SDNET: CONTEXTUALIZED ATTENTION-BASED DEEP NETWORK FOR CONVERSATIONAL QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Conversational question answering (CQA) is a novel QA task that requires the understanding of dialogue context. Different from traditional single-turn machine reading comprehension (MRC), CQA is a comprehensive task comprised of passage reading, coreference resolution, and contextual understanding. In this paper, we propose an innovative contextualized attention-based deep neural network, SDNet, to fuse context into traditional MRC models. Our model leverages both inter-attention and self-attention to comprehend the conversation and passage. Furthermore, we demonstrate a novel method to integrate the BERT contextual model as a sub-module in our network. Empirical results show the effectiveness of SDNet. On the CoQA leaderboard, it outperforms the previous best model's F_1 score by 1.6%. Our ensemble model further improves the F_1 score by 2.7%.

1 INTRODUCTION

Machine reading comprehension (MRC) is a core NLP task in which a machine reads a passage and then answers related questions. It requires a deep understanding of both the article and the question, as well as the ability to reason about the passage and make inferences. These capabilities are essential in applications like search engines and conversational agents. In recent years, there have been numerous studies in this field (Huang et al., 2017; Seo et al., 2016; Chen et al., 2017; Liu et al., 2017), with various innovations in text encoding, attention mechanisms and answer verification.

However, traditional MRC tasks often take the form of single-turn question answering. In other words, there is no connection between different questions and answers to the same passage. This oversimplifies the conversational manner humans naturally take when probing a passage, where question turns are assumed to be remembered as context to subsequent queries. Figure 1 demonstrates an example of conversational question answering in which one needs to correctly refer "she" in the last two rounds of questions to its antecedent in the first question, "Cotton." To accomplish this kind of task, the machine must comprehend both the current round's question and previous rounds of utterances in order to perform coreference resolution, pragmatic reasoning and semantic implication.

To facilitate research in conversation question answering (CQA), several public datasets have been published that evaluate a model's efficacy in this field, such as CoQA (Reddy et al., 2018), QuAC (Choi et al., 2018) and QBLink (Elgohary et al., 2018). In these datasets, to generate correct responses, models need to fully understand the given passage as well as the dialogue context. Thus, traditional MRC models are not suitable to be directly applied to this scenario.

Therefore, a number of models have been proposed to tackle the conversational QA task. DrQA+PGNet (Reddy et al., 2018) combines evidence finding and answer generation to produce answers. BiDAF++ (Yatskar, 2018) achieves better results by employing answer marking and contextualized word embeddings on the MRC model BiDAF (Seo et al., 2016). FlowQA (Huang et al., 2018) leverages a recurrent neural network over previous rounds of questions and answers to absorb information from its history context.

Once upon a time, in a barn near a farm house, there lived a little white kitten named Cotton. Cotton lived high up in a nice warm place above the barn where all of the farmer's horses slept. But Cotton wasn't alone in her little home above the barn, oh no. She shared her hay bed with her mommy and 5 other sisters...

Q_1: What color was Cotton?
A_1: white

Q_2: Where did **she** live?
A_2: in a barn

Q_3: Did **she** live alone?
A_3: no

Figure 1: Example passage and first three rounds of question and answers from CoQA dataset (Reddy et al., 2018). Pronouns requiring coreference resolution is marked in bold.

In this paper, we propose SDNet, a contextual attention-based deep neural network for the conversational question answering task. Our network stems from machine reading comprehension models, but it has several unique characteristics to tackle context understanding. First, we apply both inter-attention and self-attention on the passage and question to obtain a more effective understanding of the passage and dialogue history. Second, we prepend previous rounds of questions and answers to the current question to incorporate contextual information. Third, SDNet leverages the latest breakthrough in NLP: BERT contextual embeddings (Devlin et al., 2018).

Different from the canonical way of employing BERT as a monolithic structure with a thin linear task-specific layer, we utilize BERT as a contextualized embedder and absorb its structure into our network. To accomplish this, we align the traditional tokenizer with the Byte Pair Encoding (BPE) tokenizer in BERT. Furthermore, instead of using only the last layer's output from BERT (Devlin et al., 2018), we employ a weighted sum of BERT layer outputs to take advantage of all levels of semantic abstraction. Finally, we lock the internal parameters of BERT during training, which saves considerable computational cost. These techniques are also applicable to other NLP tasks.

