Relation Module for Non-answerable Prediction on Reading Comprehension

Kevin Huang, Yun Tang, Jing Huang, Xiaodong He, and Bowen Zhou
JD AI Research, Mountain View, CA
{kevin.huang3, yun.tang, jing.huang,
xiadong.he, bowen.zhou}@jd.com

Abstract

Machine reading comprehension (MRC) has attracted significant amounts of research attention recently, due to an increase of challenging reading comprehension datasets. In this paper, we aim to improve a MRC model's ability to determine whether a question has an answer in a given context (e.g. the recently proposed SQuAD 2.0 task). Our solution is a relation module that is adaptable to any MRC model. The relation module consists of both semantic extraction and relational information. We first extract high level semantics as objects from both question and context with multi-head self-attentive pooling. These semantic objects are then passed to a relation network, which generates relationship scores for each object pair in a sentence. These scores are used to determine whether a question is non-answerable. We test the relation module on the SQuAD 2.0 dataset using both the BiDAF and BERT models as baseline readers. We obtain 1.8% gain of F1 accuracy on top of the BiDAF reader, and 1.0% on top of the BERT base model. These results show the effectiveness of our relation module on MRC.

1 Introduction

Ever since the release of many challenging large scale datasets for machine reading comprehension (MRC) (Rajpurkar et al., 2016; Joshi et al., 2017; Trischler et al., 2016; Yang et al., 2018; Reddy et al., 2018; Jia and Liang, 2017), there have been correspondingly many models for these datasets (Yu et al., 2018; Seo et al., 2017; Liu et al., 2018b; Hu et al., 2017; Xiong et al., 2017; Wang et al., 2018; Liu et al., 2018c; Tay et al., 2018). Knowing what you don't know (Rajpurkar et al., 2018) is important in real applications of reading comprehension. Unanswerable questions are commonplace in the real world, and SQuAD 2.0 was released specifically to target this problem (see Figure 1 for an example of non-answerable questions).

Example 1
Context: Each year, the southern California area has about 10,000 earthquakes. Nearly all of them are so small that they are not felt. Only several hundred are greater than magnitude 3.0, and only about 1520 are greater than magnitude 4.0. The magnitude 6.7 1994 **Northridge earthquake** was particularly destructive, causing a substantial number of deaths, injuries, and structural collapses. It caused the most property damage of any earthquake in U.S. history, estimated at over $20 billion.
Question: What earthquake caused $20 million in damage?
Answer: None.

Figure 1: An example of non-answerable question in SQuAD 2.0. Highlighted words are the output from the BERT base model. The true answer is "None".

One problem that most of the early MRC readers have in common is the inability to predict non-answerable questions. Readers on the popular SQuAD dataset have to be modified in order to accommodate a non-answerable possibility. Current methods on SQuAD 2.0 generally attempt to learn a single fully connected layer (Clark and Gardner, 2018; Liu et al., 2018a; Devlin et al., 2018) in order to determine whether a question/context pair is answerable. This leaves out relational information that may be useful for determining answerability. We believe that relationships between different high-level semantics in the context are helpful to make better answerable or unanswerable decision. For example, "Northridge earthquake" is mistakenly taken as the answer to the question about what earthquake caused $20 million in damage. Because "$20 billon" is positioned far away from "Northridge earthquake", it is hard for a model to link these two concepts together and recognize the mismatch of "$20 million" in the question and "$20 billion" in the context.

Motivated by exploiting high level semantic relationships in the context, our first step is to extract meaningful high-level semantics from question/context. Multi-head self-attentive pooling

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 747–756
Hong Kong, China, November 3-4, 2019. ©2019 Association for Computational Linguistics

(Lin et al., 2017) has shown to be able to extract different views of a sentence with multiple heads. Each head from the multi-head self attentive pooling has different weights on the context with learned parameters. This allows each head to act as a filter in order to emphasize part of the context. By summing up the weighted context, we obtain a vector representing an instance of a high-level semantic, which we can call it an "object". With multiple heads, we generate different semantic objects, which are then fed in to a relation network.

