Attention-Based Neural Network For Question Answering

Zhengyang Tang[*]
tangzhy@stanford.edu

Songze Li[†]
songzeli@stanford.edu

Codalab username: tangzhy Songze

Abstract

There has been significant recent progress in machine comprehension (MC) and question answering (QA). Typically, these methods improve both the way we capture context-query interactions and the computational efficiency. In this paper, we did a reimplementation across multiple attention mechanisms, including Bidirectional attention flow (BiDAF), Coattention, and Self-attention. We also explored a novel neural architectures, namely BiCoattention Network (BCN), which combined BiDAF and Coattention components. We also implemented the R-NET model to study the mechanism of self-attention. On Stanford Question Answering Dataset (SQuAD), our best single model achieved an F1 score of 72.05 and an EM score of 61.93.

1 Introduction

Question answering (QA) is a crucial task in natural language processing. In this task, a computer system is required to answer a query about a given context paragraph automatically. In the past few years, question answering has been gaining popularity and achieving promising results on a variety of datasets. One of the key factors to the advancement is the use of neural attention mechanisms, which help modeling more complex interactions between the queries and contexts as well as allow for more parallelization in the computation.

Rencently with the release of the Stanford Question Answering Dataset (SQuAD) by Rajpurkar et al. (2016) [1], question answering has been driven forward further. The SQuAD consists of 107,785 question-answer pairs on 536 Wikipedia articles. Seo et at. (2017) [2] proposes the Bi-Directional Attention Flow (BiDAF) network, and unlike previous work, this model drops the way of summarizing the context paragraph into a fixed-sized vector by creating an similarity matrix of upstream representations. Their best model achieves an F1 score of 77.3%. Xiong et al. (2017) [3] creates Dynamic Coattention Network (DCN). The DCN shares the similar way of building an affinity matrix of upstream representation with the BiDAF but differs in the function of capturing interaction between the query and the context. The best DCN achieves an F1 score of 75.9%. Based on Wang&Jiang(2016) [4], Microsoft Research Asia (2017)[5] also introduced a new model called R-NET.

Inspired by the above papers, we conduct a reimplementation of BiDAF and extend it to a novel neural architecture, namely BiCoattention Network (BCN). The BCN inherits the way of creating the affinity matrix like DCN, but integrates the functions of modeling interactions between queries and context from both the BiDAF and DCN . In addition to those, we also explore the self-attention in R-NET and compared these two models' capability in solving question answering.

[*] The part of BiDAF and BCN and analysis work is done by ZT

[†] The part of r-net and related work is done by SL

2 Models

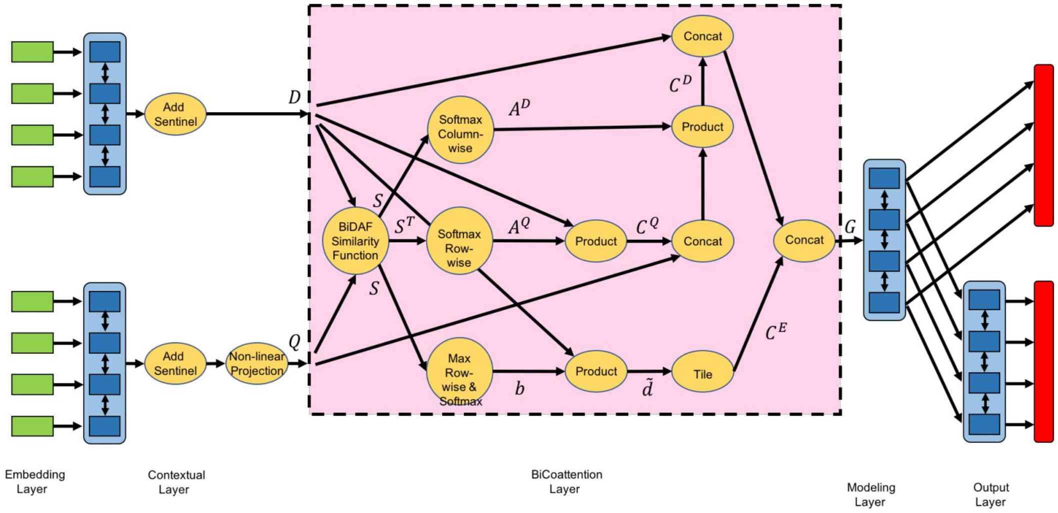


Figure 1: The BCN architecture

2.1 BCN

The BCN model is a hierarchical multi-stage process and consists of the following five modules (Figure 1): (1)

