

U-Net: Machine Reading Comprehension with Unanswerable Questions

Fu Sun[†], Linyang Li[†], Xipeng Qiu^{†*}, Yang Liu[‡]

[†] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University

[†] School of Computer Science, Fudan University

[‡] Liulishuo Silicon Valley AI Lab

{fsun17, lyli15, xpqiu}@fudan.edu.cn, yang.liu@liulishuo.com

Abstract

Machine reading comprehension with unanswerable questions is a new challenging task for natural language processing. A key subtask is to reliably predict whether the question is unanswerable. In this paper, we propose a unified model, called U-Net, with three important components: answer pointer, no-answer pointer, and answer verifier. We introduce a universal node and thus process the question and its context passage as a single contiguous sequence of tokens. The universal node encodes the fused information from both the question and passage, and plays an important role to predict whether the question is answerable and also greatly improves the conciseness of the U-Net. Different from the state-of-art pipeline models, U-Net can be learned in an end-to-end fashion. The experimental results on the SQuAD 2.0 dataset show that U-Net can effectively predict the unanswerability of questions and achieves an F1 score of 71.7 on SQuAD 2.0.

Introduction

Machine reading comprehension (MRC) is a challenging task in natural language processing, which requires that machine can read, understand, and answer questions about a text. Benefiting from the rapid development of deep learning techniques and large-scale benchmarks (Hermann et al. 2015; Hill et al. 2015; Rajpurkar et al. 2016), the end-to-end neural methods have achieved promising results on MRC task (Seo et al. 2016; Huang et al. 2017; Chen et al. 2017; Clark and Gardner 2017; Hu et al. 2017). The best systems have even surpassed human performance on the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al. 2016), one of the most widely used MRC benchmarks. However, one of the limitations of the SQuAD task is that each question has a correct answer in the context passage, therefore most models just need to select the most relevant text span as the answer, without necessarily checking whether it is indeed the answer to the question.

To remedy the deficiency of SQuAD, Rajpurkar, Jia, and Liang (2018) developed SQuAD 2.0 that combines SQuAD with new unanswerable questions. Table 1 shows two examples of unanswerable questions. The new dataset requires the MRC systems to know what they don’t know.

To do well on MRC with unanswerable questions, the model needs to comprehend the question, reason among the

Article: Endangered Species Act

Paragraph: "... Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society the species were relatively rare and little opposition was raised.

Question 1: Which laws faced significant opposition?

Plausible Answer: later laws

Question 2: What was the name of the 1937 treaty?

Plausible Answer: Bald Eagle Protection Act

Table 1: Unanswerable Questions from SQuAD 2.0 (Rajpurkar, Jia, and Liang 2018).

passage, judge the unanswerability and then identify the answer span. Since extensive work has been done on how to correctly predict the answer span when the question is answerable (e.g., SQuAD 1.1), the main challenge of this task lies in how to reliably determine whether a question is not answerable from the passage.

There are two kinds of approaches to model the answerability of a question. One approach is to directly extend previous MRC models by introducing a no-answer score to the score vector of the answer span (Levy et al. 2017; Clark and Gardner 2017). But this kind of approaches is relatively simple and cannot effectively model the answerability of a question. Another approach introduces an answer verifier to determine whether the question is unanswerable (Hu et al. 2018; Tan et al. 2018). However, this kind of approaches usually has a pipeline structure. The answer pointer and answer verifier have their respective models, which are trained separately. Intuitively, it is unnecessary since the underlying comprehension and reasoning of language for these components is the same.

In this paper, we decompose the problem of MRC with unanswerable questions into three sub-tasks: answer pointer, no-answer pointer, and answer verifier. Since these three sub-tasks are highly related, we regard the MRC with unanswerable questions as a multi-task learning problem (Caruana 1997) by sharing some meta-knowledge.

We propose the U-Net to incorporate these three sub-tasks into a unified model: 1) an answer pointer to predict a can-

* Corresponding Author.

didate answer span for a question; 2) a no-answer pointer to avoid selecting any text span when a question has no answer; and 3) an answer verifier to determine the probability of the “unanswerability” of a question with candidate answer information. Additionally, we also introduce a universal node and process the question and its context passage as a single contiguous sequence of tokens, which greatly improves the conciseness of U-Net. The universal node acts on both question and passage to learn whether the question is answerable. Different from the previous pipeline models, U-Net can be learned in an end-to-end fashion. Our experimental results on the SQuAD 2.0 dataset show that U-Net effectively predicts the unanswerability of questions and achieves an F1 score of 72.6.