Relation networks (Santoro et al., 2017) are specifically designed to model relationships between pairs of objects. In the case of reading comprehension, an object would ideally be phrase level semantics within a sentence. Relation networks are able to accomplish modeling these relationships by constraining the network to learn a score for each pair of these objects. After learning all of the pairwise scores, the relation network then summarizes all of the relations to a single vector. By taking a weighted sum of all of the relation scores that the sentence has, we generate a non-answerable score that is trained jointly with answer span scores from any MRC model to determine non-answerability.

In addition, we add in plausible answers from unanswerable examples to help train the relation module. These plausible answers help the base model learn a better span prediction and are also used to help guide our object extractor to extract relevant semantics. We train a separate layer for start-end probabilities based on the plausible answers. We then augment the context vector with hidden states from this layer. This allows the multi-head self-attentive pooling to focus on objects related to the proposed answer span, and differentiate from other objects that are not as relevant in the context.

In summary we propose a new relation module dedicated to learning relationships between high-level semantics and deciding whether a question is answerable. Our contributions are four-fold:

1. Introduce the concept of using multi-head self-attentive pooling outputs as high level semantic objects.

2. Exploit relation networks to model the relationships between different objects in a context. We then summarize these relationships to get a final decision.

3. Introduce a separate feed-forward layer trained on plausible answers so that we can augment the context vector passed into the object extractor. This results in the object extractor extracting phrases more relevant to the proposed answer span.

4. Combining all of the above into a flexible relation module that can be added to the end of a question answering model to boost non-answerable prediction.

To our knowledge, this is the first case of utilizing an object extractor to extract high level semantics, and a relation networks to encode relationships between these semantics in reading comprehension. Our results show improvement on top of the baseline BiDAF model and the state-of-the-art reader based on BERT, on the SQuAD 2.0 task.

2 Related Work

Relation Networks (RN) were first proposed by (Santoro et al., 2017) in order to help neural models to reason over the relationships between two objects. Relation networks learn relationships between objects by learning a pairwise score for each object pair. Relation networks have been applied to CLEVR (Johnson et al., 2017) as well as bAbI (Weston et al., 2015). In the CLEVR dataset, the object inputs to the relation network are visual objects in an image, extracted by a CNN, and in bAbI the object inputs are sentence encodings. In both tasks, the relation network is then used to compute a relationship score over these objects. Relation Networks were further applied to general reasoning by training the model on images (You et al., 2018).

MAC (Memory, Attention and Composition) networks (Hudson and Manning, 2018) are different models that have also been shown to learn relations from the CLEVR dataset. MAC networks operate with read and write cells. Each cell would compute a relation score between a knowledge base and question and write it into memory. Multiple read and write cells are strung together sequentially in order to model long chains of multi-hop reasoning. Although MAC networks do not explicitly reason between pairwise objects as relation networks do, MAC networks are an interesting way of generating multi-hop reasoning between objects within a context.

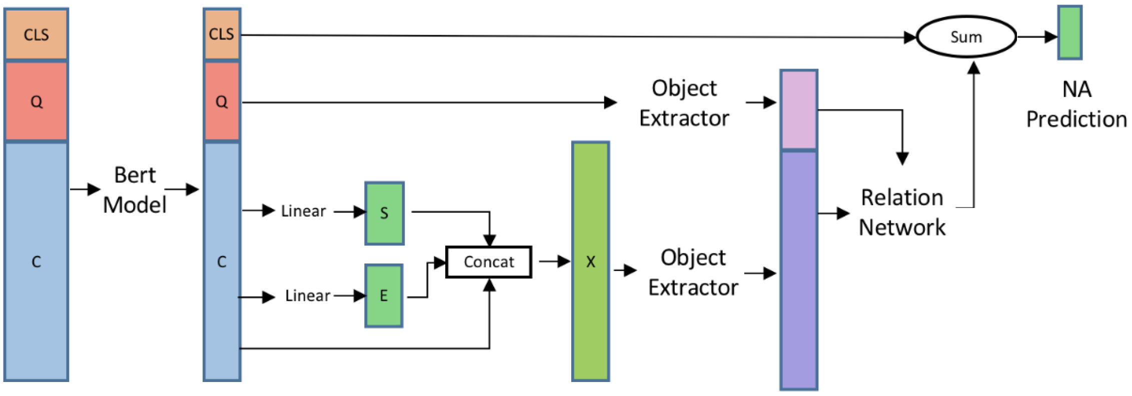

Figure 2: Relation Module on BERT. S and E are hidden states trained by plausible answers. We then concatenate S and E with the contextual representation to feed into the object extractor. After we obtain the extracted objects, we then feed into a Relation Network and pass it down for NA predictions.

