network ($QA_{\theta'}$) to compute the answer distribution. In parallel, we feed q_k, $H_k^\star$, and D to the QA network to compute another answer distribution. As mentioned in Section 4, we need to impose the condition outlined in Equation (2). To achieve this, we employ KL-Divergence between the answer distributions.

Additionally, we use cross-entropy loss to train the QA network for answer prediction. The losses are calculated as per Equation (3), where L_{CE}, L_{Cons}, and L_T represent the cross-entropy loss, consistency loss, and total loss. λ is a hyperparameter used to determine the ratio of the two losses.

$$\begin{aligned} L_{CE} &= CE(QA_{\theta'}(q_k, H_k, D), a_k^{gold}) \\ L_{Cons} &= D_{KL}(QA_{\theta'}(q_k, H_k, D), \\ &\qquad QA_{\theta'}(q_k, H_k^{\star}, D)) \\ L_T &= L_{CE} + \lambda L_{Cons} \end{aligned} \tag{3}$$

Furthermore, we acknowledge that augmenting the history for all questions may not be optimal, as initial questions in a dialog, due to their little historical context, may not require augmentation for robust reasoning. In this case augmenting their history might add unnecessary noise, potentially degrading performance. Thus, we introduce a threshold named τ and only augment the history of q_k if $k \geq \tau$. According to Miyato et al. (2019), we only pass the gradients through one network. As shown in the Figure 1, the symbol $\times$ is used to denote gradient cut. It should be noted that our method is model-agnostic, and any architecture could be used as the QA network.

5 Setup

We utilize the QuAC dataset (Choi et al., 2018), to conduct our experiments on, and data splitting is described in A. We utilize BERT (Devlin et al., 2019) as our base model to conduct experiments following the previous research. For question generation, we adopt Bart-Large (Lewis et al., 2020). Following Choi et al. (2018), we use F1, HEQ-Q, and HEQ-D as our evaluation metrics. F1 measures the overlap between a_k^{gold} and a_k^{pred}. HEQ-Q and HEQ-D are the ratio of questions and dialogs, for which the model performs better than human (Choi et al., 2018). We run multiple experiments to choose the best set of hyperparameters, resulting in setting $S = 2$, $\lambda = 2.0$, and $\tau = 6$. In Appendix C, the process of choosing all hyperparameters and their analysis is described. For all of our models, we concatenate the question with history questions, feeding them to the network. More details on reproducibility are presented in Appendix E.

6 Results

6.1 Question Generation Results

The results of question generation are evaluated in Table 1. These scores are obtained from the dev data. Bleu-1,4 (Papineni et al., 2002), Rouge-L (Lin, 2004), and BERTScore (Zhang et al., 2020) are used for criteria. We use the evaluate library[1] to implement these metrics. Find more details in Appendix B.

Table 1: Question generation results on the dev set.

Bleu-1	Bleu-4	Rouge-L	BERTScore
33.6	9.5	29.0	90.5

6.2 Baselines Performance

Table 2 shows the results of our experiments in comparison to other baselines. As stated before, BERT-HAE, HAM, and GraphFlow leverage the gold answers of history. BERT-HAE re-implementation by Siblini et al. (2021), and those of HAM and GraphFlow by Li et al. (2022) are shown in the table as BERT-HAE-Real, HAM-Real, and GraphFlow-Real, respectively, indicating a significant drop in performance. [2] In this scenario, where common baselines experience a substantial decrease, we use a basic BERT model with history concatenation as the baseline, as its performance is superior. We include the results of the reinforced history backtracking model (Qiu et al., 2021) in the table. Since this model's code is not publicly available, we have been unable to re-implement it with the correct settings and perform a meaningful comparison. However, it's worth noting that this model utilizes unrealistic settings in two stages: once for history selection and once for question answering, potentially exacerbating the modeling issues even further. We have used "Unrealistic Settings" as a term to indicate that a method uses gold answers from history in its modeling.

6.3 CoTaH Results Analysis

In Table 2, CoTaH-BERT outperforms BERT (Baseline) by 1.8% in the F1 score[3]. According to Figure 2 in Appendix D, this improvement is mostly due to an improvement in the performance of questions with a large amount of history. This

[1] https://github.com/huggingface/evaluate

[2] For a fair comparison, the ExCorD (Kim et al., 2021) model result is not included in this table, as its best-performing model by Kim et al. (2021) and the re-implementation by Li et al. (2022) use RoBERTa (Liu et al., 2019).