The contributions of this paper can be summarized as follows.

- We decompose the problem of MRC with unanswerable questions into three sub-tasks and combine them into a unified model, which uses the shared encoding and interaction layers. Thus, the three-tasks can be trained simultaneously in an end-to-end fashion.
- We introduce a universal node to encode the common information of the question and passage. Thus, we can use a unified representation to model the question and passage, which makes our model more condensed.
- U-Net is very easy to implement yet effective.

Proposed Model

Formally, we can represent the MRC problem as: given a set of tuples (Q, P, A) , where $Q = (q_1, q_2, \dots, q_m)$ is the question with m words, $P = (p_1, p_2, \dots, p_n)$ is the context passage with n words, and $A = p_{r_s:r_e}$ is the answer with r_s and r_e indicating the start and end points, the task is to estimate the conditional probability $P(A|Q, P)$.

The architecture of our proposed U-Net is illustrated in Figure 1.

U-Net consists of four major blocks: Unified Encoding, Multi-Level Attention, Final Fusion, and Prediction. As shown in Figure 1, we first combine the embedded representation of the question and passage with a universal node u and pass them through a BiLSTM to encode the whole text. We then use the encoded representation to do the information interaction. Then we use the encoded and interacted representation to fuse the full representation and feed them into the final prediction layers to do the multi-task training.

We will describe our model in details in the following.

(A) Unified Encoding

Embedding Following the successful models on SQuAD 1.1, we first embed both the question and the passage with the following features. Glove embedding (Pennington, Socher, and Manning 2014) and Elmo embedding (Peters et al. 2018) are used as basic embeddings. Besides, we use POS embedding, NER embedding, and a feature embedding that includes the exact match, lower-case match, lemma match, and a TF-IDF feature (Chen et al. 2017).

Now we get the question representation $Q = \mathbf{q}_{i=1}^m$ and the passage representation $P = \mathbf{p}_{i=1}^n$, where each word is

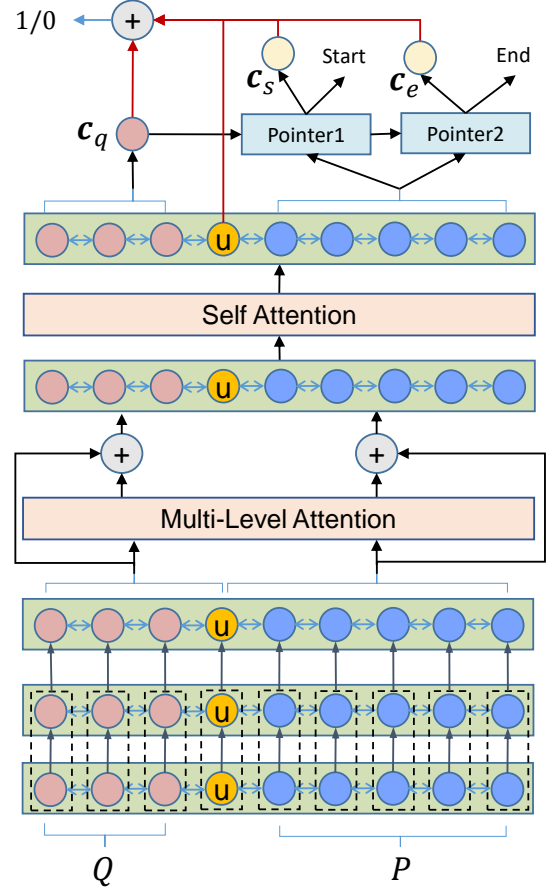


Figure 1: Architecture of the U-Net.

represented as a d -dim embedding by combining the features/embedding described above.

Universal Node We create a universal node u , which is a key factor in our model and has several roles in predicting the unanswerability of question Q .

We expect this node to learn universal information from both passage and question. This universal node is added and connects the passage and question at the phase of embedding, and then goes along with the whole representation, so it is a key factor in information representation. Since the universal node is in between and later shared between passage and question, it has an abstract semantic meaning rather than just a word embedding.

Also, the universal node is later shared in the attention interaction mechanism and used in both the answer boundary detection and classification tasks, so this node carries massive information and has several important roles in our whole model construction.

