ForceReader: a BERT-based Interactive Machine Reading Comprehension Model with Attention Separation

Zheng Chen, Kangjian Wu
School of Information and Software Engineering
University of Electronic Science and Technology of China, Chengdu, China
zchen@uestc.edu.cn, kenjewu@std.uestc.edu.cn

Abstract

The release of BERT revolutionized the development of NLP. Various BERT-based reading comprehension models have been proposed, thus updating the performance ranking of reading comprehension tasks. However, the above BERT-based models inherently employ BERT's combined input method, representing the input question and paragraph as a single packed sequence, without further modification for reading comprehension. This paper makes an in-depth analysis of this input method, proposes a problem of this approach. We call it attention deconcentration. Accordingly, this paper proposes ForceReader, a BERT-based interactive machine reading comprehension model. First, ForceReader proposes a novel solution called the Attention Separation Representation to respond to attention deconcentration. Moreover, starting from the logical nature of reading comprehension tasks, ForceReader adopts Multi-mode Reading, and Interactive Reasoning strategy. For the calculation of attention, ForceReader employs Conditional Background Attention to solve the lack of the overall context semantic after the separation of attention. As an integral model, ForceReader shows a significant improvement in reading comprehension tasks compared to BERT. Moreover, this paper makes detailed visual analyses of the attention and propose strategies accordingly. This may be another argument to the explanations of the attention.

1 Introduction

Reading comprehension can be used as a comprehensive test task to evaluate machine understanding of human language(Chen, 2018). Therefore, it has become a primary research topic in natural language processing. In the past few years, due to new datasets, algorithms, and computing capabilities, deep learning technology has evolved rapidly. Driven by deep learning approaches, machine reading comprehension has made significant breakthroughs.

Hermann et al.(2015) of DeepMind proposed CNN/Daily Mail, which known as one of the first large-scale supervised training datasets for machine reading comprehension. They also proposed The Attentive Reader(Hermann et al., 2015), an attention-based Long Short-Term Memory network(Hochreiter and Schmidhuber, 1997) model for reading comprehension. In order to overcome the shortage of CNN/Daily Mail dataset, researchers such as Rajpurkar proposed the SQuAD1.0(Rajpurkar et al., 2016) dataset in 2016, and SQuAD2.0(Rajpurkar et al., 2018) dataset in 2019. These two datasets have rapidly driven a series of studies on deep learning-based machine reading comprehension, resulting in many efficient and ingenious models. Chen et al.(2016) proposed the Stanford Attentive Reader. This end-to-end reading comprehension model combines multi granular language knowledge and attentional mechanisms, become an early representative of neural machine reading comprehension models. Minjoon et al.(2016) proposed a bidirectional attention flow reading model, BiDAF. Based on the BiLSTM backbone network that inherits from Stanford Attentive Reader, BiDAF novelly proposes a bi-directional attention interaction, which is not only from query to context but also from context to query. QANET proposed by Yu et al.(2018) in 2018, adopting a completely different approach. QANET is no longer base on BiLSTM but uses a

Proceedings of the 28th International Conference on Computational Linguistics, pages 2676–2686
Barcelona, Spain (Online), December 8-13, 2020

multi-layer convolution and self-attention mechanism as the encoding blocks, which is similar to the Transformer(Vaswani et al., 2017) that appeared later.

The introduction of BERT in 2018(Devlin et al., 2019) pushed the whole NLP society to a new height. The reading comprehension task also achieved a significant breakthrough, reaching the human-level performance on the SQuAD and other datasets. BERT is entirely based on the self-attention mechanism of the Transformer stacking structure. However, when dealing with the reading comprehension task, it concat the question and paragraph into a single sequence. Although very simple and effective, we argue that this approach may cause attention deconcentration. Despite all the arguments about the interpretability of attention(Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Serrano and Smith, 2019), we still start from the visual distribution of attention, propose an interactive reading comprehension model based on the idea of separate representation of the attention. Specifically, our main contributions in this paper are as follows.

1. We conduct a detailed analysis of the current use of BERT in machine reading comprehension, and propose, visualize, and explain the attention deconcentration problem.

