# A Simple but Effective Method to Incorporate Multi-turn Context with BERT for Conversational Machine Comprehension

**Yasuhito Ohsugi  Itsumi Saito  Kyosuke Nishida  Hisako Asano  Junji Tomita**
NTT Media Intelligence Laboratories, NTT Corporation
yasuhito.ohsugi.va@hco.ntt.co.jp

## Abstract

Conversational machine comprehension (CMC) requires understanding the context of multi-turn dialogue. Using BERT, a pre-training language model, has been successful for single-turn machine comprehension, while modeling multiple turns of question answering with BERT has not been established because BERT has a limit on the number and the length of input sequences. In this paper, we propose a simple but effective method with BERT for CMC. Our method uses BERT to encode a paragraph independently conditioned with each question and each answer in a multi-turn context. Then, the method predicts an answer on the basis of the paragraph representations encoded with BERT. The experiments with representative CMC datasets, QuAC and CoQA, show that our method outperformed recently published methods (+0.8 F1 on QuAC and +2.1 F1 on CoQA). In addition, we conducted a detailed analysis of the effects of the number and types of dialogue history on the accuracy of CMC, and we found that the gold answer history, which may not be given in an actual conversation, contributed to the model performance most on both datasets.

## 1 Introduction

Single-turn machine comprehension (MC) has been studied as a question answering method (Seo et al., 2016; Chen et al., 2017; Yu et al., 2018; Lewis and Fan, 2019). Conversational artificial intelligence (AI) such as Siri and Google Assistant requires answering not only a single-turn question but also multi-turn questions in a dialogue. Recently, two datasets, QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2018), were released to answer sequential questions in a dialogue by comprehending a paragraph. This task is called conversational machine comprehension (CMC) (Huang et al., 2019), which requires understanding the context of multi-turn dialogue that consists of the question and answer history.

Learning machine comprehension models requires a lot of question answering data. Therefore, transfer learning from pre-training language models based on a large-scale unlabeled corpus is useful for improving the model accuracy. In particular, BERT (Devlin et al., 2018) achieved state-of-the-art results when performing various tasks including the single-turn machine comprehension dataset SQuAD (Rajpurkar et al., 2016). BERT takes a concatenation of two sequences as input during pre-training and can capture the relationship between the two sequences. When adapting BERT for MC, we use a question and a passage as input and fine-tune the pre-trained BERT model to extract an answer from the paragraph. However, BERT can accept only two sequences of 512 tokens and thus cannot handle CMC naively.

Zhu et al. (2018) proposed a method for CMC that is based on an architecture for single-turn MC and uses BERT as a feature-based approach. To convert CMC into a single-turn MC task, the method uses a reformulated question, which is the concatenation of the question and answer sequences in a multi-turn context with a special token. It then uses BERT to obtain contextualized embeddings for the reformulated question and paragraph, respectively. However, it cannot use BERT to capture the interaction between each sequence in the multi-turn context and the paragraph.

In this paper, we propose a simple but effective method for CMC based on a fine-tuning approach with BERT. Our method consists of two main steps. The first step is contextual encoding where BERT is used for independently obtaining paragraph representations conditioned with the current question, each of the previous questions, and each of the previous answers. The second step



*Proceedings of the 1st Workshop on NLP for Conversational AI*, pages 11–17
Florence, Italy, August 1, 2019. 

is answer span extraction, where the start and end position of the current answer are predicted based on the concatenation of the paragraph representations encoded in the previous step.

The contributions of this paper are as follows:

- We propose a novel method for CMC based on fine-tuning BERT by regarding the sequences of the questions and the answers as independent inputs.
- The experimental results show that our method outperformed published methods on both QuAC and CoQA.
- We found that the gold answer history contributed to the model performance most by analyzing the effects of dialogue history.

## 2 Task Definition

In this paper, we define the CMC task as follows:

- **Input**: Current question $Q_i$, paragraph $P$, previous questions $\{Q_{i-1}, ..., Q_{i-k}\}$, and previous answers $\{A_{i-1}, ..., A_{i-k}\}$
- **Output**: Current answer $A_i$ and type $T_i$

where $i$ and $k$ denote the turn index in the dialogue and the number of considered histories (turns), respectively. Answer $A_i$ is a span of paragraph $P$. Type $T_i$ is *SPAN*, *YES*, *NO*, or *UNANSWERABLE*.