1. **Embedding Layer** maps each word to a vector space using pre-trained word embeddings.
2. **Contextual Layer** refines the word embeddings by utilizing contextual information across the surrounding words.
3. **BiCoattention Layer** combines the attention mechanisms from BiDAF and DCN and produces a better version of query-aware context representations.
4. **Modeling Layer** operates a classical RNN layer on the upstream representation.
5. **Output Layer** yields the start and end positions of answer span to the query.

2.1.1 Embedding Layer

In the embedding layer, we used the pre-trained GloVe vectors to represent the words for the context and question. Given the complexity of matrix computation, we finalized the embedding size to 100. Besides, the word embeddings were all keeped static during the training process out of the consideration for better generalization on dev and test sets.

2.1.2 Contextual Layer

We denote the sequence of word vectors in the context document as $x_1^D, x_2^D, ..., x_m^D \in \mathbb{R}^d$, and denote the same for the query as $x_1^Q, x_2^Q, ..., x_n^Q \in \mathbb{R}^d$. These embeddings are fed into a 1-layer bidirectional GRU and we concatenate the forward and backward hidden states to obtain the context and the question hidden states respectively. Then we add a sentinel vector to the context document matrix and define it as $D = [x_1^D \ \ldots \ x_m^D \ x_\emptyset^D] \in \mathbb{R}^{2d\times(m+1)}$, where the sentinel vector's role is to allow the model to not attend to any particular word in the input. The query matrix is created in the similar way with a sentinel vector but introduced an additional non-linear projection layer on top of the encoding. This could allow for more variation between the query encoding space and the context encoding space. More specifically, we define an intermediate query representation $Q' = [x_1^Q \ \ldots \ x_n^Q \ x_\emptyset^Q] \in \mathbb{R}^{2d\times(n+1)}$ and the final query matrix comes from $Q = \tanh{(W^Q Q' + b^Q)} \in \mathbb{R}^{2d\times(n+1)}$.

2.1.3 BiCoattention Layer

The BiCoattention layer is responsible for linking and fusing information from the context document D and the query Q. In this layer, the attentions are computed based on a shared similarity matrix in two directions: from context to query as well as from query to context. Inspired by Seo et al. (2017) [2] and Xiong et a. (2017) [3], we conduct an integration of the two attention mechanisms and create a more complex similarity matrix.

Similarity Matrix First, the similarity matrx $S \in \mathbb{R}^{(m+1)\times(n+1)}$ is computed by:

$$S_{ij} = \alpha(D_{:i}, Q_{:j}) \in \mathbb{R}$$

where S_{ij} indicates the similarity between i-th context word and j-th query word. α is a trainable scalar function encoding the similarity of the couple input, $D_{:i}$ is the i-th column vector of D, and $Q_{:j}$ is the j-th column vector of Q. Similar to Seo et al. (2017) [2], we choose $\alpha(d, q) = W^T_{(S)}[d; q; d \circ q]$, where $W_{(S)} \in \mathbb{R}^{6d}$ is a trainable vector, $\circ$ is the elementwise multiplication, $[;]$ stands for vector concatenation across row. And finally we get the shared similarity matrix S to compute bidirectional attentions.

BiCoattention Next, the similarity matrix is normalized row-wise and column-wise separately in order to produce the attention weights A^Q which scans the context word for each word in the query, and A^D which scans the same for each word in the context.

$$A^Q = softmax(S) \in \mathbb{R}^{(m+1)\times(n+1)}$$

$$A^D = softmax(S^T) \in \mathbb{R}^{(n+1)\times(m+1)}$$

Then we compute the summaries of context in light of the words in the query:

$$C^Q = DA^Q \in \mathbb{R}^{2d\times(n+1)}$$

we compute the summaries of query in light of the words in the context by QA^D. Meanwhile we could also derive the summaries of previous attention C^Q in light of the words in the context. Since these operations could be done in parallel,, we perform them in:

$$C^D = [Q; C^Q]A^D \in \mathbb{R}^{4d\times(m+1)}$$

where C^D is actually a concatenation of query awared and query-to-context summaries awared context representations. It plays the important role as a co-dependent representation in capturing the interaction of the query and the context.