We evaluate SDNet on the CoQA dataset, and it improves on the previous state-of-the-art F_1 score by 1.6% (from 75.0% to 76.6%). The ensemble model further increases the F_1 score to 79.3%.

2 Approach

In this section, we propose our neural model, SDNet, for the conversational question answering task. We first formulate the problem and then present an overview of the model before delving into the details of the model structure.

2.1 Problem formulation

Given a passage/context $\mathcal{C}$, and question-answer pairs from previous rounds of conversation $Q_1, A_1, Q_2, A_2, ..., Q_{k-1}, A_{k-1}$, the task is to generate response A_k given the latest question Q_k. The response is dependent on both the passage and historic questions and answers.

To incorporate conversation history into response generation, we employ the idea from DrQA+PGNet (Reddy et al., 2018) to prepend the latest N rounds of QAs to the current question Q_k . The problem is then converted into a single-turn machine reading comprehension task, where the reformulated question is $\mathcal{Q}_k = \{Q_{k-N}; A_{k-N}; ..., Q_{k-1}; A_{k-1}; Q_k\}$[1].

2.2 Model Overview

Encoding layer encodes each token in passage and question into a fixed-length vector, which includes both word embeddings and contextualized embeddings. For contextualized embedding, we

[1]To differentiate between question and answering, we add a special word $\langle Q \rangle$ before each question and $\langle A \rangle$ before each answer.

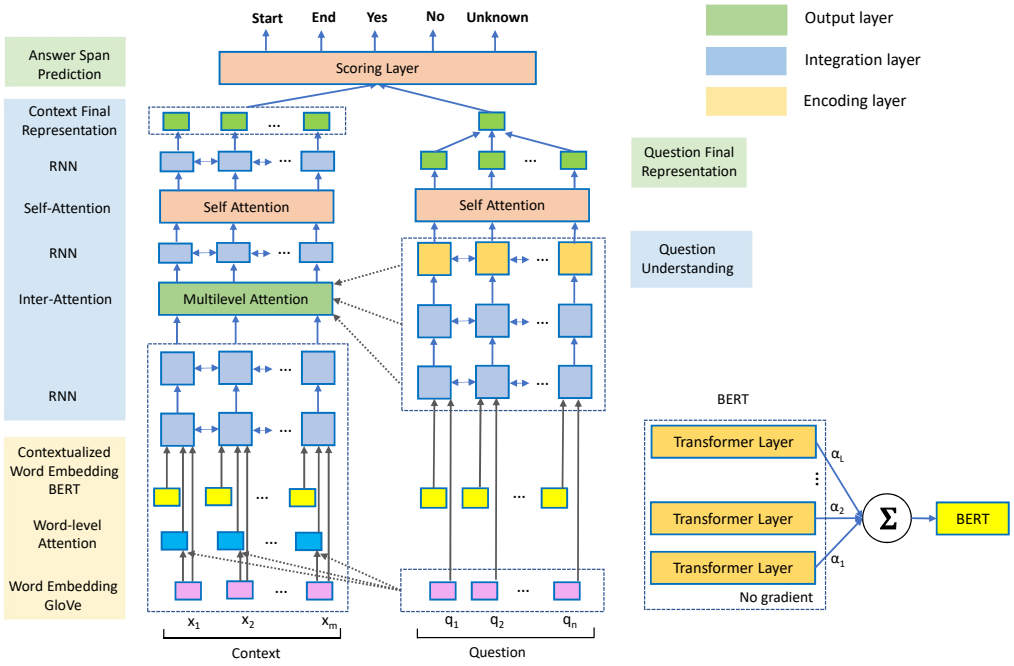

Figure 2: SDNet model structure.

utilize the pretrained language understanding model BERT (Devlin et al., 2018). Different from previous work, we fix the parameters in BERT model and use the linear combination of embeddings from different layers in BERT.