Another similar line of work investigated pretraining relationship embeddings across word pairs on large unlabelled corpus (Jameel et al., 2018; Joshi et al., 2018). These pre-trained pairwise relational embeddings were added to the attention layers of BiDAF, where higher level abstract reasoning occurs. The paper showed an impressive gain of 2.7% on the SQuAD 2.0 development set on top of their version of BiDAF.

Many MRC models have been adapted to work on SQuAD 2.0 recently (Hu et al., 2019; Liu et al., 2018a; Sun et al., 2018; Devlin et al., 2018). (Hu et al., 2019) added a separately trained answer verifier for no-answer detection with their Mnemonic Reader. The answer sentence that is proposed by the reader and the question are passed to three combinations of differently configured verifiers for fine-grained local entailment recognition. (Liu et al., 2018a) just added one layer as the unanswerable binary classifier to their SAN reader. (Sun et al., 2018) proposed the U-net with a universal node that encodes the fused information from both the question and passage. The summary U-node, question vector and two context vectors are passed to predict whether the question is answerable. Plausible answers were used for no-answer pointer prediction, while in our approach, plausible answers were used to augment context vector for object extraction that later help the no-answer prediction.

Pretraining embeddings on large unlabelled corpus has been shown to improve many downstream tasks (Peters et al., 2018; Howard and Ruder, 2018; Alec et al., 2018). The recently released BERT (Devlin et al., 2018) greatly increased the F1 scores on the SQuAD 2.0 leaderboard. BERT consists of stacked Transformers (Vaswani et al., 2017), that are pre-trained on vast amounts of unlabeled data with a masked language model. The masked language model helps finetuning on downstream tasks, such as SQuAD 2.0. BERT models contains a special CLS token which is helpful for the SQuAD 2.0 task. This CLS token is trained to predict if a pair of sentences follow each other during the pre-training, which helps encode entailment information between the sentence pair. Due to a strong masked language model to help predict answers and a strong CLS token to encode entailment, BERT models are the current state-of-the art for SQuAD 2.0.

3 Relation Module

Our relation module is flexible, and can be placed on top of any MRC model. We now describe the relation module in detail.

3.1 Augmenting Inputs

Figure 2 shows our relation module on top of the base reader BERT. In addition to the original start-end prediction layers trained from true answers in the base reader, we include a separate start-end prediction layer, with separate parameters, trained specifically on plausible and true answers available in SQuAD 2.0. The context output C from BERT is projected into two hidden state layers S and E, where C, S and $E \in \mathbf{R}^{L \times h}$, L is the context length and h is the hidden size. The S and E layers are then projected down to a hidden dimension of 1, and trained with Cross-Entropy Loss against the plausible and true answer starts and ends. The hidden states S and E of this layer are concatenated with the last context layer output C and projected back to the original dimension to obtain the augmented context vector X, which is fused with start-end span information.

$$S = tanh(CW_1 + b_1) \tag{1}$$
$$E = tanh(CW_2 + b_2) \tag{2}$$
$$X = [C; S; E]W \tag{3}$$

where [;;] is concatenation of multiple tensors and $X \in \mathbf{R}^{L\times h}$. This process is shown in Figure 2, where S and E are hidden states trained on plausible and true answer spans. This tensor X and the last question layer output Q are passed to the object extractor layer.