Besides, we also obtain the attention weights for query to context words through a max pooling mechanism by $b = softmax(max_{row-wise}(S)) \in \mathbb{R}^{m+1}$, where the maximum function $max_{row-wise}$ is performed in row-wise convention. Then the attended context vecotr is $\tilde{d} = \sum_i b_i D_{:i} \in \mathbb{R}^{2d}$. $\tilde{d}$ indicates the weighted sum of the most important words in the context with respect to the query, and we simply tile the $\tilde{d}$ for $m + 1$ times to get the $C^E \in \mathbb{R}^{2d\times(m+1)}$.

Finally, the contextual embeddings, the co-dependent context representation and the attended context vector are combined together to yield G. We define G by:

$$G_{:i} = \beta(D_{:i}, C^D_{:i}, C^E_{:i}) \in \mathbb{R}^{d_G}, \forall i \in \{1, \ldots, m\}$$

where $G_{:i}$ is the i-th column vector corresponding to the i-th context word, β is a function fusing the three input vectors, and d_G is the output dimension of the β function. Here, we use the simple element-wise multiplication and concatenation function as following: $\beta(D, C^D, C^E) = [D; C^D; [D; D] \circ C^D; D \circ C^E] \in \mathbb{R}^{12d\times m}$

2.1.4 Modeling Layer

The input to the modeling layer is G. Here we employ 2 layers of bidirectional GRU whose hidden size is d for each direction. Thus this layer give us a higher level representation matrix $M \in \mathbb{R}^{2d\times m}$, which models the interaction within the upstream context representation itself, and therefore provides us informative contextual information with respect to the entire context and the query.

2.1.5 Output Layer

At last, we are required to predict the span in the context to answer the query. Similar to the BiDAF, we select out the span by predicting the start and the end position of the phrase in the context. For the start position, we obtain its probability distribution by:

$$p^1 = softmax(W_{p^1}^T [G; M])$$

where $W_{p^1}^T \in \mathbb{R}^{14d}$ is a trainable weight vector. For the end position, we employ another bidirectional GRU layer on the M and get the $M^2 \in \mathbb{R}^{2d \times m}$. Then we obtain the end position's probability distribution by:

$$p^2 = softmax(W_{p^2}^T [G; M^2])$$

where again $W_{p^2}^T \in \mathbb{R}^{14d}$ is a trainable weight vector, and in this way we enable the end position predictions to be conditioned on the start position predictions.

2.2 R-NET

The main idea of R-NET model is mainly about attention. There are mainly four parts:(1)the recurrent network encoder to build representation for questions and passages separately,(2)the gated matching layer to match the question and passage,(3)the self-matching layer to aggregate information from the whole passage, and(4) the pointer-network based answer boundary prediction layer. My implementation of R-NET model referenced from GitHub[7].More detail and the contribution of these parts are followed:

2.2.1 Attention-Based recurrent network

This gated attention-based recurrent network is used to acount for the fact that words in the passage are of different importance to answer a particular question for reading comprehension and question answering.The gated attention-based recurrent network assigns different levels of importance to passage parts depending on their relevance to the question, masking out irrelevant passage parts and emphasizing the important ones.This additional gate is based on the current passage word and its attention-pooling of the question, which focuses on the relation between the question and current passage word.

2.2.2 Self-Matching Attention mechanism

The self-matching mechanism is used to aggregate evidence from the whle passage to infer the answer. Through a gated matching layer, the resulting question-aware passage represention effectively encodes question informatin for each passage word. Since the natural problem of recurrent network which could only memorize limited passage context in practice despite its theoretical capability. To address this problem, self-matching layer to used to dynamically refine passage representation based on the whole passage.

2.2.3 Output Layer

Output Layer incorporate pointer networks to predict the start and end position of the answer. The pointer network is used to select the start position and end position from the passage. In the output layer, attention-pooling is also used again as the initial state of the answer recurrent network.