[3] It should be noted that our test set for BERT (Baseline) and CoTaH-BERT is different from previous methods, but it has been drawn from the same distribution.

Table 2: Comparison of our methods with other benchmarks on the test set. Hist.: History.

Model Name	F1	HEQ-Q	HEQ-D	Unrealistic Settings
GraphFlow-Real (Li et al., 2022)	49.6	-	-	
BERT-HAE-Real (Siblini et al., 2021)	53.5	-	-	
HAM-Real (Li et al., 2022)	57.2	-	-	
BERT (Baseline)	58.9	52.9	5.3	
CoTaH-BERT	60.7	55.3	5.9	
BERT-HAE (Qu et al., 2019a)	62.4	57.8	5.1	✓
HAM (Qu et al., 2019b)	64.4	60.2	6.1	✓
GraphFlow (Chen et al., 2020)	64.9	60.3	5.1	✓
Reinforced Hist. Backtracking (Qiu et al., 2021)	66.1	62.2	7.3	✓

confirms that our intuition is valid that our method enhances the base model's ability to answer questions with a large historical context. Moreover, while BERT-HAE outperforms CoTaH-BERT in terms of F1 score, CoTaH-BERT exhibits superior performance in HEQ-D. This highlights the better consistency of our model to maintain its performance throughout the entire dialog, which is achieved through superiority in answering the questions in the later turns.

Table 3: Unanswerable accuracy on the test set.

	Unanswerable Accuracy
BERT (Baseline)	61.9
CoTaH-BERT	**68.6**

Avoiding answering unanswerable questions is an indication of language understanding (Zhu et al., 2019). Table 3 shows that CoTaH-BERT brings a considerable improvement in terms of detecting unanswerable questions.

6.4 Ablation Study

Table 4 demonstrates the effectiveness of using the threshold (τ) in enhancing the model capability, with more details provided in Appendix C. Moreover, the table indicates that question filtering has a tangible effect on improving performance by filtering out degenerate questions with high similarity. Lastly, we observe that using a uniform distribution is more advantageous than a linear one for question selection. We observe a relatively 1% drop in both F1 and HEQ-Q scores with the linear distribution, concluding that our hypothesis has not been true regarding the greater robustness that the linear distribution might pose. We suspect that since the linear distribution picks more synthetic questions near the original question, it undermines the importance of immediate history, which is potentially more important than distant history, causing the consistency loss to act as a misleader instead of a regularizer in some cases.

Table 4: The effect of threshold, question filtering, and question selection distribution type on the dev set. QS Dist.: Question Selection Distribution.

CoTaH-BERT	F1	HEQ-Q	HEQ-D
w/o Threshold	59.4	54.8	5.1
w/ Threshold	**59.9**	**55.2**	**5.5**
w/o Question Filtering	59.9	55.2	**5.5**
w/ Question Filtering	**60.9**	**56.3**	5.3
w/ Linear QS Dist.	59.9	55.2	**5.9**
w/ Uniform QS Dist.	**60.9**	**56.3**	5.3

7 Conclusions

In this paper, we introduced a novel model-agnostic method to make the reasoning of conversational question-answering models robust to irrelevant history. We coped with this issue by augmenting the history and training the model with consistency training. In our experiments, we didn't follow the wrong modeling of past research in using the gold answers of history. We examined our method with BERT which exhibited a 1.8% performance boost compared to the baseline model. It was demonstrated that this improvement is primarily attributed to the enhancement of the model's performance on questions with a substantial historical context, suggesting that our method has been successful in making the reasoning robust for these questions.

8 Limitations

Our model requires a phase of question generation. For synthetic question generation, the history augmentation module could be slow and the speed is directly correlated to the number of questions that one opts to generate. However, question generation is trained only once and all questions are generated in a single run, and all other experiments are conducted by only training the QA module. Moreover, although our model doesn't need any further computation during evaluation than merely running the QA network, we need two forward passes during the training phase, which makes the training of the QA network a bit more time-consuming than training the baseline model. We have used only the QuAC dataset to report our experiments. This choice was made so that we are able to compare our results with other research, such as Qu et al. (2019a), Qu et al. (2019b), Siblini et al. (2021), and Li et al. (2022), which only use QuAC for their experiments. Thus, other datasets, such as CoQA (Reddy et al., 2019), are not tested in our research. Lastly, our research does not cover experiments on high-performing large language models, like ChatGPT. Brown et al. (2020) reports the results on the QuAC, using GPT-3 (Brown et al., 2020) in zero-shot, one-shot, and few-shot manners. However, these results are substantially inferior compared to other fine-tuning-based models that are mentioned in Table 2. Therefore, further experiments on ChatGPT and other state-of-the-art large language models are needed to better determine the placement of CoTaH and previous baselines in terms of performance.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Graphflow: Exploiting conversation flow with graph neural networks for conversational machine comprehension. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1230–1236. ijcai.org.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto. 2021. Chaincqg: Flow-aware conversational question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2061–2070. Association for Computational Linguistics.