Integration layer uses multi-layer recurrent neural networks (RNN) to capture contextual information within passage and question. To characterize the relationship between passage and question, we conduct word-level attention from question to passage both before and after the RNNs. We employ the idea of history-of-word from FusionNet (Huang et al., 2017) to reduce the dimension of output hidden vectors. Furthermore, we conduct self-attention to extract relationship between words at different positions of context and question.

Output layer computes the final answer span. It uses attention to condense the question into a fixed-length vector, which is then used in a bilinear projection to obtain the probability that the answer should start and end at each position.

An illustration of our model SDNet is in Figure 2.

2.3 Encoding Layer

We first use GloVe (Pennington et al., 2014) embedding for each word in the context and question. Additionally, we compute a feature vector $\boldsymbol{f}_w$ for each context word, following the approach in DrQA (Chen et al., 2017). This feature vector contains a 12-dim POS embedding, an 8-dim NER embedding, a 3-dim exact matching vector em_i indicating whether this word, its lower-case form or its stem appears in the question, and a 1-dim normalized term frequency.

BERT as Contextual Embedder. We design a number of methods to leverage BERT (Devlin et al., 2018) as a contextualized embedder in our model.

First, because BERT uses Byte Pair Encoding (BPE) (Sennrich et al., 2015) as the tokenizer, the generated tokens are sub-words and may not align with traditional tokenizer results. To incorporate BERT into our network, we first use a conventional tokenizer (e.g. spaCy) to get word sequences, and then apply the BPE tokenizer from BERT to partition each word w in the sequence into sub-words $w = (b_1, ..., b_s)$. This alignment makes it possible to concurrently use BERT embeddings and other word-level features. The contextual embedding of w is defined to be the averaged BERT embedding of all sub-words $b_j, 1 \leq j \leq s$.

Second, Devlin et al. (2018) proposes the method to append thin task-specific linear layers to BERT, which takes the result from the last transformer layer as input. However, as BERT contains multiple layers, we employ a weighted sum of these layer outputs to take advantage of information from all levels of semantic abstraction. This can help boost the performance compared with using only the last transformer's output.

Third, as BERT contains hundreds of millions of parameters, it takes a lot of time and space to compute and store their gradients during optimization. To tackle this problem, we lock the internal weights of BERT during training, only updating the linear combination weights. This can significantly increase the efficiency during training, which can be especially useful when computing resource is limited.

To summarize, suppose a word w is tokenized to s BPE tokens $w = (b_1, b_2, ..., b_s)$, and BERT has L layers that generate L embedding outputs for each BPE token, $\boldsymbol{h}_t^l, 1 \leq l \leq L, 1 \leq t \leq s$. The contextual embedding BERT_w for word w is computed as:

$$\text{BERT}_w = \sum_{l=1}^{L} \alpha_l \frac{\sum_{t=1}^{s} \boldsymbol{h}_t^l}{s}, \tag{1}$$

where $\alpha_1, ..., \alpha_L$ are trainable parameters.

2.4 INTEGRATION LAYER

Word-level Inter-Attention. We conduct attention from question to context (passage) based on GloVe word embeddings. Suppose the context word embeddings are $\{\boldsymbol{h}_1^C, ..., \boldsymbol{h}_m^C\} \subset \mathbb{R}^d$, and the question word embeddings are $\{\boldsymbol{h}_1^Q, ..., \boldsymbol{h}_n^Q\} \subset \mathbb{R}^d$. Then the attended vectors from question to context are $\{\hat{\boldsymbol{h}}_1^C, ..., \hat{\boldsymbol{h}}_m^C\}$:

$$S_{ij} = \text{ReLU}(U\boldsymbol{h}_i^C) D \,\text{ReLU}(U\boldsymbol{h}_j^Q) \tag{2}$$

$$\alpha_{ij} \propto e^{S_{ij}}, \tag{3}$$

$$\hat{\boldsymbol{h}}_i^C = \sum_j \alpha_{ij} \boldsymbol{h}_j^Q, \tag{4}$$

where $D \in \mathbb{R}^{k \times k}$ is a diagonal matrix and $U \in \mathbb{R}^{d \times k}$, k is the attention hidden size.