2. We propose novel approaches including Attention Separate Representation, Multi-mode Reading, Conditional Background Attention, and Interactive Reasoning for the attentional deconcentration problem.

3. We perform detailed experiments and comparisons for the improvements proposed in this paper. According to the experiment results, our model obtains a significant improvement compared to the BERT.

4. We also conduct detailed visual analyses. By illustrating the attention of our model, we visually demonstrated the learning ability of our model. Besides, it is also an example of the Transformer's interpretability.

2 Attention Deconcentration

Machine reading comprehension requires a machine to answer question Q based on a given paragraph P. BERT handles this task by encoding the Q and P into a single sequence of words as the input. Then, it performs the classification task only on the output fragment corresponding to the context. Although BERT shows excellent performance, we argue that there are problems with this approach.

First, based on the common sense of reading comprehension, in order to effectively reason, we need to have an accurate understanding of the question and paragraph's semantics. However, BERT's joint-input method may let the semantic of one section affected by the words of the other. These interfering words are often not helpful for understanding. Second, it is difficult for the self-attentive mechanism to accurately distinguish the question and paragraph pair($< Q, P >$) in the joint input sequence when dealing with question and paragraph interactions, thereby establishing the appropriate bi-directional attention between words of question and paragraph.

To better understand these problems, we give a brief example of question-paragraph-answer triple for further discussion as follows.

Q: I want to ask you a question, who is the founder of Microsoft?
P: I know from the book that Bill Gates founded Microsoft. (1)
A: Bill Gates

In the example1, we can easily tell that when we try to understand the Q, we only need to pay attention to few words such as *who*, *founder*, and *Microsoft*. Words like *I*, *know*, *book*, in P do not play big roles in this particular Q&A scenario. Therefore, a basic understanding is that when modeling this particular $< Q, P >$ pair, the reading comprehension model should pay more attention to the more important words, while less critical words do not need much attention. However, the self-attention mechanism, finetuned

from the BERT pre-trained model, will always assign attention to all words in P. For the second problem, we can see that when reasoning interactively with the $< Q, P >$ pair, word like *founder* in Q should pay more attention to the words*Bill*, *Gates*, *founded* and *Microsoft* in P, than to the words *you*, *question*, *is* and *the* in it's own section. However, in the present BERT based approaches, attention is always given to these words. We call this problem attention deconcentration. Compare with the Masked Language Modelling, and Next Sentence Prediction task in BERT pre-training, the supervised training of the Machine Reading Comprehension task is extremely insufficient. Hence, we think it is impossible to eliminate attention deconcentration through adequate training.

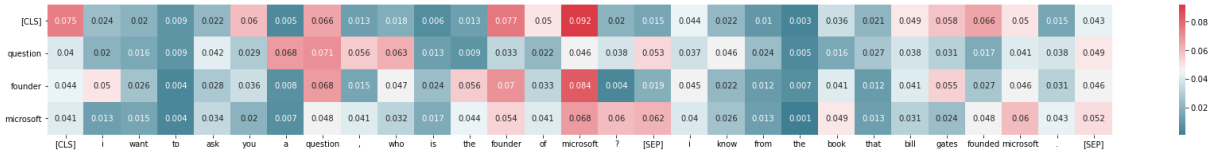

Figure 1: Visualization of the attention deconcentration problem

In Fig.1, we visualize the $< Q, P >$ pair of the example1. We can intuitively see the attention deconcentration phenomenon we discussed. Besides, through more experiments, we find out the longer the input sequence, the more attention deconcentration observed.

3 ForceReader Model

We give the overall structure of the ForceReader model, which we proposed, in Fig.2, including Attention Separate Representation, Multi-mode Reading, Conditional Background Attention, and Interactive Reasoning.

3.1 Attention Separate Representation

To address the problem caused by attentional deconcentration, we adopt Attention Separate Representation. In this approach, we input Q and P to BERT, separately. That is, the Transformer model have to compute attention on Q and P, respectively, without inter-attention. In this way, the overall semantic attention of the Q will only distribute over its own words, but not distracted by certain perturbed words in the P. This model makes it easier to capture the semantic core words of a text section, such as *who*, *founder* and *Microsoft*. Specifically, for a particular word, *founder* in Q, for example, it is easier to get the relevant semantic qualifier, *Microsoft*, so that the semantic richness of the interaction can better be matched in later interactions.