The universal node u is first represented by a d -dim randomly-initialized vector. We concatenated question representation, universal node representation, passage representation together as:

$$V = [Q, u, P] = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m, \mathbf{u}, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n], \quad (1)$$

$V \in \mathbb{R}^{d \times (m+n+1)}$ is a joint representation of question and passage.

Word-level Fusion Then we first use two-layer bi-directional LSTM (BiLSTM) (Hochreiter and Schmidhuber 1997) to fuse the joint representation of question, universal node, and passage.

$$H^l = \text{BiLSTM}(V), \quad (2)$$

$$H^h = \text{BiLSTM}(H^l), \quad (3)$$

where H^l is the hidden states of the first BiLSTM, representing the low-level semantic information, and H^h is the hidden states of the second BiLSTM, representing the high-level semantic information.

Finally, we concatenate H^l and H^h together and pass them through the third BiLSTM and obtain a full representation H^f .

$$H^f = \text{BiLSTM}([H^l; H^h]). \quad (4)$$

Thus, $H = [H^l; H^h; H^f]$ represents the deep fusion information of the question and passage on word-level. When a BiLSTM is applied to encode representations, it learns the semantic information bi-directionally. Since the universal node u is between the question and passage, its hidden states \mathbf{h}_{m+1} can learn both question and passage information. When the passage-question pair was encoded as a unified representation and information flows via the BiLSTM, the universal node has an important role in information representation.

(B) Multi-Level Attention

To fully fuse the semantic representation of the question and passage, we use the attention mechanism (Bahdanau, Cho, and Bengio 2014) to capture their interactions on different levels.

We expected that we could simply use self-attention on the encoded representation H for interaction between question and passage, which contains both bi-attention (Seo et al. 2016) and self-attention (Wang et al. 2017) of the question and passage. But we found that it performed slightly worse than the traditional bi-directional attention with the universal node included. Therefore, we use a bi-directional attention between the question and passage.

We first divide H into two representations: attached passage H_q and attached question H_p , and let the universal node representation \mathbf{h}_{m+1} attached to both the passage and question, i.e.,

$$H_q = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{m+1}], \quad (5)$$

$$H_p = [\mathbf{h}_{m+1}, \mathbf{h}_{m+2}, \dots, \mathbf{h}_{m+n+1}], \quad (6)$$

Note \mathbf{h}_{m+1} is shared by H_q and H_p . Here the universal node works as a special information carrier, and both passage and question can focus attention information on this node so that the connection between them is closer than a traditional bi-attention interaction.

Since both $H_q = [H_q^l; H_q^h; H_q^f]$ and $H_p = [H_p^l; H_p^h; H_p^f]$ are concatenated by three-level representations, we followed

previous work FusionNet (Huang et al. 2017) to construct their iterations on three levels.

Take the first level as an example. We first compute the affine matrix of H_q^l and H_p^l by

$$S = \left(\text{ReLU}(W_1 H_q^l) \right)^T \text{ReLU}(W_2 H_p^l), \quad (7)$$

where $S \in \mathbb{R}^{(m+1) \times (n+1)}$; W_1 and W_2 are learnable parameters. Next, a bi-directional attention is used to compute the interacted representation \widehat{H}_q^l and \widehat{H}_p^l .

$$\widehat{H}_q^l = H_p^l \times \text{softmax}(S^T), \quad (8)$$

$$\widehat{H}_p^l = H_q^l \times \text{softmax}(S), \quad (9)$$

where $\text{softmax}(\cdot)$ is column-wise normalized function.

We use the same attention layer to model the interactions for all the three levels, and get the final fused representation $\widehat{H}^l, \widehat{H}^h, \widehat{H}^f$ for the question and passage respectively.

Note that while dealing with the attention output of the universal node, we added two outputs from passage-to-question attention and question-to-passage attention. So after the interaction, the fused representation $\widehat{H}^l, \widehat{H}^h, \widehat{H}^f$ still have the same length as the encoded representation H^l, H^h and H^f .

(C) Final Fusion

After the three-level attentive interaction, we generate the final fused information for the question and passage. We concatenate all the history information: we first concatenate the encoded representation H and the representation after attention \widehat{H} (again, we use H^l, H^h, H^f , and $\widehat{H}^l, \widehat{H}^h, \widehat{H}^f$ to represent 3 different levels of representation for the two previous steps respectively).

Following the success of DenseNet (Huang, Liu, and Weinberger 2016), we concatenate the input and output of each layer as the input of the next layer.