2.3 Training Details

For the padding strategy, as can be seen in Figure 2, over 98% contexts have the length at most 289 while over 99.9% queries have the length at most 29 in the training set, and for the sake of efficiency and performance, we set the maximum length of the context and query to be 400 and 30. For the Optimizer, we employ the Adam optimizer with initial learning rate 0.001. We also try to vary the learning rate over the course of training by $lr = d \cdot min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$, where the d and $warmup\_steps$ are the hyper-parameter that need setting. But it turns out that the BCN model is pretty sensitive to the change of learning rate, and that doesn't help. For avoiding overfitting, we apply dropout rate with both 0.2. The tuned parameter of out R-NET model is the same as the parameter discussed above.

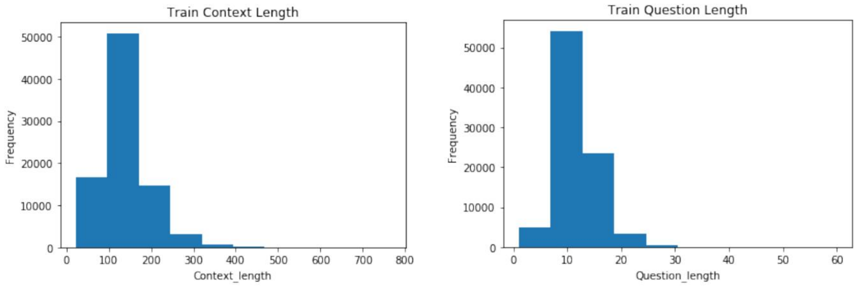


Figure 2: Histogram plots of the lengths of the context and query in the training data

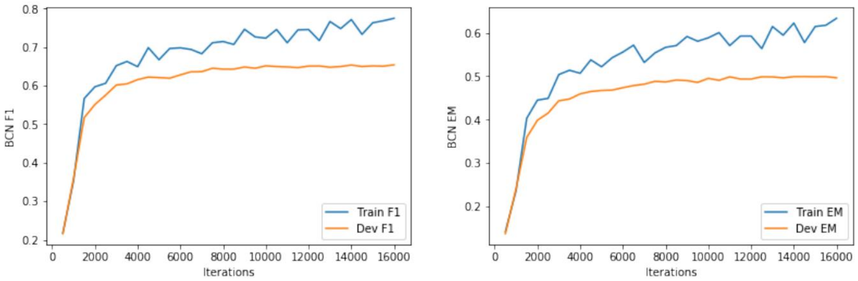


Figure 3: BCN performance during training

3 Results and Analysis

We achieved F1 score 71.21 and EM 60.28 of the BCN and F1 score 72.05 and EM 61.93 of the R-NET. The BCN's training process is showed in Figure 3, where the plots show that even we conduct a coarse search of hyper-parameter, the overfitting problem still exists and largely affects the BCN performance. The detailed comparision with state-of-the-art models can be found in Table 1, and we conjecture the remaining gap might be resulted from the lack of character level embeddings.

3.1 Model Ablations

We analyze the performance of BCN and its ablations on the SQuAD dev set as can be seen in Table 2. The contextual layer does not contribute to the BCN but make it worse. Since the contextual layer envolves much complex non-linear transformation and sentinel vectors, we conjecture current model

Model	F1 Score	EM	F1 Score	EM
	Dev Set		Test Set	
Baseline(our implementation, single)	43.66	34.03	-	-
BiDAF(our implementation, single)	70.94	60.55	-	-
BCN(our implementation, single)	71.21	60.28	71.37	60.97
R-NET(our implementation, single)	72.05	61.93	-	-
BiDAF(reference implementation, single)	77.30	67.70	77.30	68.00
DCN(reference implementation, single)	75.60	65.40	75.90	66.20
R-NET(reference implementation, single)	77.50	68.40	-	-

Table 1: Performance comparision with other methods

BCN's Different Modules	F1 Score	EM
	Dev Set	
Baseline	43.66	34.03
Baseline + Contextual Encoder	36.03	27.81
Baseline + Contextual Encoder + Modeling and Output Layer	67.40	56.40
Baseline + Contextual Encoder + Modeling and Output Layer + BiCoattention	71.21	60.28