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2019. Flowqa: Grasping flow in history for conversational machine comprehension. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Gangwoo Kim, Hyunjae Kim, Jungsoo Park, and Jaewoo Kang. 2021. Learn to resolve conversational dependency: A consistency training framework for conversational question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6130–6141. Association for Computational Linguistics.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer

Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Huihan Li, Tianyu Gao, Manan Goenka, and Danqi Chen. 2022. Ditch the gold standard: Re-evaluating conversational question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8074–8085. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced dynamic reasoning for conversational question generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2114–2124. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Minghui Qiu, Xinjing Huang, Cen Chen, Feng Ji, Chen Qu, Wei Wei, Jun Huang, and Yin Zhang. 2021. Reinforced history backtracking for conversational question answering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13718–13726. AAAI Press.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. BERT with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1133–1136. ACM.

Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. 2019b. Attentive history selection for conversational question answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1391–1400. ACM.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Wissam Siblini, Baris Sayil, and Yacine Kessaci. 2021. Towards a more robust evaluation for conversational question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP, Virtual Event*, pages 1028–1034.

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

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Haichao Zhu, Li Dong, Furu Wei, Wenhui Wang, Bing Qin, and Ting Liu. 2019. Learning to ask unanswerable questions for machine reading comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4238–4248. Association for Computational Linguistics.

A Data Splitting

Since the test set of QuAC is not publicly available, we divide the development (dev) set into dev/test sets randomly, such that the number of questions in dev and test sets is almost equal. The total number

of dev and test questions is 3678 and 3676, respectively, after splitting. In our splitting, each dialog, with all of its questions, is either attributed to the dev set or the test set, in order to prevent test data leakage. Further, according to Choi et al. (2018), the original dev set of QuAC contains unique documents, meaning that a single document will not be shared among the final dev and test sets, potentially preventing test data leakage.

B Question Generation Considerations

Gu et al. (2021) reports better results for the question generation, yet we didn't aim to optimize Bart-Large meticulously as the generated questions have a good quality for our task. The point is that in this research, we only utilize questions alone without considering answers. Thus, if the generated questions have less correlation with answers, it's tolerable as they are still relevant questions considering the overall flow of the conversation. It should be noted that if a future research wants to incorporate predicted answers into its modeling, it should be more cautious about the quality of the question generation to ensure that the right synthetic questions are generated concerning their answers. Moreover, it should be noted that while it is true we use gold answers from history in the training of CQG_θ, this does not threaten the realism of our model. The point is that only the training set of QuAC is used to train CQG_θ, and later, the history of the training set is augmented for the use of the QA network. On the other hand, we never augment the history of the dev and test sets for the use of the QA network.

C Hyperparameter Selection & Sensitivity Analysis

Initially, we determine M and γ by assessing some examples of the training data, setting $M = 10$ and $\gamma = 0.8$ based on our appraisal. Next, we determine the values of S, λ, and τ by conducting experiments on the dev set. In Table 5, we evaluate the effects of the model's two main hyperparameters, S and λ, through a grid search with the following values: $S \in \{1, 2, 3\}$ and $\lambda \in \{1.0, 1.5, 2.0\}$. Firstly, it is evident that the model performs better when $S \in \{1, 2\}$ compared to when $S = 3$ overall. This suggests that $S = 3$ introduces too much noise, which could be detrimental to performance. Furthermore, when $\lambda \in \{1.5, 2.0\}$, the performance is better compared to $\lambda = 1.0$, indicating that the introduction of λ is helpful, as simply adding L_{CE} and L_{KL} (or equally setting $\lambda = 1.0$) produces inferior performance. For the remaining experiments, we set $S = 2$ and $\lambda = 2.0$ as these settings yield the best F1 and HEQ-Q scores.

Table 5: The effect of S and λ on the dev set.