## 3 Pre-trained Model

BERT is a powerful language representation model (Devlin et al., 2018), which is based on bi-directional Transformer encoder (Vaswani et al., 2017). BERT can obtain language representation by unsupervised pre-training with a huge data corpus and by supervised fine-tuning, and it can achieve outstanding results in various NLP tasks such as sentence pair classification, single sentence tagging, and single-turn machine comprehension.

Here, we explain how to adapt BERT for single-turn machine comprehension tasks such as SQuAD (Rajpurkar et al., 2016). In SQuAD, a question and a paragraph containing the answer are given, and the task is to predict the answer text span in the paragraph. In the case of using BERT for SQuAD, after the special classification token [CLS] is added in front of the question, the question and the paragraph are concatenated with special tokens [SEP] into one sequence. The sequence is inputted to BERT with segment embeddings and positional embeddings. Then, the final hidden state of BERT is converted to the probabilities of answer span by a linear layer and softmax function. The fined-tuned BERT for the SQuAD dataset can capture the relationship between one question and one paragraph so that BERT achieved state-of-the-art performance on the SQuAD. However, BERT itself cannot be used for a task requiring multiple queries or multiple paragraphs, because BERT can accept only two segments in one input sequence. This limitation can be a problem for the CMC task because there are multi-turn questions about the same paragraph.

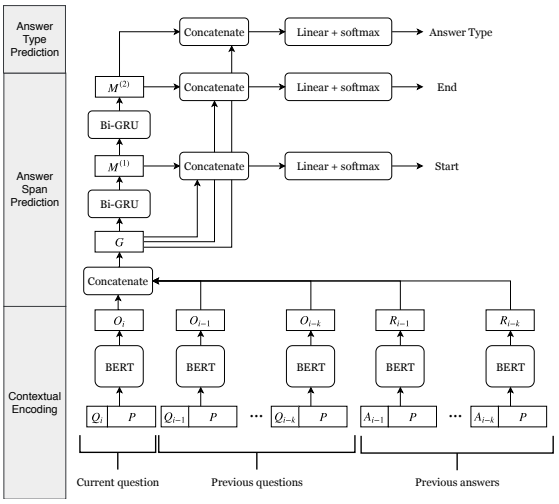


Figure 1: Our model

## 4 Proposed Method

In the CMC task, it is necessary to consider not only the current question $Q_i$ but also the question history $\{Q_{i-1}, ..., Q_{i-k}\}$ and the answer history $\{A_{i-1}, ..., A_{i-k}\}$. We propose a method of modeling the current question, question history, and answer history by using BERT (Figure 1). Our method consists of two steps: contextual encoding and answer span prediction. On top of that, answer type is predicted only in the case of CoQA (see Section 4.3).

### 4.1 Contextual Encoding

In this step, we use BERT to encode not only the relationship between the current question and the paragraph but also the relationship between the history and the paragraph. We define the method

of extracting features by using BERT as follows,

$$\boldsymbol{z} = f(\text{BERT}(x, y|\theta)), \quad (1)$$

where $x$, $y$, and $\boldsymbol{z}$ denote the input query sequence, input paragraph sequence, and output feature, respectively. The function BERT$(\cdot)$ outputs BERT's $d$-dimensional final hidden states with parameters $\theta$, and the function $f(\cdot)$ extracts features corresponding to the segment of the paragraph in the final hidden states. Namely, if input paragraph text $y$ has $T$ tokens, then, $\boldsymbol{z} \in \mathbb{R}^{d \times T}$. This step consists of three parts, and each part shares the BERT parameters $\theta$. First, we encode the current question as follows,

$$\boldsymbol{O_i} = f(\text{BERT}(Q_i, P|\theta)). \quad (2)$$

Second, we encode the question history $\{Q_{i-1}, ..., Q_{i-k}\}$ in the same manner.