Table 2: BCN ablations on the dev set

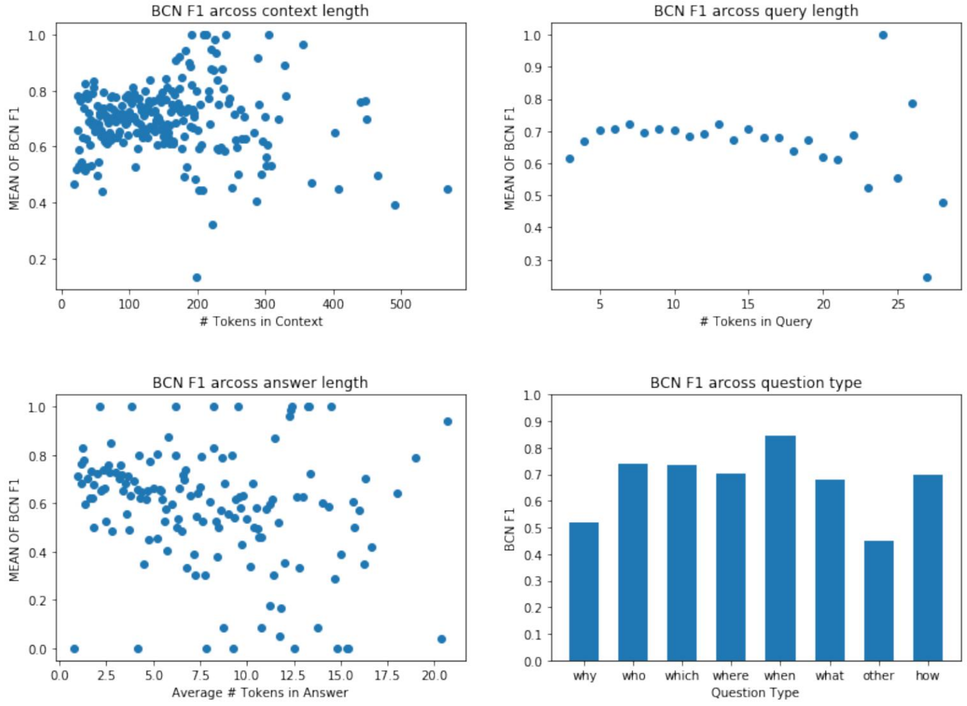


Figure 4: BCN performance analysis across length and query type

architecture does not well solve the gradient back propagation problem, which makes it harder for the contextual layer to update in the correct directions. We also study the role of the modeling and output layer, and the result is pretty promising. The 2 layers GRU and methods of conditioning end prediction on start prediction improve the F1 score greatly by 54% from our baseline. This is a huge step which proves their strong ability in capturing interactions within upstream context representations. Finally, we replace the basic attention mechanism with our bicoattention module, and this boost the BCN model performance by around 6%.

3.2 Performance Across Length and Query Type

We also conduct an analysis on the performance of the BCN and R-NET with respect to length variance. As can be seen in Figure 4, there is no notable performance deterioration with the change of context and query length. This indicates that our bicoattention module is effective in capturing global interactions between the context and the query correctly as well as selecting out the most relevant phrases while ignoring the rest of the context. As for the average length of the answer, the BCN performs a little worse when the average length goes longer, which is also meet our expectation, because normally the longer the answer is, the more challenging it can be answered. Besides, we study the performance of BCN across the question types as well. The statistics shows that our model is adept at the 'when' question but struggles with the 'why' questions. We conjecture this is