		F1	HEQ-Q	HEQ-D
$S = 1$	$\lambda = 1.0$	58.6	53.5	4.8
	$\lambda = 1.5$	59.1	54.8	**5.5**
	$\lambda = 2.0$	59.0	54.2	4.4
$S = 2$	$\lambda = 1.0$	57.9	52.7	4.0
	$\lambda = 1.5$	58.2	53.5	4.2
	$\lambda = 2.0$	**59.4**	**54.8**	5.1
$S = 3$	$\lambda = 1.0$	58.3	53.5	5.1
	$\lambda = 1.5$	58.6	53.5	5.0
	$\lambda = 2.0$	58.8	54.1	4.2

After setting the right amount for S and λ, we opt to examine whether the introduction of the threshold (τ) is effective. Thus, we conduct experiments on three different amounts of this hyperparameter. In Table 6, it's evident that the right amount of τ has a considerable effect on the performance, confirming our intuition about the functionality of τ. For all tested values of τ within the set $\{5, 6, 7\}$, performance has increased compared to the base settings with $\tau = 0$ (or equivalently, using no threshold). Notably, the maximum performance improvement is observed when $\tau = 6$.

Table 6: The effect of τ on the dev set

	F1	HEQ-Q	HEQ-D
$\tau = 0$	59.4	54.8	5.1
$\tau = 5$	59.6	55.2	5.5
$\tau = 6$	**59.9**	**55.2**	**5.5**
$\tau = 7$	59.5	54.9	5.1

D Additional Results

In Figure 2, a comparison between the F1 scores of questions for each turn in BERT and CoTaH-BERT on the test set is presented. The score for the k-th turn represents the average F1 score for all questions in the k-th turn across all dialogs in the test set. Questions with a considerable amount of historical context are answered more effectively with our method. For $0 \leq k \leq 1$, the performances of both

BERT and CoTaH-BERT are nearly equal, which is sensible as these questions contain little historical context and thus have little irrelevant history. However, for most of $k > 1$ dialog turns, CoTaH-BERT outperforms BERT or it has on par performance with BERT. The performance upgrade is especially evident towards the end of dialogs, where questions contain significant historical context. This finding indicates the superiority of CoTaH-BERT over BERT in establishing greater robustness in answering these questions, by identifying and ignoring the irrelevant history turns.

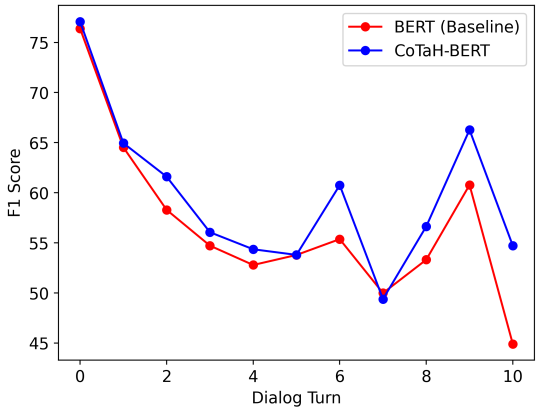

Figure 2: The F1 score of the test set dialog turns

A case study regarding the performance comparison of CoTaH-BERT and BERT for a question from QuAC dataset (Choi et al., 2018) with a large history is provided in Appendix F.

E Reproducibility

The seed for all experiments, except the training of CQG_θ, is 1000. All of the experiments to train the $QA_{\theta'}$ are conducted on a single RTX 3070 Ti with 8GB memory, on which each experiment takes approximately 6 hours. CQG_θ is trained on a single Tesla T4 from Google Colab. For each model, BERT or CoTaH-BERT, the hyperparameters are optimized on the dev set, and a final model will be trained on the train set with the optimized hyperparameters. Subsequently, a single result on the test set will be reported as depicted in Table 2. The source code can be found on our GitHub page.[4]

F Case Study

In Figure 3, a document sample with its corresponding dialog in the dev set is depicted. In the figure, the ninth turn question, q_9, with its history, H_9, are shown. The answers of BERT and CoTaH-BERT to q_9 are compared, showing that CoTaH-BERT has been successful in answering this question with a full F1 score, while BERT has been unsuccessful. q_9 asks about the release date of the album stated in q_2. This is a suitable sample for our context, as there are significant irrelevant history turns between q_9 and q_2. We observe that CoTaH-BERT has been successful in identifying the relevant history by answering the question correctly. However, the BERT model has mistakenly reported another date, which is wrong. As BERT has returned a span containing the word "mixing", it's possible that BERT has incorrectly identified the previous turn question, q_8, as relevant and has returned a span by text matching encompassing the word "mixing", and containing merely some random dates.