3.2 Multi-mode Reading

When people do reading comprehension tasks, they tend to have different modes. Somebody may read the question before reading the paragraph and then find the answer from the paragraph with the background knowledge of the question, which we call it the $Q2P$ mode. Other people may read the paragraph before reading the question and then answer the question with a memory of the paragraph. We call it the $P2Q$ mode. Another way to use paragraphs and questions is to read them together and to perform comparative reasoning. We name it the QCP mode. The $Q2P$ mode has a hierarchical structure in the paragraph. When the answer to a question is in a particular paragraph of the paragraph, the knowledge required for the answer is better located. The $P2Q$ mode makes the question to be better understand. It is suitable for the complicated questions which need paragraph as the context to understand. The QCP mode is for the cases that require reasoning from multiple parts of the paragraph to grasp the central idea of the entire paragraph. In summary, the three modes of solving reading comprehension are different and can be used to unique advantage in different problem contexts. Therefore, our model will combine these three reading modes, enabling knowledge to be acquired through multiple modes to address reading comprehension in different contexts.

Before performing the Multi-mode Reading, we first follow the Attention Separate Representation to get the vector representation of the Q and the P, which are$\boldsymbol{H_Q}$ and $\boldsymbol{H_P}$. In the $Q2P$ mode, for each

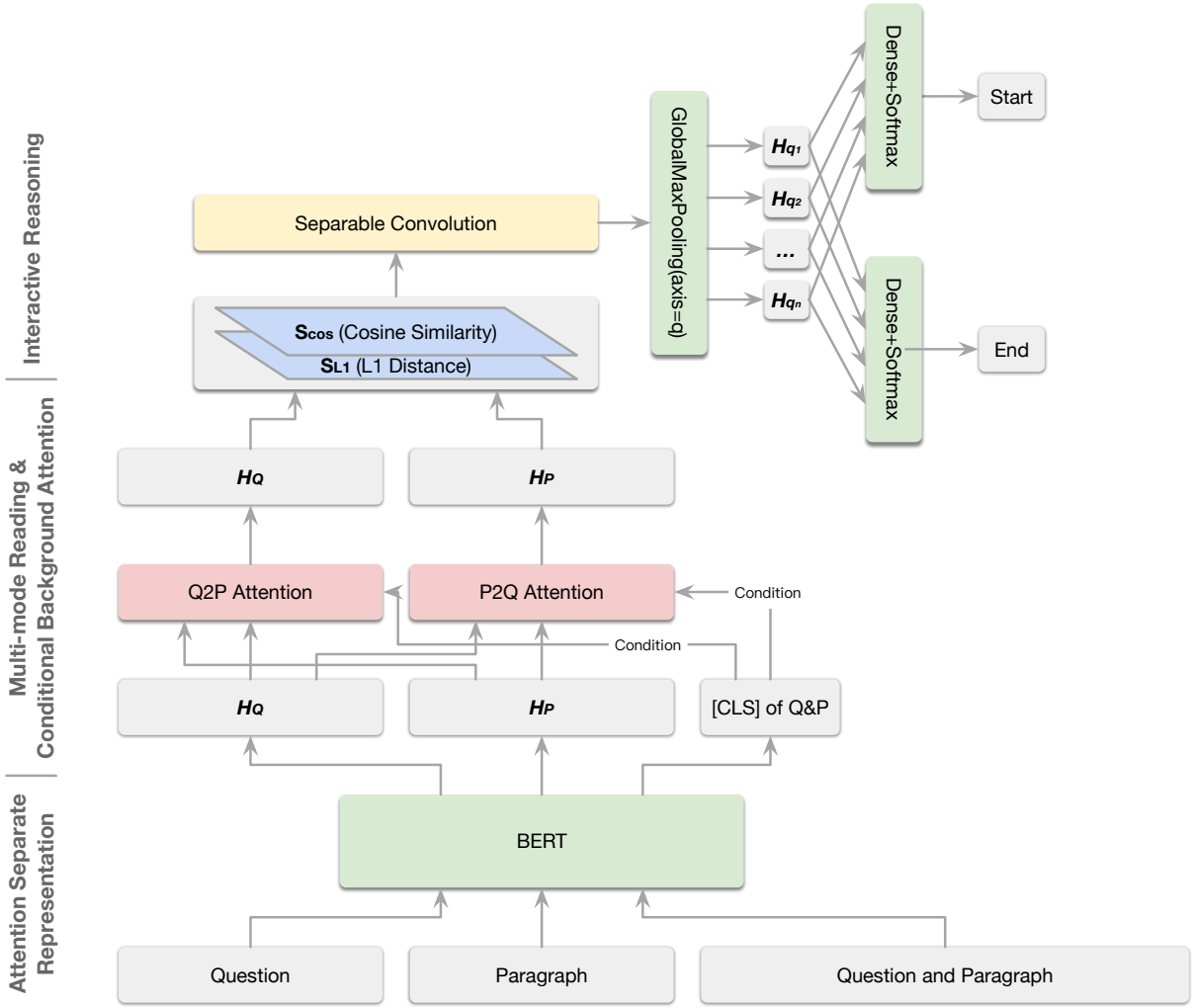

Figure 2: ForceReader overall model structure

word w_{Q_i} in the Q, we compute it with the attention in the pre-generated paragraph sequence $\boldsymbol{H_P}$. And then, fuse the representation $\boldsymbol{H_P}$ in the paragraphs through attention. At last, we add $\tilde{\boldsymbol{H_{Q_i}}}$ back to the $\boldsymbol{h_{Q_i}}$ to finally get the represent w_{Q_i}, as shown below.