To simplify notation, we denote the above attention function as $\text{Attn}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$, which linearly combines the vector set $\boldsymbol{C}$ using attention scores computed from vector sets $\boldsymbol{A}$ and $\boldsymbol{B}$. This resembles the definition of attention in transformer (Vaswani et al., 2017). It follows that the word-level inter-attention can be rewritten as $\text{Attn}(\{\boldsymbol{h}_i^C\}_{i=1}^m, \{\boldsymbol{h}_i^Q\}_{i=1}^n\}, \{\boldsymbol{h}_i^Q\}_{i=1}^n\})$.

Therefore, the input vector for each context word and question word is:

$$\tilde{\boldsymbol{w}}_i^C = [\text{GloVe}(w_i^C); \text{BERT}_{w_i^C}; \hat{\boldsymbol{h}}_i^C; \boldsymbol{f}_{w_i^C}], \tag{5}$$

$$\tilde{\boldsymbol{w}}_i^Q = [\text{GloVe}(w_i^Q); \text{BERT}_{w_i^Q}] \tag{6}$$

RNN. In this component, we use two separate bidirectional LSTMs (Hochreiter & Schmidhuber, 1997) to form the contextualized understanding for $\mathcal{C}$ and $\mathcal{Q}$:

$$\boldsymbol{h}_1^{C,k}, ..., \boldsymbol{h}_m^{C,k} = \text{RNN}(\boldsymbol{h}_1^{C,k-1}, ..., \boldsymbol{h}_m^{C,k-1}) \tag{7}$$

$$\boldsymbol{h}_1^{Q,k}, ..., \boldsymbol{h}_n^{Q,k} = \text{RNN}(\boldsymbol{h}_1^{Q,k-1}, ..., \boldsymbol{h}_n^{Q,k-1}), \tag{8}$$

where $\boldsymbol{h}_i^{C,0} = \tilde{\boldsymbol{w}}_i^C$, $\boldsymbol{h}_i^{Q,0} = \tilde{\boldsymbol{w}}_i^Q$, $1 \leq k \leq K$ and K is the number of RNN layers. We use variational dropout (Kingma et al., 2015) for the input vector to each layer of RNN, i.e. the dropout mask is shared over different timesteps.

Question Understanding. For each question word in $\mathcal{Q}$, we employ one more RNN layer to generate a higher level of understanding of the question.

$$\boldsymbol{h}_1^{Q,K+1}, ..., \boldsymbol{h}_n^{Q,K+1} = \text{RNN}(\boldsymbol{h}_1^Q, ..., \boldsymbol{h}_n^Q), \tag{9}$$

$$\boldsymbol{h}_i^Q = [\boldsymbol{h}_i^{Q,1}; ...; \boldsymbol{h}_i^{Q,K}] \tag{10}$$

Self-Attention on Question. As the question has integrated previous utterances, the model needs to directly relate the previously mentioned concept with the current question for context understanding. Therefore we employ self-attention on question:

$$\{\boldsymbol{u}_i^Q\}_{i=1}^n = \text{Attn}(\{\boldsymbol{h}_i^{Q,K+1}\}_{i=1}^n, \{\boldsymbol{h}_i^{Q,K+1}\}_{i=1}^n, \{\boldsymbol{h}_i^{Q,K+1}\}_{i=1}^n) \tag{11}$$

$\{\boldsymbol{u}_i^Q\}_{i=1}^n$ is the **final representation of question words**.

Multilevel Inter-Attention. After multiple RNN layers extract different levels of semantic abstraction, we conduct inter-attention from question to context based on these representations.

However, the cumulative output dimensions from all previous layers can be very large and computationally inefficient. Here we leverage the history-of-word idea from FusionNet (Huang et al., 2017): the attention uses all previous layers to compute scores, but only linearly combines one RNN layer output.