3.2 Object Extractor

The augmented context tensor X (and separately, question tensor Q) is passed through the object extractor to generate object representations from the tensor. We pass the inputs through a multi-head self-attentive pooling layer. This object extractor can be thought of as a set filters extracting out areas of interest within a sentence. We multiply the input tensor X with a multi-head self attention matrix A which is defined as

$$A = Softmax(W_4\sigma(W_3X^T)) \tag{4}$$
$$O = AX \tag{5}$$

where $W_3 \in R^{h\times h}$, and $W_4 \in R^{n\times h}$; σ is an activation function, such as $tanh$; n is the number of heads, and h is the hidden dimension. The output $O \in R^{n\times h}$ contains the n objects with hidden dimension h that are passed to the next layer.

3.2.1 Object Extraction Regularization

In order to help encourage the multiple heads to extract different meaningful semantics in the text, a regularization loss (Xia et al., 2018) is introduced to encourage each head to attend to slightly different sections of the context. Overlapping objects centered on the answer span are expected, due to information fused from S and E, but we do not want the entire weight distribution of the head to be solely focused on the answer span. As we show in later figures, many heads heavily weight the answer span, but also weight information relevant to the answer span needed to make a better non-answerable prediction. Our regularization term also helps prevent the multi-headed attentive pooling from learning a noisy distribution over all of the context. This regularization loss is defined as

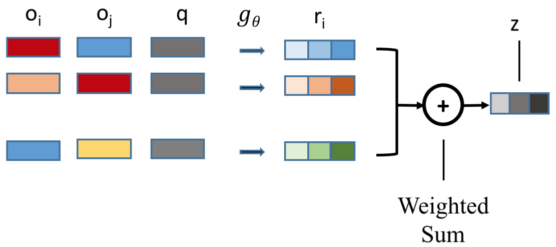

Figure 3: Illustration of a Relation Network. The g_θ is a MLP to score relationships between pairs

$$L_{aux} = \alpha||AA^T - I||_2 \tag{6}$$

where A is the weight matrix for the attention heads and I is the identity matrix. α is set to be 0.0005 in our experiments.

3.3 Relation Networks

Extracted objects are subsequently passed to a relation network. We use two layer MLP g_θ (in Figure 3) as a scoring function to compute the similarity between objects. In the question-answering task, the context contains the contextual information necessary to determine whether a question is answerable. Phrases and ideas from various parts of the context need to come together in order to fully understand whether or not a question is answerable. Therefore our relation module takes all pairs of context objects to score, and use the question objects to guide the scoring function. We use 2 question heads q_0, q_1, so our scoring function is:

$$r_i = \sum_{j=0}^{n} \omega_{i,j} * g_\theta(o_i, o_j, q_0, q_1) \tag{7}$$
$$z = \sum_{i=0}^{n} \gamma_i * f_\phi(r_i) \tag{8}$$

where the outputs r_i is the weighted sum of the relation values for object o_i from O, and z is a summarized relation vector. The weights $\Omega_i = [\omega_{i,0}, ..., \omega_{i,n}]$ and $\Gamma = [\gamma_0, ..., \gamma_n]$ are computed by projecting down the relations scores into a hidden size of 1, and applying softmax.

$$\Omega_i = Softmax(g_\theta(o_i, :, q_0, q_1)w_g) \tag{9}$$
$$\Gamma = Softmax(f_\phi(:)w_f) \tag{10}$$

g_θ and f_ϕ are two layer MLP with activation function $tanh$ to compute and aggregate relational scores. Figure 3 shows the process of a single relation network, where two context objects and question objects are passed in to g_θ to obtain the output z.

We project the weighted sum of f_ϕ with a linear layer to a single value as a representation of the non-answerable score. This score is combined with the start/end logits from the base reader, and trained jointly with the reader's cross-entropy loss. By training jointly, the model is able to make a better prediction based on the confidence of the span prediction, as well as the confidence based on the non-answerable score from the relation module.

4 Question Answering Baselines

We test the relation module on top of our own PyTorch implementation of the BiDAF model (Seo et al., 2017), as well as the recent released BERT base model (Devlin et al., 2018) for the SQuAD 2.0 task. For both of these models, we obtain improvement from adding the relation module. Note that, we do not test our relation module on top of the current leaderboard, as the details are not yet out. We also do not test on top of BERT + Synthetic Self Training (Devlin, 2019) due to lack of computational resources available. We are showing the effectiveness of our method and not trying to compete with the top of the leaderboard.