First, we pass the concatenated representation H through a BiLSTM to get H^A .

$$H^A = \text{BiLSTM}([H^l; H^h; H^f; \widehat{H}^l; \widehat{H}^h; \widehat{H}^f]), \quad (10)$$

where the representation H^A is a fusion of information from different levels.

Then we concatenate the original embedded representation V and H^A for better representation of the fused information of passage, universal node, and question.

$$A = [V; H^A]. \quad (11)$$

Finally, we use a self-attention layer to get the attention information within the fused information. The self-attention layer is constructed the same way as (Vaswani et al. 2017):

$$\widehat{A} = A \times \text{softmax}(A^T A), \quad (12)$$

where \widehat{A} is the representation after self-attention of the fused information A . Next we concatenated representation H^A and \widehat{A} and pass them through another BiLSTM layer:

$$O = \text{BiLSTM}[H^A; \widehat{A}]. \quad (13)$$

Now O is the final fused representation of all the information. At this point, we divide O into two parts: O^P , O^Q , representing the fused information of the question and passage respectively.

$$O^P = [\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_m], \quad (14)$$

$$O^Q = [\mathbf{o}_{m+1}, \mathbf{o}_{m+2}, \dots, \mathbf{o}_{m+n+1}], \quad (15)$$

Note for the final representation, we attach the universal node only in the passage representation O^P . This is because we need the universal node as a focus for the pointer when the question is unanswerable. These will be fed into the next decoder prediction layer.

(D) Prediction

The prediction layer receives fused information of passage O^P and question O^Q , and tackles three prediction tasks: (1) answer pointer, (2) no-answer pointer and (3) answer verifier.

First, we use a function shown below to summarize the question information O^Q into a fixed-dim representation \mathbf{c}_q .

$$\mathbf{c}_q = \sum_i \frac{\exp(W_q^\top o_i^Q)}{\sum_j \exp(W_q^\top o_j^Q)} o_i^Q, \quad (16)$$

where W_q is a learnable weight matrix and o_i^Q represents the i_{th} word in the question representation. Then we feed \mathbf{c}_q into the answer pointer to find boundaries of answers (Wang and Jiang 2016), and the classification layer to distinguish whether the question is answerable.

(i) Answer Pointer We use this answer pointer to detect the answer boundaries from the passage when the question is answerable (i.e., the answer is a span in the passage). This layer is a classic pointer net structure (Vinyals, Fortunato, and Jaitly 2015). We use two trainable matrices W_s and W_e to estimate the probability of the answer start and end boundaries of the i_{th} word in the passage, α_i and β_i .

$$\alpha_i \propto \exp(\mathbf{c}_q W_s o_i^P), \quad (17)$$

$$\beta_i \propto \exp(\mathbf{c}_q W_e o_i^P), \quad (18)$$

Note here when the question is answerable, we do not consider the universal node in answer boundary detection, so we have $i > 0$ ($i = 0$ is the universal node in the passage representation). The loss function for the answerable question pairs is:

$$\mathcal{L}_A = -(\log \alpha_a + \log \beta_b), \quad (19)$$

where a and b are the ground-truth of the start and end boundary of the answer.

(ii) No-Answer Pointer Then we use the same pointer for questions that are not answerable. Here the loss \mathcal{L}_{NA} is:

$$\mathcal{L}_{NA} = -(\log \alpha_0 + \log \beta_0), \quad (20)$$

α_0 and β_0 correspond to the position of the universal node, which is at the front of the passage representation O_p . For this scenario, the loss is calculated for the universal node.

Additionally, since there exists a plausible answer for each unanswerable question in SQuAD 2.0, we introduce an auxiliary *plausible answer pointer* to predict the boundaries of the plausible answers. The plausible answer pointer has the same structure as the answer pointer, but with different parameters. Thus, the total loss function is:

$$\mathcal{L}_{NA} = -(\log \alpha_0 + \log \beta_0) - (\log \alpha'_{a^*} + \log \beta'_{b^*}), \quad (21)$$

where α' and β' are the output of the plausible answer pointer; a^* and b^* are the start and end boundary of the unanswerable answer.

The no-answer pointer and plausible answer pointer are removed at test phase.

(iii) Answer Verifier We use the answer verifier to distinguish whether the question is answerable.

Answer verifier applies a weighted summary layer to summarize the passage information into a fixed-dim representation \mathbf{c}_p (as shown in Eq.(16)).