In detail, we conduct $K+1$ times of multilevel inter-attention from each RNN layer output of question to context $\{\boldsymbol{m}_i^{(k),C}\}_{i=1}^m = \text{Attn}(\{\text{HoW}_i^C\}_{i=1}^m, \{\text{HoW}_i^Q\}_{i=1}^n, \{\boldsymbol{h}_i^{Q,k}\}_{i=1}^n), 1 \le k \le K+1$, where HoW is the history-of-word vector:

$$\text{HoW}_i^C = [\text{GloVe}(w_i^C); \text{BERT}_{w_i^C}; \boldsymbol{h}_i^{C,1}; ..., \boldsymbol{h}_i^{C,k}], \tag{12}$$

An additional RNN layer is added to context $\mathcal{C}$:

$$\boldsymbol{y}_i^C = [\boldsymbol{h}_i^{C,1}; ...; \boldsymbol{h}_i^{C,k}; \boldsymbol{m}_i^{(1),C}; ...; \boldsymbol{m}_i^{(K+1),C}], \tag{13}$$

$$\boldsymbol{v}_1^C, ..., \boldsymbol{v}_m^C = \text{RNN}(\boldsymbol{y}_1^C, ..., \boldsymbol{y}_m^C) \tag{14}$$

Self Attention on the Context. Similar to questions, SDNet applies self-attention to the context. Again, it uses the history-of-word concept to reduce the output dimensionality:

$$\boldsymbol{s}_i^C = [\text{HoW}_i^C; \boldsymbol{m}_i^{(1),Q}; ...; \boldsymbol{m}_i^{(K+1),Q}; \boldsymbol{v}_i^C], \tag{15}$$

$$\{\tilde{\boldsymbol{v}}_i^C\}_{i=1}^m = \text{Attn}(\{\boldsymbol{s}_i^C\}_{i=1}^m, \{\boldsymbol{s}_i^C\}_{i=1}^m, \{\boldsymbol{v}_i^C\}_{i=1}^m). \tag{16}$$

The self-attention is followed by an additional RNN layer to generate the **final representation of context words**:

$$\{\boldsymbol{u}_i^C\}_{i=1}^m = \text{RNN}([\boldsymbol{v}_1^C; \tilde{\boldsymbol{v}}_1^C], ..., [\boldsymbol{v}_m^C; \tilde{\boldsymbol{v}}_m^C]) \tag{17}$$

2.5 Output Layer

Question Condensation. The question is condensed into a single representation vector:

$$\boldsymbol{u}^Q = \sum_i \beta_i \boldsymbol{u}_i^Q, \tag{18}$$

$$\beta_i \propto \exp(\boldsymbol{w}^T \boldsymbol{u}_i^Q), \tag{19}$$

where $\boldsymbol{w}$ is a trainable vector.

Generating answer span. As SDNet outputs answers of interval forms, the output layer generates the probability that the answer starts and ends at the i-th context word, $1 \le i \le m$:

$$P_i^S \propto \exp((\boldsymbol{u}^Q)^T W_S \boldsymbol{u}_i^C), \tag{20}$$

$$\boldsymbol{t}^Q = \text{GRU}(\boldsymbol{u}^Q, \sum_i P_i^S \boldsymbol{u}_i^C), \tag{21}$$

$$P_i^E \propto \exp((\boldsymbol{t}^Q)^T W_E \boldsymbol{u}_i^C), \tag{22}$$

where W_S, W_E are parameters. The use of GRU is to transfer information from start position to end position computation.

Table 1: Domain distribution in CoQA dataset.

Domain	#Passage	#QA turn
Child Story	750	14.0
Literature	1,815	15.6
Mid/High Sc.	1,911	15.0
News	1,902	15.1
Wikipedia	1,821	15.4
Out of domain		
Science	100	15.3
Reddit	100	16.6
Total	8,399	15.2

Special answer types. SDNet can also output special types of answer, such as affirmation "yes", negation "no" or no answer "unknown". We separately generate the probabilities of these three answers: P_Y, P_N, P_U. For instance, the probability that the answer is "yes", P_Y, is computed as:

$$P_i^Y \propto \exp\left((\boldsymbol{u}^Q)^T W_Y \boldsymbol{u}_i^C\right) \tag{23}$$

$$P_Y = (\sum_i P_i^Y \boldsymbol{u}_i^C)^T \boldsymbol{w}_Y \tag{24}$$

where W_Y and $\boldsymbol{w}_Y$ are parametrized matrix and vector, respectively.