4.1 BiDAF

We implement the baseline BiDAF model for SQuAD 2.0 task (Clark and Gardner, 2018) with some modifications: adding features that are commonly used in question answering tasks such as TF-IDF, POS/NER tagging, etc, and the auxiliary training losses from (Hu et al., 2019). These modifications to the original BiDAF bring about 3.8% gain of F1 on the SQuAD 2.0 development set (see Table 1).

The input to the relation module is the context vector that is generated from the bi-directional attention flow layer. This context layer is augmented with the hidden states of linear layers trained against plausible answers, which also takes the context layer from the attention flow layer as input. This configuration is shown in Figure 4.

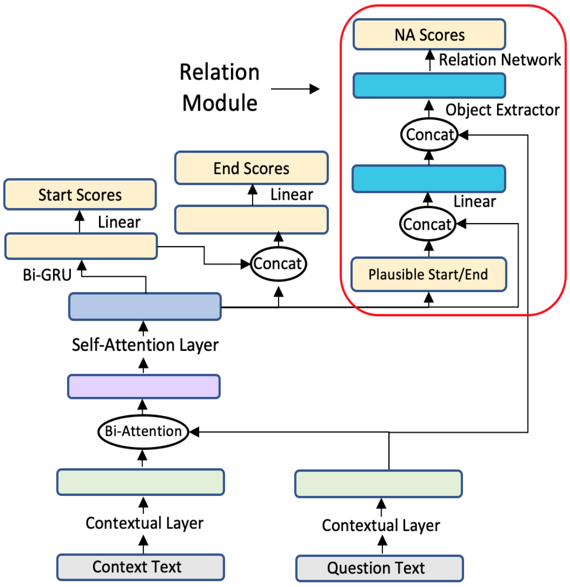

Figure 4: Relation Module applied on BiDAF.

4.2 BERT

BERT is a masked language model pre-trained on large amounts of data that is the core component of all of the current state-of-the-art models on the SQuAD 2.0 task. The input to BERT is the concatenation of a question and context pair in the form of ["CLS"; question; "SEP"; context; "SEP"]. BERT comes with its own special "CLS" token, which is pre-trained on a next sentence pair objective in order to encode entailment information between the two sentences during the pre-training scheme.

We leverage this "CLS" node with the relation module by concatenating it with the output of our Relation Module, and projecting the values down to a single dimension. This combines the information stored in the "CLS" token that has been learned from the pre-training, as well as the information that we learn through our relation module. We allow gradients to be passed through all layers of BERT, and finetune the initialized weights with the SQuAD 2.0 dataset.

5 Experiments

We experiment on the SQuAD 2.0 dataset (Rajpurkar et al., 2018) which contains question and context examples that are crowd-sourced from Wikipedia. Each example contains an answer span in the passage, or an empty string, indicating that an answer doesn't exist. The results are reported on the SQuAD 2.0 development set.

Model	EM(%)	F1(%)
(Clark and Gardner, 2018)	61.9	64.8
Our Implementation of BiDAF	65.7	68.6
BiDAF + Relation Module	**67.7**	**70.4**
BERT-base	73.6	76.6
BERT-base + Relation Module	**74.2**	**77.6**
BERT-large	78.9	82.1
BERT-large + Relation Module	**79.2**	**82.6**

Table 1: Model performance on SQuAD 2.0 development set averaged over three random seeds.

Model	EM(%)	F1(%)
BERT-base	70.7	74.4
BERT-base +Answer Verifier	71.7	75.5
BERT-base + Relation Module	**73.2**	**76.8**

Table 2: SQuAD 2.0 leaderboard numbers on the BERT-base Models. Our model shows improvement over the public BERT-base models on the official evaluation.