$$\boldsymbol{O_{i-l}} = f(\text{BERT}(Q_{i-l}, P|\theta)), \quad (3)$$

where $l$ denotes the index of the previous context. Last, we encode the answer history $\{A_{i-1}, ..., A_{i-k}\}$. Note that previous answer $A_{i-l}$ is given as text, even if the current answer is predicted as the span of the paragraph. The encoded feature can be obtained as follows,

$$\boldsymbol{R_{i-l}} = f(\text{BERT}(A_{i-l}, P|\theta)). \quad (4)$$

### 4.2 Answer Span Prediction

In this step, the current answer span is predicted. Let $s_i$ and $e_i$ represent the start index and the end index, respectively. First, the output features of the previous step are concatenated as follows,

$$\boldsymbol{G} = [\boldsymbol{O_i}; \boldsymbol{O_{i-1}}; ...; \boldsymbol{O_{i-k}}; \boldsymbol{R_{i-1}}; ...; \boldsymbol{R_{i-k}}], \quad (5)$$

where $[;]$ is vector concatenation across row and $\boldsymbol{G} \in \mathbb{R}^{(2k+1)d \times T}$. Then, $\boldsymbol{G}$ is passed to BiGRU over tokens and converted to $\boldsymbol{M^{(1)}} \in \mathbb{R}^{2d \times T}$. To predict the start index $s_i$, the probability distribution is calculated by,

$$p^{\text{s}} = \text{softmax}\left(\boldsymbol{w}_1^\top [\boldsymbol{G}; \boldsymbol{M}^{(1)}] + \boldsymbol{b}_1\right), \quad (6)$$

where $\boldsymbol{w}_1$ and $\boldsymbol{b}_1 \in \mathbb{R}^{(2k+3)d}$ are trainable vectors. Next, to predict the end index $e_i$, $\boldsymbol{M^{(1)}}$ is passed to another BiGRU over tokens and converted to $\boldsymbol{M^{(2)}} \in \mathbb{R}^{2d \times T}$. Then, the probability distribution is calculated by

$$p^{\text{e}} = \text{softmax}\left(\boldsymbol{w}_2^\top [\boldsymbol{G}; \boldsymbol{M}^{(2)}] + \boldsymbol{b}_2\right), \quad (7)$$

where $\boldsymbol{w}_2$ and $\boldsymbol{b}_2 \in \mathbb{R}^{(2k+3)d}$ are trainable vectors.

### 4.3 Answer Type Prediction

Some questions should be simply answered as "yes" or "no" and not answered as a rationale text. To address these questions, the probability of the answer type is calculated as follows,

$$p^{\text{ans}} = \left[\text{softmax}\left(\boldsymbol{w}_3^\top [\boldsymbol{G}; \boldsymbol{M}^{(2)}] + \boldsymbol{b}_3\right)\right]_{e_i}, \quad (8)$$

where $\boldsymbol{w}_3$ and $\boldsymbol{b}_3 \in \mathbb{R}^{(2k+3)d}$ are trainable vectors and $e_i$ is the end index of the predicted span.

### 4.4 Fine-tuning and Inference

In the fine-tuning phase, we regard the sum of the negative log likelihood of the true start and end indices as training loss,

$$L = -\frac{1}{N}\sum_{l=1}^{N}\left[\log(p^{\text{s}}_{y_l^1}) + \log(p^{\text{e}}_{y_l^2})\right], \quad (9)$$

where $N$, $y_l^1$, and $y_l^2$ denote the number of examples, true start, and true end indices of the $l$-th example, respectively. If answer type prediction is necessary, we add the cross entropy loss of the answer type to the training loss. In the inference phase, the answer span $(s_i, e_i)$ is calculated by dynamic programming, where the values of $p^{\text{s}}$ and $p^{\text{e}}$ are maximum and $1 \leq s_i \leq e_i \leq T$.

## 5 Experiment

In this section, we evaluate our method on two conversational machine comprehension datasets, QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2018).

### 5.1 Datasets and Evaluation Metrics

Although CoQA is released as an abstractive CMC dataset, Yatskar (2018) shows that the extractive approach is also effective for CoQA. Thus, we also use our extractive approach on CoQA. To handle answer types in CoQA, we predict the probability distribution of the answer type (*SPAN*, *YES*, *NO*, and *UNANSWERABLE*) and replace the predicted span with "yes", "no", or "unknown" tokens except for the "SPAN" answer type. In QuAC, the unanswerable questions are handled as an answer span ($P$ contains a special token), and the type prediction for yes/no questions is not evaluated on the leaderboard. Therefore, we skip the answer type prediction step.