2.6 Training and Inference

During training, all rounds of questions and answers for the same passage form a batch. The goal is to maximize the probability of the ground-truth answer, including span start/end position, affirmation, negation and no-answer situations. Therefore, we minimize the cross-entropy loss function $\mathcal{L}$:

$$\mathcal{L} = -\sum_k I_k^S(\log(P_{i_k^s}^S) + \log(P_{i_k^e}^E)) + I_k^Y \log P_k^Y + I_k^N \log P_k^N + I_k^U \log P_k^U, \tag{25}$$

where i_k^s and i_k^e are the ground-truth span start and end position for the k-th question. $I_k^S, I_k^Y, I_k^N, I_k^U$ indicate whether the k-th ground-truth answer is a passage span, "yes", "no" and "unknown", respectively.

During inference, we pick the largest span/yes/no/unknown probability. The span is constrained to have a maximum length of 15.

3 Experiments

We evaluated our model on CoQA (Reddy et al., 2018), a large-scale conversational question answering dataset. In CoQA, many questions require understanding of both the passage and previous rounds of questions and answers, which poses challenge to conventional machine reading models. Table 1 summarizes the domain distribution in CoQA. As shown, CoQA contains passages from multiple domains, and the average number of question answering turns is more than 15 per passage.

For each in-domain dataset, 100 passages are in the development set, and 100 passages are in the test set. The rest in-domain dataset are in the training set. The test set also includes all of the out-of-domain passages.

3.1 Implementation Details

We use spaCy for word tokenization and employ the uncased BERT-large model to generate contextual embedding.

During training, we use a dropout rate of 0.4 for BERT layer outputs and 0.3 for other layers. We use Adamax (Kingma & Ba, 2014) as the optimizer, with a learning rate of $\alpha = 0.002, \beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. We train the model for 30 epochs. The gradient is clipped at 10.

Table 2: Model and human performance (% in F1 score) on the CoQA test set.

	Child.	Liter.	Mid-High.	News	Wiki	Reddit	Science	Overall
PGNet	49.0	43.3	47.5	47.5	45.1	38.6	38.1	44.1
DrQA	46.7	53.9	54.1	57.8	59.4	45.0	51.0	52.6
DrQA+PGNet	64.2	63.7	67.1	68.3	71.4	57.8	63.1	65.1
BiDAF++	66.5	65.7	70.2	71.6	72.6	60.8	67.1	67.8
FlowQA	73.7	71.6	76.8	79.0	80.2	67.8	76.1	75.0
SDNet (single)	**75.4**	**73.9**	**77.1**	**80.3**	**83.1**	**69.8**	**76.8**	**76.6**
SDNet (ensemble)	**78.7**	**77.1**	**80.2**	**81.9**	**85.2**	**72.3**	**79.7**	**79.3**
Human	90.2	88.4	89.8	88.6	89.9	86.7	88.1	88.8

The word-level attention has a hidden size of 300. The self attention layer for question words has a hidden size of 300. The RNNs for question and context have $K = 2$ layers and each layer has a hidden size of 125. The multilevel attention from question to context has a hidden size of 250. The self attention layer for context has a hidden size of 250. The final RNN layer for context words has a hidden size of 125.

3.2 Baseline Models and Results

We compare SDNet[2] with the following baseline models: DrQA+PGNet (Reddy et al., 2018), BiDAF++ (Yatskar, 2018) and FlowQA (Huang et al., 2018). Aligned with the official leaderboard, we use F_1 as the evaluation metric, which is the harmonic mean of precision and recall at word level between the predicted answer and ground truth.[3]

Table 2 shows the performance of SDNet and baseline models.[4] As shown, SDNet achieves significantly better results than baseline models. In detail, the single SDNet model improves overall F_1 by 1.6%, compared with previous state-of-art model on CoQA, FlowQA. We also trained an ensemble model consisting of 12 SDNet models with the same structure but different random seeds for initialization. The ensemble model uses the answer from the most number of models as its predicted answer. Ensemble SDNet model further improves overall F_1 score by 2.7%.

Figure 3 shows the F_1 score of SDNet on development set during training. As seen, SDNet overpasses all but one baseline models after the second epoch, and achieves state-of-the-art results after 8 epochs.