We use the following parameters in our BiDAF experiment: 16 context heads, 2 question heads. We set our regularization loss weight for the object extractor to be 0.0005. We use Adam optimizer (Kingma and Ba, 2014), with a start learning rate of 0.0008 and decay the learning rate by 0.5 with a patience of 3 epochs. We add auxiliary losses for plausible answers, and re-rank the non-answerable loss as in (Hu et al., 2019).

BERT comes in two different sizes, a BERT-base model (comprising of roughly 110 million parameters), and a BERT-large model (comprising of roughly 340 million parameters). We use the BERT-base model to run our experiments due to the limited computing resources that training the BERT-large model would take. We only use the BERT-large model to show that we still get improvements with the relation module. The relation module on top of the BERT-base model only contains roughly 10 million parameters.

We use the BERT-base model to run our experiments with the same hyper-parameters given on the official BERT GitHub repository. We use 16 context objects, 2 question heads, and a regularization loss of 0.0005. We also show that on top of the BERT-large model, on the development set, our relation module still obtains performance gain[1]. We use the same number of objects, and the same regularization losses for the BiDAF model experiments.

	Answerable	Non-Answerable
BERT-base	81.5	78.3
+ Relation Module	**82.1**	**82.1**

Table 3: Prediction accuracies on answerable and non-answerable questions on development set.

Table 1 presents the results of the baseline readers with and without the relation module on the SQuAD development set. Our proposed relation module improves the overall F1 and EM accuracy: 2.0% gain on EM and 1.8% gain on F1 on the BiDAF, as well as 0.8% gain on EM and 1.0% gain on F1 on the BERT-base model. Our relation module is able to take relational information between object-pairs and form a better no-answer prediction than a model without it. The module obtains less gain (0.5% gain of F1) on BERT large model due to the better performance of BERT large model. This module is reader independent and works for any reading comprehension model related to non-answerable tasks.

Table 2 presents performance of three BERT-base models with minimum additions taken from the official SQuAD 2.0 leaderboard. We see that our relation module gives more gain than an Answer Verifier on top of the BERT-base model. Our module gains 1.3% F1 over the Answer Verifier.

Since our relation module is designed to help a MRC model's ability to judge non-answerable questions, we examine the accuracy when a question is answerable and when a question is non-answerable. Table 3 compares these accuracy numbers for these questions with and without the relation module on top of the BERT-base model. The relation module improves prediction accuracy for both types of questions, and with more accuracy gain on the non-answerable questions: close to 4% gain on the non-answerable questions, which is more than 200 non-answerable questions are correctly predicted.

6 Ablation Study

We conduct an ablation study to show how different components of the relation module affects the overall performance for the BERT-base model. First we test only adding plausible answers on top of the BERT-base model, in order to quantify the gain in span prediction that adding these extra answers in would give. We show that with just adding plausible answers, the average of the

[1] We do not have enough time to get official SQuAD 2.0 evaluation results for the large BERT models.

Model	EM(%)	F1(%)
BERT-base	73.6	76.6
BERT-base+Plausible Answers	73.5	76.9
BERT-base+RM-Plausible Answers	73.6	76.9
BERT-base+RM (4 heads)	74.1	77.4
BERT-base+RM (16 heads)	**74.2**	**77.6**
BERT-base+RM (64 heads)	74.0	77.2

Table 4: Ablation study on our Relation Module. We experiment with just having plausible answers, just having relation network, and different number of heads for the objects extracted by the relation network. Each of these values are averaged over three random seeds.

three seeds gain only about a 0.3 F1. This gain in F1 is due to the BERT layers being fine-tuned on more answer span data that we provide. Next we study the effects of removing augmenting the context vector with plausible answers. We feed the output of our BERT-base model directly into the object extractor and subsequently to the relation network. This quantifies the effect of forcing the self-attentive heads to focus on a plausible answer span. We notice that this performs comparably to just adding plausible answers, also with only around a 0.3 F1 gain.

Finally, we conduct a study to see the effects of different number of heads on our relation module. We experiment with 4, 16, and 64 heads, with 16 heads performing the best out of these three configurations. Having too few heads hinders the performance due to not enough information being propagated for the relation network to operate on. Having too many heads will introduce redundant information, as well as incorporating extraneous noise for our model to sift through to generate meaningful relations.