| | In-domain | | | | | Out-of-domain | | In-domain | Out-of-domain | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Child. | Liter. | Mid-High. | News | Wiki | Reddit | Science | overall | overall | Overall |
| DrQA + PGNet | 64.2 | 63.7 | 67.1 | 68.3 | 71.4 | 57.8 | 63.1 | 67.0 | 60.4 | 65.1 |
| BiDAF++ (3-ctx) | 66.5 | 65.7 | 70.2 | 71.6 | 72.6 | 60.8 | 67.1 | 69.4 | 63.8 | 67.8 |
| FlowQA (1-ans) | 73.7 | 71.6 | 76.8 | 79.0 | 80.2 | 67.8 | 76.1 | 76.3 | 71.8 | 75.0 |
| SDNet (single) | 75.4 | 73.9 | 77.1 | 80.3 | **83.1** | 69.8 | 76.8 | 78.0 | 73.1 | 76.6 |
| BERT w/ 2-ctx | **76.0** | **77.0** | **80.5** | **82.1** | 83.0 | **72.5** | **79.6** | **79.8** | **75.9** | **78.7** |
| ConvBERT (single) | - | - | - | - | - | - | - | 87.7 | 84.6 | 86.8 |
| Google SQuAD 2.0 + MMFT (single) | - | - | - | - | - | - | - | 88.5 | 86.0 | 87.8 |

Table 1: The results on the CoQA test set of single models ($F_1$ score). Our BERT w/ 2-ctx model ranked 13th among all unpublished and published models (including ensemble) on the leaderboard at the submission time (April 13, 2019). The ConvBERT and the Google SQuAD 2.0 + MMFT are the current state-of-the-art models, but they are unpublished.

As evaluation metrics for CoQA, we use the $F_1$ score. CoQA contains seven domains as paragraph contents: childrens stories, literature, middle and high school English exams, news articles, Wikipedia articles, science articles, and Reddit articles. We report $F_1$ for each domain and the overall domains. On the other hand, as evaluation metrics of QuAC, we use not only $F_1$ but also the human equivalence score for questions (HEQ-Q) and for dialogues (HEQ-D) (Choi et al., 2018). HEQ-Q represents the percentage of exceeding the model performance over the human evaluation for each question, and HEQ-D represents the percentage of exceeding the model performance over the human evaluation for each dialogue.

### 5.2 Comparison Systems

We compare our model (BERT w/ k-ctx) with the baseline models and published models. For QuAC, we use the reported scores of BiDAF++ w/ k-ctx (Choi et al., 2018) and FlowQA (Huang et al., 2019). For CoQA, the comparison system is DrQA+PGNet (Reddy et al., 2018), BiDAF++ w/ x-ctx, FlowQA, and SDNet (Zhu et al., 2018). Note that the scores of BiDAF++ w/ x-ctx on CoQA are reported by Yatskar (2018). In addition, we use gold answers as the answer history, except for the investigation of the effect of answer history. More information on our implementation is available in Appendix A.

### 5.3 Results

**Does our model outperform published models on both QuAC and CoQA?** Table 1 and Table 2 show the results on CoQA and QuAC, respectively. On CoQA, our model outperformed all of the published models regarding the overall $F_1$ score. Although our model was comparable with SDNet for the Wikipedia domain, our model outperformed SDNet for the other domains. On QuAC, our model also obtained the best score among the published models for all of the metrics and obtained state-of-the-art scores on March 7th, 2019.

| | $F_1$ | HEQ-Q | HEQ-D |
|---|---|---|---|
| BiDAF++ (2-ctx) | 60.1 | 54.8 | 4.0 |
| FlowQA (2-ans) | 64.1 | 59.6 | 5.8 |
| BERT w/ 2-ctx | **64.9** | **60.2** | **6.1** |
| ConvBERT (single) | 68.0 | 63.5 | 9.1 |
| Bert-FlowDelta (single) | 67.8 | 63.6 | 12.1 |

Table 2: The results on the QuAC test set of single models. Our BERT w/ 2-ctx model ranked 1st among all unpublished and published models on the leaderboard at the submission time (March 7, 2019). The ConvBERT and Bert-FlowDelta are the current state-of-the-art models, but they are unpublished.

| | # contexts | CoQA | QuAC |
|---|---|---|---|
| BERT w/ 0-ctx | 0 | 72.8 | 55.0 |
| BERT w/ 1-ctx | 1 | 79.2 | 63.4 |
| BERT w/ 2-ctx | 2 | 79.6 | **65.4** |
| BERT w/ 3-ctx | 3 | 79.6 | 65.3 |
| BERT w/ 4-ctx | 4 | 79.4 | 64.8 |
| BERT w/ 5-ctx | 5 | **79.7** | 64.5 |
| BERT w/ 6-ctx | 6 | 79.5 | 64.9 |
| BERT w/ 7-ctx | 7 | **79.7** | 64.4 |