$$\begin{aligned} \alpha_{i,j} &= \frac{exp(\boldsymbol{H_{P_j}} \boldsymbol{H_{Q_i}}^T)}{\sum_j^{l_p} exp(\boldsymbol{H_{P_j}} \boldsymbol{H_{Q_i}}^T)} \\ \tilde{\boldsymbol{H_{Q_i}}} &= \sum_j^{l_p} \alpha_{i,j} \boldsymbol{H_P} \\ \boldsymbol{H_{Q_i}} &= \boldsymbol{H_{Q_i}} + \tilde{\boldsymbol{H_{Q_i}}} \end{aligned} \tag{2}$$

Similarly, in the $P2Q$ mode, for each word w_{P_i} in the P, we compute it with the attention in the per-recipient paragraph sequence $\boldsymbol{H_Q}$, fuse the representation $\boldsymbol{H_Q}$ in the problem through attention and add it with $\boldsymbol{h_{P_i}}$ to represent w_{P_i}. The equations are the similar to $Q2P$'s.

In the QCP mode, P and Q should pay attention to each other. Other than designing a new bi-directional attention flow, we consider the original BERT with a combined input of P and Q. It is also the most commonly used method for machine reading comprehension. However, unlike $P2Q$ mode and $Q2P$ mode, after encoding the P, Q pair with BERT, we just get the *[CLS]* as the overall semantic representation $\boldsymbol{h_{QCP}}$.

3.3 Conditional Background Attention

The feature fusion methods commonly used in neural networks are addition, concatenation, and projection. All of these modalities are post-fusion mechanisms. However, in order to perform more interactive reasoning in reading comprehension tasks, we want to do feature fusion as early as possible. Therefore, these modalities can benefit from the results of the other modalities. In particular, for the QCP model, as we mentioned above, we use only *[CLS]* as the holistic semantic encoding of $\boldsymbol{h_{QCP}}$, but not the encoding of the whole input sequence. There are two reasons for this. First, we think that holistic semantics has essential benefits for the interaction of questions and paragraphs. Second, we need to avoid the effects of the attention deconcentration problem we addressed before.