7 Analysis

In order to gain better understanding on how the relation module helps on the unanswerable prediction, we examine the objects extracted from the multi-head self-attentive pooling. This is to check whether the relevant semantics are extracted for the relation network. Examples are selected from the development set for data analysis.

In Example 1, the BERT-base model incorrectly outputs “Northridge earthquake” (in red) as the answer. However, after adding our relation module, the model rejects this possible answer and outputs a non-answerable prediction.

Example 1
Context: Each year, the southern California area has about 10,000 earthquakes. Nearly all of them are so small that they are not felt. Only several hundred are greater than magnitude 3.0, and only about 1520 are greater than magnitude 4.0. The magnitude 6.7 1994 **Northridge earthquake** was particularly destructive, causing a substantial number of deaths, injuries, and structural collapses. It caused the most property damage of any earthquake in U.S. history, estimated at over $20 billion.
Question: What earthquake caused $20 million in damage?
Answer: None.

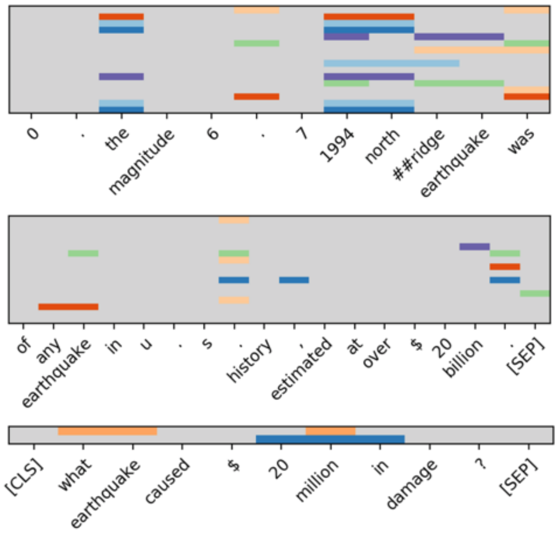

Figure 5: In each subplot, each row represents one object from our object extractor; for each object we highlight the top 5 tokens with highest weights in the entire context and question. We show the two windows where the majority of these top 5 weights occur. For example, the top purple object in the context looks at key phrases such as “##ridge earthquake” in the top subplot and “billion” in the middle subplot; the blue object in the question looks at “20 million in” in the bottom subplot.

The two objects from the question highly attend to token “million” (see the bottom subplot in Figure 5). The top row purple object covers token “1994” , “##ridge earthquake” in the possible answer span window, and “billion” near the end of the context window. We hypothesize that the relation network rejects the possible answer “Northridge earthquake” due to the mismatch of “million” in the question objects and “billion” in the purple context object, and relation scores from all other object pairs.

Example 2 shows another example of non-answerable question and context pair. The BERT-base model incorrectly outputs “input encoding” (in red) as its prediction, while adding our relation module on the BERT-base model predicts correctly that the question is not answerable. Fig-

Example 2
Context: Even though some proofs of complexity-theoretic theorems regularly assume some concrete choice of **input encoding**, one tries to keep the discussion abstract enough to be independent of the choice of encoding. This can be achieved by ensuring that different representations can be transformed into each other efficiently.
Question: What is the abstract choice typically assumed by most complexity-theoretic theorems?
Answer: None.

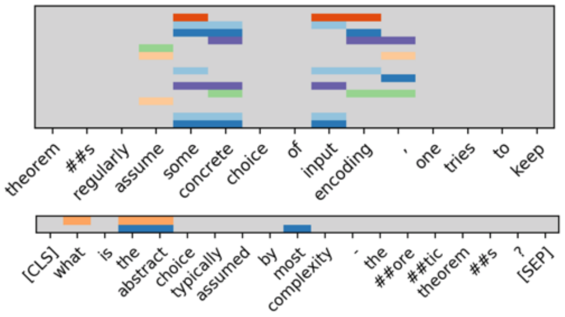

Figure 6: In each subplot, each row represents one object from our object extractor; for each object we highlight the top 5 tokens with highest weights in the entire context and question. We show a window where the majority of the top 5 weights occur. For example, there are numerous objects in the context window that look at the key phrase "some concrete" in the top subplot; the two objects in the question look at the key phrase "the abstract" in the bottom subplot.