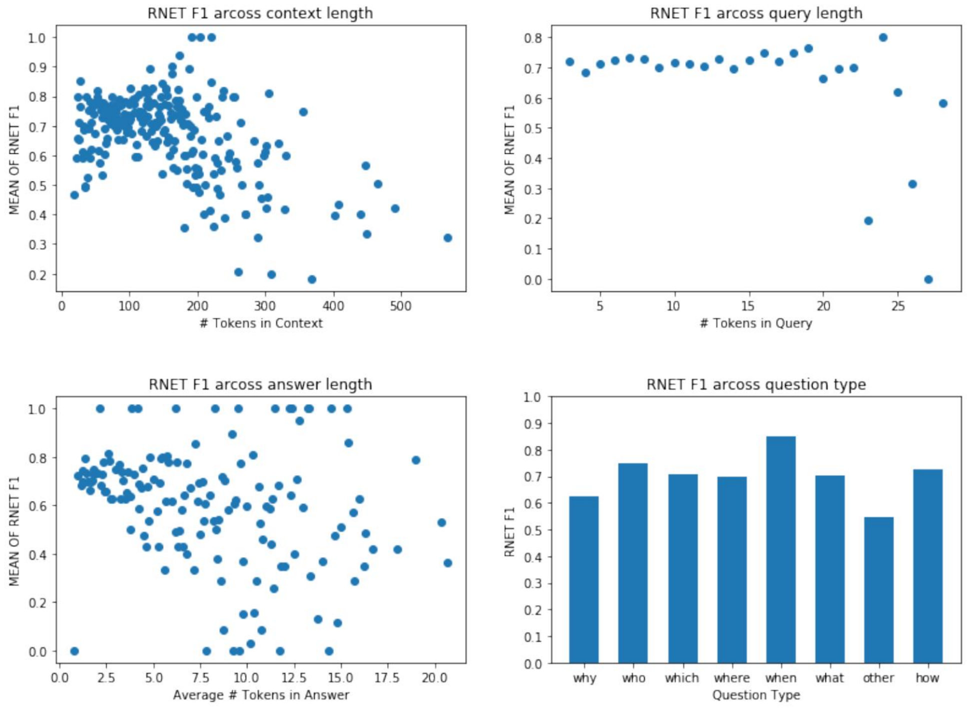


Figure 5: R-NET performance analysis across length and query type

also consistent with the natural difficulty in answering these two question types since the answer to the 'why' might need more thinking.

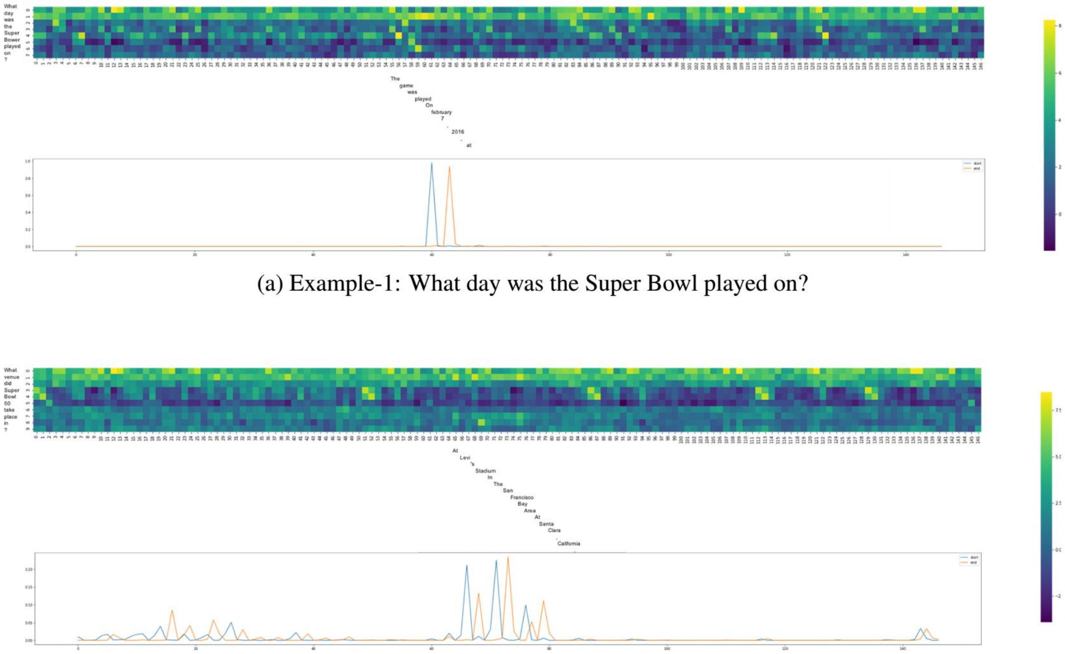
Figure 5 shows the R-NET performance across length and query type. Compared to the BCN model in Figure 4, it is interesting to notice that f1 score for large context and answers tokens of BCN is greater than R-NET. The better phenomenon possibly could be attribute to the BiCoattention Layer, which is responsible for linking and fusing information from the context and query. However, it is also interesting to note that the f1 score of R-NET for ”WHY” style question is greater than that of BCN model. It might be the result the self-attention layer in the context of R-NET, since the ”WHY” style question is more needed to infer from the context document, which is job self-matching layer does in R-NET.