[4] https://github.com/HamedHematian/SynCQG

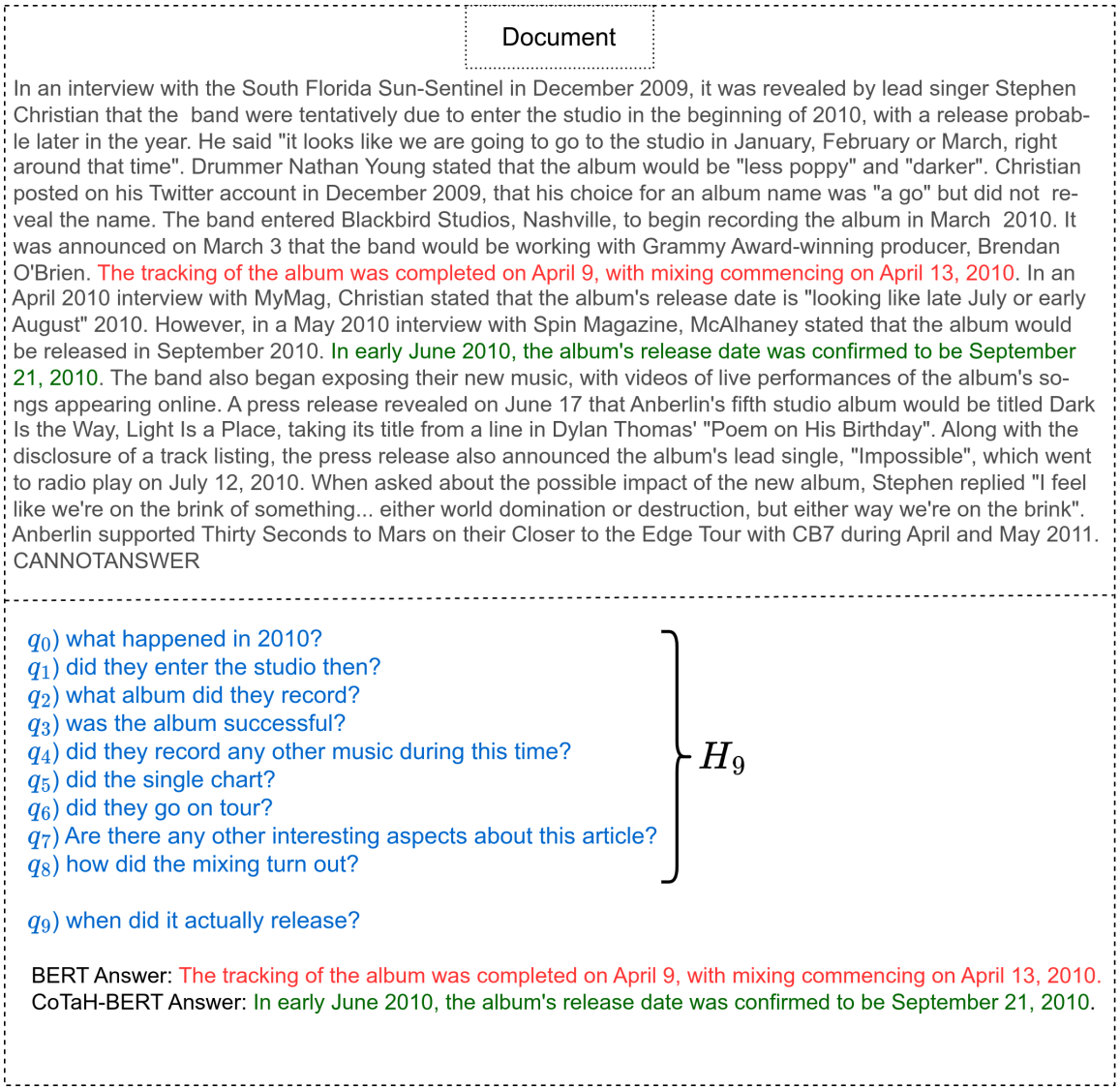

Figure 3: A comparison between BERT and CoTaH-BERT extracted answers to a question, showing that CoTaH-BERT has been able to successfully ignore the irrelevant history by extracting the correct answer. However, the BERT model has been confused and returned a wrong answer. The dialog and the document are presented from the QuAC dataset (Choi et al., 2018).