ure 6 gives a visual illustration of objects extracted from context and question. In Figure 6, the upper plot illustrates the 16 semantic objects shown in this context window and the lower plot illustrates the two semantic objects from the question. We see that from the upper plot, "some concrete" and "input encoding" are highlighted, while in the lower plot, "what", "the abstract", "most" are highlighted. The mismatch of "the abstract" from the question objects and "some concrete" from the context objects helps indicate that the question is unanswerable.

8 Conclusion

In this work we propose a new relation module that can be applied on any MRC reader and help increase the prediction accuracy on non-answerable questions. We extract high level semantics from multi-head self-attentive pooling. The semantic object pairs are fed into the relation network which makes a guided decision as to whether a question is answerable. In addition we augment the context vector with plausible answers, allowing us to extract objects focused on the proposed answer span, and differentiate from other objects that are not as relevant in the context. Our results on the SQuAD 2.0 dataset using the relation module on both BiDAF and BERT models show improvements from the relation module. These results prove the effectiveness of our relation module.

For future work, we plan to generalize the relation module to other aspects of question answering, including span prediction or multi-hop reasoning.

9 Acknowledgements

We would like to thank Robin Jia and Pranav Rajpurkar for running the SQuAD evaluation on our submitted models.

References

Alec, Karthik Radford, Tim Narsimhan, Salimans, Illya, and Sutskever. 2018. Improving language understanding with generative pre-training.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. *Association for Computational Linguistics.*

Jacob Devlin. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. Lecture Slides.

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

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *Association for Computational Linguistics*, abs/1801.06146.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Mnemonic reader for machine comprehension. *27th International Joint Conference on Artificial Intelligence*, abs/1705.02798.

Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Ming Zhou. 2019. Read + verify: Machine reading comprehension with unanswerable questions. *Association for the Advancement of Artificial Intelligence.*

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. *International Conference on Learning Representations 2018*, abs/1803.03067.

Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. 2018. Unsupervised learning of distributional relation vectors. In *Association of Computational Linguistics.*

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, abs/1707.07328.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *Conference on Computer Vision and Pattern Recognition*.

Mandar Joshi, Eunsol Choi, Omer Levy, Daniel S. Weld, and Luke Zettlemoyer. 2018. pair2vec: Compositional word-pair embeddings for cross-sentence inference. *CoRR*, abs/1810.08854.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Association for Computational Linguistics*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding.

Xiaodong Liu, Wei Li, Yuwei Fang, Aerin Kim, Kevin Duh, and Jianfeng Gao. 2018a. Stochastic answer networks for squad 2.0. *Association for Computational Linguistics*.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018b. Stochastic answer networks for machine reading comprehension. *CoRR*, abs/1712.03556.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018c. Stochastic answer networks for machine reading comprehension. In *Association of Computational Linguistics*, pages 1705–1714. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

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, Association of Computational Linguistics 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789.

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

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

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. 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 4967–4976. Curran Associates, Inc.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *In Proceedings of International Conference on Learning Representations*.

Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. 2018. U-net: Machine reading comprehension with unanswerable questions. *Association for the Advancement of Artificial Intelligence*.

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018. Densely connected attention propagation for reading comprehension. In *Proceedings of Neural Information Processing Systems*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *In Proceedings for Neural Information Processing Systems*, abs/1706.03762.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Association of Computational Linguistics*, pages 1705–1714. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. DCN+: mixed objective and deep residual coattention for question answering. *In Proceedings of International Joint Conferences on Artificial Intelligence*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics.

Haoxuan You, Yifan Feng, Xibin Zhao, Changqing Zou, Rongrong Ji, and Yue Gao. 2018. Pvrnet: Point-view relation neural network for 3d shape recognition. *the 33th AAAI Conference on Artificial Intelligence (AAAI2019)*, abs/1812.00333.

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. *In Proceedings of International Conference on Learning Representations*.