Table 3: The results with the number of previous contexts on the development set of QuAC and CoQA ($F_1$ score)

Our method uses the paragraph representations independently conditioned with each question and each answer. This model structure is suitable for the pre-trained BERT, which was trained with two input segments. Therefore, our model was able to capture the interaction between a dialogue history and a paragraph, and it achieved high accuracy.

| | CoQA | QuAC |
|---|---|---|
| BERT w/ 0-ctx | 72.8 | 55.0 |
| BERT w/ 2-ctx (gold ans.) | **79.6** | **65.4** |
| w/o question history | 78.0 | 64.7 |
| w/o answer history | 77.7 | 59.3 |
| BERT w/ 2-ctx (predicted ans.) | 77.2 | 56.7 |

Table 4: Ablation study on the development set of QuAC and CoQA ($F_1$ score)

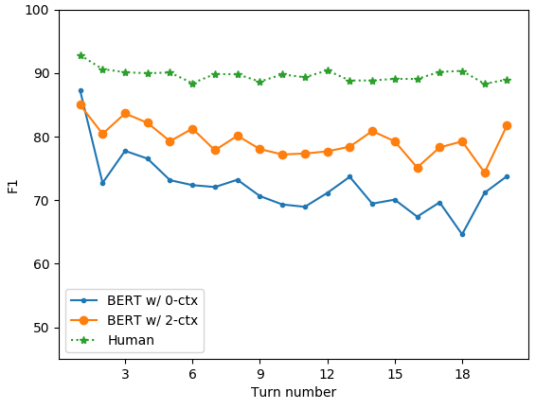


Figure 2: The $F_1$ scores with turn number on CoQA development set

**Does our model improve the performance when the number of previous contexts increases?** Table 3 shows the results with the number of previous contexts. On both of the datasets, it was effective to use previous contexts. However, on CoQA, the number of contexts had little effect on the score even if the long context was considered. On QuAC, the best score was obtained in the case of using two contexts, and the score decreased with more than two contexts. As Yatskar (2018) mentioned, the topics in a dialogue shift more frequently on QuAC than on CoQA. Thus, the previous context on QuAC can include the context that is unrelated to the current question, and this unrelated context can decrease the score. This result suggests that it is important to select context that is related to the current question and not use the whole context in any cases.

**Which is more important, the question history or the answer history?** Table 4 shows the contribution of the dialogue history. We can see from the results that the model performance decreased significantly when we removed the gold answer history on QuAC. In dataset collection, CoQA allows the asker to see the evidence paragraph. On the other hand, the asker in QuAC cannot see

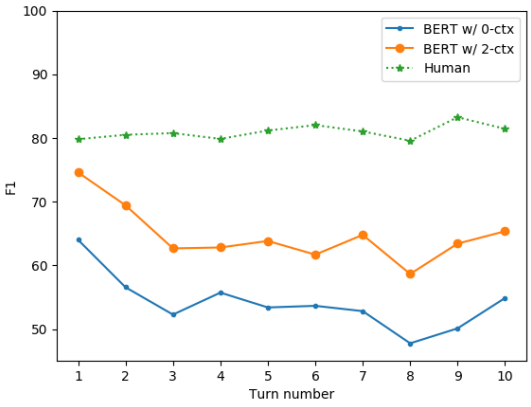


Figure 3: The $F_1$ scores with turn number on QuAC development set

the evidence paragraph. As a result, questions in QuAC are far from the phrases in the passage and are less effective in improving the model performance. For CoQA, the model could substitute the question history for the gold answer history. The model performance did not decrease significantly when we remove the answer history.

**Does our model maintain the performance when using the predicted answer history?** In actual conversation, the gold answer history may not be given in the CMC model. In this experiment, we trained the models with the gold answer history and evaluated the model with the predicted answer history.

As shown in Table 4, when using the predicted answer history, the model performance decreased significantly on QuAC. This result also suggests that the model can substitute the question history for the gold answer history in CoQA. We think the CMC setting where the history of questions posed by an asker that does not see the evidence paragraph is given and the gold answer is not given for input is a more realistic and important setting.