Ablation Studies. We conduct ablation studies on SDNet to verify the effectiveness of different parts of the model. As Table 3 shows, our proposed weighted sum of per-layer output from BERT is crucial, boosting the performance by 1.75% compared with the canonical method of using only the last layer's output. This shows that the output from each layer in BERT is useful in downstream tasks. Using BERT-base instead of the BERT-large pretrained model hurts the F_1 score by 2.61%.Variational dropout and self attention can each improve the performance by 0.24% and 0.75% respectively.

Contextual history. In SDNet, we utilize conversation history via prepending the current question with previous N rounds of questions and ground-truth answers. We experimented with the effect of N and present the result in Table 4.

As shown, excluding dialogue history ($N = 0$) can reduce the F_1 score by as much as 8.56%, manifesting the importance of contextual information in conversational QA task. The performance of our model peaks when $N = 2$, which was used in the final SDNet model.

[2]We have open-sourced our SDNet code. For anonymity, we will share the link afterwards.

[3]According to official evaluation of CoQA, when there are more than one ground-truth answers, the final score is the average of max F_1 against all-but-one ground-truth answers.

[4]Result was taken from official CoQA leaderboard on Nov. 30, 2018.

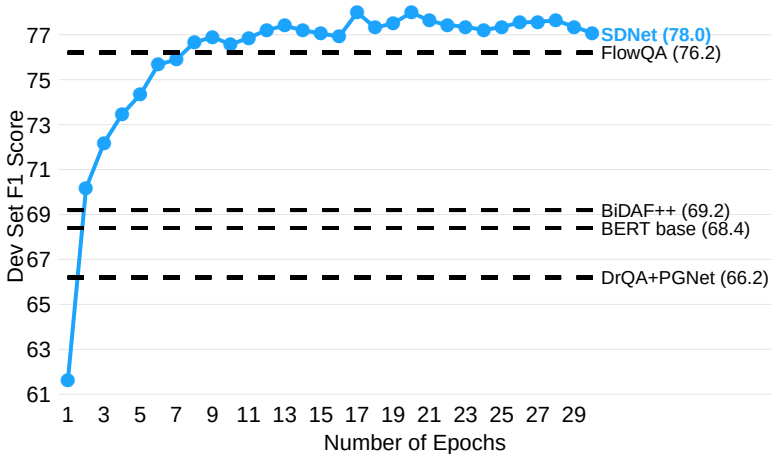

Figure 3: F_1 score on CoQA dev set over training epochs. Note that for BERT-base model, we use the number on test set from the leaderboard.

Table 3: Ablation study of SDNet on CoQA development dataset.

Model	F_1
SDNet	77.99
–Variational dropout	77.75
–Question self attention	77.24
Using last layer of BERT output (no weighted sum)	76.24
BERT-base	75.38

Table 4: Performance of SDNet on development set when prepending different number of rounds of history questions and answers to the question. The model uses BERT-Large contextual embedding and fixes BERT's weights.

#previous QA rounds N	F_1
0	69.43
1	76.70
2	**77.99**
3	77.39

4 Conclusions

In this paper, we propose a novel contextual attention-based deep neural network, SDNet, to tackle the conversational question answering task. By leveraging inter-attention and self-attention on passage and conversation history, the model is able to comprehend dialogue flow and the passage. Furthermore, we leverage the latest breakthrough in NLP, BERT, as a contextual embedder. We design the alignment of tokenizers, linear combination and weight-locking techniques to adapt BERT into our model in a computation-efficient way. SDNet achieves superior results over previous approaches. On the public dataset CoQA, SDNet outperforms previous state-of-the-art model by 1.6% in overall F_1 score and the ensemble model further improves the F_1 by 2.7%.

Our future work is to apply this model to open-domain multiturn QA problem with large corpus or knowledge base, where the target passage may not be directly available. This will be a more realistic setting to human question answering.

REFERENCES

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

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

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

Ahmed Elgohary, Chen Zhao, and Jordan Boyd-Graber. A dataset and baselines for sequential open-domain question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1077–1083, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *arXiv preprint arXiv:1711.07341*, 2017.

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*, 2017.

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

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. pp. 5998–6008, 2017.

Mark Yatskar. A qualitative comparison of coqa, squad 2.0 and quac. *arXiv preprint arXiv:1809.10735*, 2018.