3.3 Visualizations

Examples of the BCN predictions:

- **Context**: “The game was played on February 7, 2016, at Levi’s Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl . . . ”
- **Question-1**: “What day was the Super Bowl played on?”
- **Prediction-1**: “February 7, 2016”
- **Answer-1**: [“February 7, 2016”, “February 7”, “February 7, 2016”]
- **Question-2**: “What venue did Super Bowl 50 take place in?”
- **Prediction-2**: “San Francisco Bay”
- **Answer-2**: [“Levi’s Stadium”, “Levi’s Stadium”, “Levi’s Stadium in the San Francisco Bay Area at Santa Clara”]

In this subsection, we visualize the BCN attention matrices (zoom in to see details in Figure 6 6) from some question-context tuples in the dev set, where the attention is obtained directly from the



(a) Example-1: What day was the Super Bowl played on?

(b) Example-2: What venue did Super Bowl 50 take place in?

Figure 6: BCN bicoattention visualization aligned with logits

bicoattention module. In the first example, the day in the query matches the date in the context perfectly and makes it easy for output layer to predict both the start and end position. While in the second example, the $What$ in the query obtains a relatively noisier attention to the context. Despite that, the distributions of start position and end position probability actually find the relevant span, and select out part of the answer correctly.

3.4 Error Analysis

Moreover, we also try to understand the BCN's prediction error pattern through a manual analysis. In this subsection, we randomly select 50 incorrect BCN predicted cases based on EM and category them into 6 classes, which is similar to Seo et at. (2017) [2]. But we replace the incorrect preprocessing with padding problems after scanning the samples. Our manual analysis shows that 46% errors are caused by imprecise answer boundaries, 30% involve syntactic complications and ambiguities, 12% are the multiple sentences problem, 6% are resulted from paddding problem, 4% are paraphrase problems and 2% require external knowledge. See Appendix A which is attached in supplementary material for the examples of BCN error analysis.

4 Conclusion

In this paper, we conduct a reimplementation of BiDAF and R-NET, and we introduce the BiCoattention Network (BCN) as an extension which combines the BiDAF and Dynamic Coattention Network core parts. The experimental evaluations show that our model achieves competitive results on the Stanford Question Answering Dataset (SQuAD). We also explored very detailed analysis of BCN from multiple perspectives, and we learn more clearly about what component of BCN is capable of answering complex questions while the others still need improving by compared with R-NET, especially for the attention mechanism. It is worth mentioning that the visualization and error analysis indeed helps us a lot in understanding the BCN's downside, and we are excited about future work such as integrating Transformer Network [6] into BCN.

Acknowledgments

The anthors would like to thank TAs for insightful discussions and timely feedback about improving the models.

References

[1] Rajpurkar, Pranav, 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 , 2383-92.

[2] Seo, Min Joon, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. “Bidirectional Attention Flow for Machine Comprehension” International Conference on Learning Representations.

[3] Xiong, Caiming, Victor Zhong, and Richard Socher. 2017. “Dynamic Coattention Networks For Question Answering.” International Conference on Learning Representations.

[4] Wang, Shuohang, and Jing Jiang. 2017. “Machine Comprehension Using Match-LSTM and Answer Pointer.” International Conference on Learning Representations.

[5] Wang W, Yang N, Wei F, et al. Gated self-matching networks for reading comprehension and question answering[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017, 1: 189-198.

[6] Vaswani, Ashish, et al. “Attention Is All You Need.” Neural Information Processing Systems, 2017, pp. 6000-6010.

[7] https://github.com/unilight/R-NET-in-Tensorflow

A ERROR ANALYSIS

Table 3 summarizes the types of errors by BCN and shows examples for each category of error in SQuAD dev set.