We take $\boldsymbol{h_{QCP}}$, the result of the QCP mode, as the conditional background semantics. Integrate it in the calculation of the attention distribution α of $Q2P$ and $P2Q$. Since the attention score of these modes is calculated based on the conditional background semantics of QCP, we call it Conditional Background Attention. Take $Q2P$ as an example. The calculation process of α_{ij} is shown as follows.

$$\begin{aligned} \alpha_{ij} &= Attention((\boldsymbol{H_P}, \boldsymbol{H_{Q_i}})|\boldsymbol{h_{QCP}}) \\ &= \frac{exp((\boldsymbol{H_{P_j}} + \boldsymbol{h_{QCP}})\boldsymbol{H_{Q_i}}^T)}{\sum_j^{l_p} exp((\boldsymbol{H_{P_j}} + \boldsymbol{h_{QCP}})\boldsymbol{H_{Q_i}}^T)} \end{aligned} \tag{3}$$

The same is true for the $P2Q$ model of attention distribution. The design of $Q2P$ and $P2Q$ not only covers the common-sense models of reading comprehension but also effectively addresses the second problem of attention deconcentration by reducing the influence of single lyrics affected by irrelevant verbs. The present of conditional attention further brings the results of the QCP mode to the computation of two other modes of attention, so that these modes no longer lack knowledge of overall semantics.

3.4 Interactive Reasoning

After modelling Multi-mode Reading and Conditional Background Attention, we can obtain semantic tensor $\boldsymbol{H_P}$ and $\boldsymbol{H_Q}$corresponding to P and Q . People generally use $\boldsymbol{H_P}$ to classify answers directly word by word, as in AttentiveReader, by predicting the start and end index of the answer sequence. However, we believe that the interaction between the question and the paragraph is a better way to represent the underlying patterns in the reading comprehension task. There are always various patterns between the P and Q, such as synonyms or word overlap. Take example1 as an example, *founder* in P and *founded* in Q have similar semantics, and both P and Q share the same *Microsoft* words.

For interactive reasoning, we believe that the literal similarity between questions and paragraphs is crucial underlying information. Therefore, we use Cosine Similarity and $L1$ distance to characterize this similarity and stack the results of both measures together to obtain the interaction tensor $\boldsymbol{S}$.

$$\begin{aligned} \boldsymbol{S_{cos}} &= \frac{\boldsymbol{H_P}\boldsymbol{H_Q}^T}{\|\boldsymbol{H_P}\|\|\boldsymbol{H_Q}\|} \\ \boldsymbol{S_{L_1}} &= ||\boldsymbol{H_P} - \boldsymbol{H_Q}|| \\ \boldsymbol{S} &= Stack(\boldsymbol{S_{cos}}, \boldsymbol{S_{L_1}}) \end{aligned} \tag{4}$$

where $\boldsymbol{S} \in \mathbb{R}^{2 \times l_p \times l_q}$.

By now, the interaction tensor $\boldsymbol{S}$ contains only literal information such as predicates, word overlaps, distance metrics. The combination of these literal features is also essential for the inference of the answer.

For example $< Q, P >$, we need not only to directly word interactions but also to be able to interact with phrase fragments such as *founder of Microsoft* and *founded Microsof*, *who is the founder* and *Bill Gates founded*. In many cases, this can intuitively drive the model to the answer. Therefore, we should further extract the combinatorial features from the interaction tensor $\boldsymbol{S}$ as follows.

$$
\begin{aligned}
&for \quad i \quad in \quad range(n): \\
&\quad \boldsymbol{S} = PointWiseConv(\boldsymbol{S}) \\
&\quad \boldsymbol{S} = DeepWiseConv(\boldsymbol{S})
\end{aligned} \tag{5}
$$

We also need our model to combine multiple layers of abstraction based on word interactions to capture the information interaction between different window segments. The great success of convolutional neural networks in the field of image processing is precisely due to the ability to perform local feature detection and then employ multi-channel modeling to extract more abstract and rich combined features. Similarly, convolutional neural networks can also produce good results on text classification(Kim, 2014). Inspired by this, we use a multi-channel separable convolutional neural network (Separable CNN) (Chollet, 2017; Kaiser et al., 2017) for the extraction of multiple interactive features. First, we perform point-by-point convolutional operations on each channel with different receptor eye sizes. Then we perform 1×1 channel convolutional operations in order to fuse information between channels. After multi-layer separable convolutional operations, we perform global maximum pooling and transposition on the l_q dimension to obtain the output.

$$
\boldsymbol{H} = GlobalMaxPool(\boldsymbol{S}, axis = l_q)^T \tag{6}
$$

where $\boldsymbol{H} \in \mathbb{R}^{l_p \times c}$, c is the number of output channels in the last convolutional layer.