**Does our model performance approach human performance as the dialogue progresses?** We calculated $F_1$ scores over the turns, where the data in each turn contained more than 100 question/answer pairs. Figure 2 and Figure 3 show that the score was lower than human performance over all turns on both datasets and that the score with context was higher than that without context on both datasets, except for the first question on CoQA. This result indicates that there is still room for improvement with long turn questions.

## 6 Related Work

QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2018) were released as the CMC dataset. On QuAC, the answers are extracted from source paragraph as spans. On CoQA, the answers are free texts based on span texts extracted from the source paragraph. On these datasets, the baseline models were based on conventional models for single-turn machine comprehension such as BiDAF (Seo et al., 2016) and DrQA (Chen et al., 2017). For QuAC, Choi et al. (2018) extended BiDAF (an extractive machine comprehension model) to BiDAF++ w/ x-ctx by concatenating word embeddings of the source paragraph and embeddings of previous answer span indexes. For CoQA, Reddy et al. (2018) proposed DrQA+PGNet as an abstractive method by concatenating previous questions and previous answers with special tokens. However, most of the recently published methods about CoQA were extractive approaches, since the abstractive answers on CoQA are based on span texts in the paragraph and Yatskar (2018) shows that the extractive approach is also effective for CoQA. Huang et al. (2019) proposed FlowQA for both QuAC and CoQA by stacking bidirectional recurrent neural networks (RNNs) over the words of the source paragraph and unidirectional RNNs over the conversational turns. Zhu et al. (2018) proposed SDNet for CoQA by regarding the concatenation of previous questions and answers as one query.

Most recently, BERT (Devlin et al., 2018) was proposed as a contextualized language representation that is pre-trained on huge unlabeled datasets. By fine-tuning a supervised dataset, BERT obtained state-of-the-art scores on various tasks including single-turn machine reading comprehension datasets such as SQuAD (Rajpurkar et al., 2016). Since the relationship between words can be captured in advance, pre-training approaches such as BERT and GPT-2 (Radford et al., 2019) can be useful especially for tasks with a small amount of supervised data. For QuAC and CoQA, many approaches on the leaderboard[1,2] use BERT, including SDNet. However, SDNet uses BERT as contextualized word embedding without updating the BERT parameters. This is one of the differences between SDNet and our model.

[1] https://quac.ai/

[2] https://stanfordnlp.github.io/coqa/

## 7 Conclusion

In this paper, we propose a simple but effective method based on a fine-tuning approach with BERT for a conversational machine comprehension (CMC) task. Our method uses questions and answers simply as the input of BERT to model the interaction between the paragraph and each dialogue history independently and outperformed published models on both QuAC and CoQA.

From detailed analysis, we found that the gold answer history, which may not be given in real conversational situations, contributed to the model performance most on both datasets. We also found that the model performance on QuAC decreased significantly when we used predicted answers instead of gold answers. On the other hand, we can substitute the question history for the gold answer history on CoQA. For future work, we will investigate a more realistic and more difficult CMC setting, where the history of questions posed by the asker that does not see the evidence paragraph is given and the gold answer is not given for input. We will also investigate how to obtain related and effective context for the current question in the previous question and answer history.

## References


Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

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*, pages 2174–2184. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Hsin-Yuan Huang, Eunsol Choi, and Wen tau Yih. 2019. FlowQA: Grasping flow in history for conversational machine comprehension. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,*

*ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Mark Yatskar. 2018. A qualitative comparison of coqa, squad 2.0 and quac. *CoRR*, abs/1809.10735.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *CoRR*, abs/1812.03593.

## A Implementation Details

We used the BERT-base-uncased model implemented by PyTorch [3]. We used a maximum sequence length of 384, document stride of 128, maximum query length of 64, and maximum answer length of 30. The optimizer was Adam (Kingma and Ba, 2015) with a learning rate of 3e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L2 weight decay of 0.01, learning rate warmup over the first 10 % of training steps, and linear decay of the learning rate. The number of training epochs was 2. The batch size of training was 8 or 12. In the case of QuAC, we used dialogs whose paragraphs have under 5,000 characters. In the case of CoQA, we followed Huang et al. (2019) and regarded a span with maximum $F_1$ overlap with respect to given abstractive answers as gold answers during training. We used four NVIDIA Tesla V100 32GB GPUs.

[3] https://github.com/huggingface/pytorch-pretrained-BERT