Error type	Ratio (%)	Example
Imprecise answer boundaries	46	**Context**: “Another example of the richness of the zeta function and a glimpse of modern algebraic number theory is the following identity (Basel problem), due to Euler,” **Question**:“Of what mathematical nature is the Basel problem?” **Prediction**: “modern algebraic” **Answer**: “modern algebraic number theory”
Syntactic complications and ambiguities	30	**Context**: “To the east is the Colorado Desert and the Colorado River at the border with Arizona” **Question**:“What is the name of the water body that is found to the east?’ **Prediction**: “colorado desert” **Answer**: “Colorado River”
Paraphrase problems	4	**Context**: “The two forces finally met in the bloody Battle of Lake George between Fort Edward and Fort William Henry. The battle ended inconclusively, with both sides withdrawing from the field.” **Question**:“Who won the battle of Lake George?” **Prediction**: “fort edward and fort william henry” **Answer**: “The battle ended inconclusively”
Multi-sentence	12	**Context**: “Construction projects can suffer from preventable financial problems. Underbids happen when builders ask for too little money to complete the project. ” **Question**:“What is an underbid?” **Prediction**: “construction projects can suffer from preventable financial problems. underbids happen when builders ask for too little money to complete the project” **Answer**: “when builders ask for too little money to complete the project”
Padding problem	6	**Context**: “In an interview with newspaper editor Arthur Brisbane, Tesla said that he did not believe in telepathy,” **Question**:“What was Brisbane’s job?” **Prediction**: “” **Answer**: “newspaper editor”
External knowledge	2	**Context**: “which states that there always exists at least one prime number p with $n < p < 2n - 2$, for any natural number$n > 3$.” **Question**:“How is the prime number p in Bertrand’s postulate expressed mathematically?” **Prediction**: “$2n - 2$” **Answer**: “$n < p < 2n - 2$”

Table 3: BCN error analysis on SQuAD

Table 4 summarizes the types of errors by R-NET and shows examples for each category of error in SQuAD dev set.Compared to the error table of BCN, it is interesting to notice that there is a error reduction for the ”Imprecise answer boundaries” ratio. It might because the pointer-network mechanism of R-NET.

Error type	Ratio (%)	Example
Imprecise answer boundaries	40	**Context**: “Robert Nozick argued that government redistributes wealth by force (usually in the form of taxation), and that the ideal moral society would be one where all individuals are free from force. However??? ” **Question**:“When are inequalities in wealth justified, according to John Rawls?” **Prediction**: “when they improve society as a whole”’ **Answer**: “when they improve society as a whole, including the poorest members”
Syntactic complications and ambiguities	26	**Context**: “To the east is the Colorado Desert and the Colorado River at the border with Arizona” **Question**:“What is the name of the water body that is found to the east?’ **Prediction**: “colorado desert” **Answer**: “Colorado River”
Paraphrase problems	6	**Context**: “In the final years of the apartheid era, parents at white government schools were given the option to convert to a ”semi-private” form called Model C, and many of these schools changed their admissions policies to accept children of other races....” **Question**:“How do academic results in former Model C schools compare to other schools?” **Prediction**: “than government schools formerly reserved for other race groups ” **Answer**: “better”
Multi-sentence	8	**Context**: “Construction projects can suffer from preventable financial problems. Underbids happen when builders ask for too little money to complete the project. ” **Question**:“What is an underbid?” **Prediction**: “construction projects can suffer from preventable financial problems. underbids happen when builders ask for too little money to complete the project” **Answer**: “when builders ask for too little money to complete the project”
Padding problem	18	**Context**: “In an interview with newspaper editor Arthur Brisbane, Tesla said that he did not believe in telepathy,” **Question**:“What was Brisbane’s job?” **Prediction**: “” **Answer**: “newspaper editor”
External knowledge	2	**Context**: “which states that there always exists at least one prime number p with $n < p < 2n - 2$, for any natural number$n > 3$.” **Question**:“How is the prime number p in Bertrand’s postulate expressed mathematically?” **Prediction**: “$2n - 2$” **Answer**: “$n < p < 2n - 2$”

Table 4: R-NET error analysis on SQuAD