3.5 Answer Prediction

Like most models, we can make stepwise classifications on $\boldsymbol{H} \in \mathbb{R}^{l_p \times c}$, to predict the beginning and end of the answer in the paragraph. During the training, we use the cross-entropy loss function as follows.

$$
L(\theta) = -\sum(logP^s y^s + logP^e y^e) \tag{7}
$$

where θ is the set of parameters in the entire model that can be learned by training, P^s, P^e represents the probability of beginning and ending predictions, and y^s, y^e represents the index of where the ground-truth answer begins and ends.

4 Experiment

4.1 Dataset

We run our experiments on both SQuAD1.1 and SQuAD2.0. Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset consisting of a set of questions in Wikipedia articles. The answer to every question is a segment of text corresponding to the reading paragraph. SQuAD 1.1(Rajpurkar et al., 2016) is an early version of the SQuAD dataset and contains 100,000+ question-answer pairs on more than 500 articles. SQuAD2.0(Rajpurkar et al., 2018) combines 100,000 questions in SQuAD1.1 with more than 50,000 unanswerable questions. In order to achieve excellent results on SQuAD2.0, the system must not only answer questions when possible but also must determine when the paragraph does not support any answers.

4.2 Experiment models

We employ BERT-Large as our baseline model. Then, we perform a series of improvements according to section 3.1, 3.2, 3.3 and 3.4 over the BERT-Large base model. First, we use BERT to encode Q and P separately, thereby achieving Attention Separate Representation. Then perform an AttentiveReader like approach to get the final answer. We name this model as **SepBERTReader**. Next, we add multi-mode modeling to the SepBERTReader to verify the Multi-mode Reading strategy discussed in section 3.2. The second model is called **MultiModeReader**. Furthermore, we include the Conditional Background Attention to the MultiModeReader, then get a new model called **CondAttentionReader**. Finally, we add the interaction module to CondAttentionReader, which constitutes our model that covers all the improvement schemes mentioned in this paper. The final model is called **ForceReader**.

We use a progressive top-to-bottom experiment that corresponds to the proposed improvements. In the experiment results, we can intuitively compare the contribution of each improvement strategy.

4.3 Model setups

For the BERT encoder of all these models, we load the parameters from Google's pre-trained BERT-Large. For the non-BERT layer, we randomly initialized its parameters. The BERT layer only needs to perform fine-tuning learning according to specific downstream tasks. If the learning rate is too large in fine-tuning learning, it will affect the pre-trained model's stability. Therefore, we set the learning rate to 3e-5 in the BERT layer of the model. For the non-BERT layer, we set the learning rate to 1e-4, which can accelerate the parameter update in the gradient descent algorithm. We used AdamW (Loshchilov and Hutter, 2017) as the optimizer, and the learning rate adjustment strategy adopted Warmup(Vaswani et al., 2017; You et al., 2017; Gotmare et al., 2018) with a batch size of 12 for a total of 16000 iterations. To alleviate the problem of overfitting, a combination of $L2$ regularization(Cortes et al., 2012) and dropout(Hinton et al., 2012) was used. By using $L2$ regularization, the loss function of the model is as follows.

$$L_{reg}(\theta) = L(\theta) + \lambda \sum ||\theta|| \tag{8}$$

where θ represents parameters of the model, $\sum ||\theta||$ represents the sum of L_2 paradigms of all model parameters, λ is the penalty term coefficient, which we set to 1e-4.

4.4 Results

According to the settings and parameters described above, we conducted experiments on SQUAD1.1 and SQUAD2.0, respectively. The experiment results are shown in Table1.

Table 1: Comparison of experiment results on the SQuAD

Model	**SQuAD1.1**		**SQuAD2.0**	
	EM	F1	EM	F1
Human	80.3	90.5	**86.3**	**89.0**
Stanford Attentive Reader	69.5	78.8	-	-
BiDAF	67.7	77.3	-	-
QANET	73.6	82.7	-	-
BERT-Large(Google's)	84.1	90.9	-	-
BERT-Large(Ours)	83.3	90.5	78.7	82.0
SepBERTReader	84.1	90.8	79.9	82.9
MultiMode Reader	86.4	91.6	81.1	85.2
Cond Attention Reader	87.3	92.2	82.3	86.6
ForceReader	**88.1**	**93.4**	**84.7**	**88.1**

For BERT-Large, we report the official Google results as well as our results. We attempt to reproduce the BERT-Large results over the Dev set according to the configuration provided by Google. However, the results of the two are still slightly different. Although we believe that such a small difference will not affect any conclusion, but in order to compare under the same conditions, the following comparative analysis will be based on the results of our implementation on the Dev set. Besides, we also listed the performance of Stanford Attentive Reader, BiDAF, QANET, and humans on these data as a performance comparison.

5 Analysis

5.1 Analysis of Results

In Table1, we report the results on the SQuAD datasets. For SepBERTReader, we found that compared with the baseline model, the improvement is relatively small. EM / F1 of SQuAD1.1 and SQuAD2.0

increased by about 1%. However, each indicator has indeed improved. Therefore, we believe that the strategy of Attention Separate Representation has played a certain role in improvement. Since the upper layer of SepBERTReader only adopts a simple inference mode, the effect of Attention Separate Representation is not significant. Observing MultiModeReader, we found that EM / F1 increased by 3.1% and 1.1% on SQUAD1.1, and EM / F1 increased by 2.4% and 3.2% on SQuAD2.0. The improvement is noticeable, which proves that based on the Attention Separate Representation, combining with the multi-mode reading scheme can significantly improve the model's understanding and reasoning ability. This also explains that SepBERTReader's insufficient interactive ability is the main reason for its insignificant performance improvement. Regarding CondAttentionReader, we found that EM / F1 increased by 4% and 1.7% on SQUAD1.1, and 3.6% and 4.6% on SQuAD2.0. The result shows that the model's attention calculation is more accurate in the context of the question-paragraph combination, which can help the model to extract answer fragments better. Finally, observing ForceReader, we found that EM / F1 increased by 4.8% and 2.9% on SQUAD1.1, and EM / F1 increased by 6% and 6.1% on SQuAD2.0, respectively. ForceReader contains all the strategies we have proposed so far. Each indicator has significantly improved on the two datasets. Compared with CondAttentionReader, the improvement of this model is also very significant. We can conclude that upper-layer interactive reasoning and convolutional networks are of great help to the model. Due to the interactive reasoning, the model can learn better patterns and correlations between question and paragraph. Besides, the results show that our model has surpassed human performance on SQuAD1.1. It is also close to human performance on SQuAD2.0.

5.2 Visualization of Attention Enhancement Strategies

To increase the interpretability of the model and explore the learning capabilities of the strategies we proposed in this paper, we visualized the attention of the ForceReader model. Through visualization, we try to display the knowledge learned by the model in an intuitive way and analyze the reason behind the effectiveness of these solutions.

In section 3.1, we adopted the Attention Separate Representation to encode the semantics of the question and the paragraph separately to solve the problem of attention deconcentration. Therefore, we used the separate encoding of the model to encode the examples in 1. We select the core words to visualize their attention weight in the entire question sequence.

Our encoder Transformer has 24 layers, and each layer has 16 attention heads. In the process of multi-layer coding, some studies believe that the lower layers may have more syntactic and grammatical features. In contrast, the upper layers may have more overall abstract semantic features(Peters et al., 2018). Therefore, we choose the attention value of the last layer for visual analysis. For the convenience of visualization, we average the attention weights of the 16 attention heads to represent the final attention distribution, as shown in Fig.3.

Figure 3: Attention Visualization of Core Words in Question

We chose *[CLS]*, *question*, *founder* and *microsoft* as the core words of the question. Comparing with Fig.1, the attentions in the Fig.3 are very consistent with language cognition. For example, the word *question* pays more attention to *a*, *who* and *founder*, while it pays less attention to words such as *i* and *want*. Similarly, the word *founder* pays more attention to *who*, *of*, *microsoft* and itself, and less attention to other unrelated words. Such attention distribution can intuitively allow the model to learn the combination semantics of *founder of microsoft*, which is of great help in understanding the problem. For the overall semantics of the question, *[CLS]*'s attention distribution is also very reasonable, so we can have the

conclusion that the Attention Separate Representation can help the model better understand the meaning of the question. The same is true for the paragraph. Due to space limitations, we will not present additional examples in detail.

Reading comprehension is an interactive process. Hence, after Attention Separate Representation, we propose Multi-mode Reading, Conditional Background Attention, and Interactive Reasoning, to allow the question and the paragraph to interact with each other. To visualize the knowledge learned in these interactions, we select the core words in the question and show the attention of these words relative to the paragraph words in Fig.4.

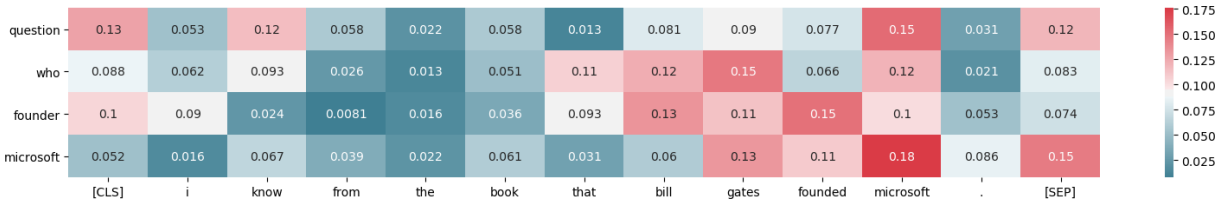

Figure 4: Question Words Attention to Paragraph Sequence

We choose *question*, *who*, *founder* and *microsoft* as the core words of the question. We can see from Fig.4 that these words all have more attention to *[CLS]* and *[SEP]* of the paragraph. We think this is because these two symbols can represent the overall semantics of the paragraph sequence. What is more interesting is that the *question* has better attention to *know*. This result shows that the model has somehow learned context syntactic knowledge and, thus, the ability of reference resolution. Intuitively, word *who* pays the most attention to *that*, *bill* and *gates*, which seems to be able to directly answer the question of *who*. In addition, *founder* and *microsoft* also has greater attention on *bill*, *gates*, *founded* and *microsoft*. In summary, the attention distribution of these core words can significantly help the question get answers in the paragraph.

Figure 5: Paragraph Words Attention to Question Sequence

These attentions is shown in Fig.5. We can see that all these words also have more attention to *[CLS]* and *[SEP]* of the paragraph sequence. This result is consistent with the previous explanation. In addition to pay more attention to *founder*, *microsoft* and *question*, these core words of paragraph also have a higher attention of the word *who*, which largely demonstrates the model's understanding of the question. Regarding the result of non-core word *i*, since it is not very helpful for the overall semantic of the question-answering, its attention is relatively scattered, which is consistent with our expectation.

6 Conclusion

In this paper, we analyzed the use of BERT in machine reading comprehension. We proposed the attention deconcentration problem and conducted a detailed analysis of the impact on reading comprehension. In response to this problem, we have proposed an efficient and straightforward model that integrates Attention Separate Representation, Multi-mode Reading, Conditional Background Attention, and Interactive Reasoning. Attention separate representation effectively solves the problem of attention deconcentration. Multi-mode Reading, Conditional Background Attention, and Interactive Reasoning can make the model better adapt to the high interaction of reading comprehension. The experimental results proved the contribution of each improvement. Besides, we visually analyzed the attention and intuitively demonstrated the learning ability and interpretability of our model. This work is a new approach for machine reading comprehension and provides a new argument for the interpretability of the Transformer model.

Acknowledgements

We are grateful for the help of Dr. Mingzhu Deng that greatly improved the manuscript. We would also like to thank the anonymous reviewers for their insightful comments and suggestions, which are helpful in improving the quality of the paper.

References

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2367.

Danqi Chen. 2018. *Neural reading comprehension and beyond*. Ph.D. thesis, Stanford University.

François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. 2012. L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*.

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

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *International Conference on Learning Representations*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

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

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. 2017. Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

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

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20.

Yang You, Igor Gitman, and Boris Ginsburg. 2017. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.