And we use the weight matrix obtained from the answer pointer to get two representations of the passage.

$$\mathbf{c}_s = \sum_i \alpha_i \cdot o_i^P \quad (22)$$

$$\mathbf{c}_e = \sum_i \beta_i \cdot o_i^P \quad (23)$$

Then we use the universal node \mathbf{o}_{m+1} and concatenate it with the summary of question and passage to make a fixed vector

$$F = [\mathbf{c}_q; \mathbf{o}_{m+1}; \mathbf{c}_s; \mathbf{c}_e]. \quad (24)$$

This fixed F includes the representation \mathbf{c}_q representing the question information, and \mathbf{c}_s and \mathbf{c}_e representing the passage information. Since these representations are highly summarized specially for classification, we believe that this passage-question pair contains information to distinguish whether this question is answerable. In addition, we include the universal node as a supplement. Since the universal node is pointed at when the question is unanswerable and this node itself already contains information collected from both the passage and question during encoding and information interaction, we believe that this node is important in distinguishing whether the question is answerable.

Finally, we pass this fixed vector F through a linear layer to obtain the prediction whether the question is answerable.

$$p^c = \sigma(W_f^\top F) \quad (25)$$

where σ is a sigmoid function, W_f is a learnable weight matrix.

Here we use the cross-entropy loss in training.

$$\mathcal{L}_{AV} = -(\delta \cdot \log p^c + (1 - \delta) \cdot (\log(1 - p^c))), \quad (26)$$

where $\delta \in \{0, 1\}$ indicates whether the question has an answer in the passage.

Compared with other relatively complex structures developed for this MRC task, our U-Net model passes the original question and passage pair through embedding and encoding, which then interacts with each other, yielding fused information merged from all the levels. The entire architecture is very easy to construct. After we have the fused representation of the question and passage, we pass them through the pointer layer and a fused information classification layer in a multi-task setup.

Training

We jointly train the three tasks by combining the three loss functions. The final loss function is:

$$\mathcal{L} = \delta \mathcal{L}_A + (1 - \delta) \mathcal{L}_{NA} + \mathcal{L}_{AV}, \quad (27)$$

where $\delta \in \{0, 1\}$ indicates whether the question has an answer in the passage, \mathcal{L}_A , \mathcal{L}_{NA} and \mathcal{L}_{AV} are the three loss functions of the answer pointer, no-answer pointer, and answer verifier.

Although the three tasks could have different weights in the final loss function and be further fine-tuned after joint training, here we just consider them in the same weight and do not fine-tune them individually.

At the test phase, we first use the answer pointer to find a potential answer to the question, while the verifier layer judges whether the question is answerable. If the classifier predicts the question is unanswerable, we consider the answer extracted by the answer pointer as plausible. In this way, we get the system result.

Experiment

Datasets

Recently, machine reading comprehension and question answering have progressed rapidly, owing to the computation ability and publicly available high-quality datasets such as SQuAD. Now new research efforts have been devoted to the newly released answer extraction test with unanswerable questions, SQuAD 2.0 (Rajpurkar, Jia, and Liang 2018). It is constructed by combining question-answer pairs selected from SQuAD 1.0 and newly crafted unanswerable questions. These unanswerable questions are created by workers that were asked to pose questions that cannot be answered based on the paragraph alone but are similar to the answerable questions. It is very difficult to distinguish these questions from the answerable ones. We evaluate our model using this data set. It contains over 100,000+ questions on 500+ wikipedia articles.

Implementation Details

We use Spacy to process each question and passage to obtain tokens, POS tags, NER tags and lemmas tags of each text. We use 12 dimensions to embed POS tags, 8 for NER tags (Chen et al. 2017). We use 3 binary features: exact match, lower-case match and lemma match between the question and passage (Lee et al. 2016). We use 100-dim Glove pre-trained word embeddings and 1024-dim Elmo embeddings. All the LSTM blocks are bi-directional with one single layer.

We set the hidden layer dimension as 125, attention layer dimension as 250. We added a dropout layer over all the modeling layers, including the embedding layer, at a dropout rate of 0.3 (Srivastava et al. 2014). We use Adam optimizer with a learning rate of 0.002 (Kingma and Ba 2014).

During training, we omit passage with over 400 words and question with more than 50 words. For testing, when the passage has over 600 words and the question is over 100 words, we simply label these questions as unanswerable.

Main Results

Our model achieves an F1 score of 74.0 and an EM score of 70.3 on the development set, and an F1 score of 72.6 and an EM score of 69.2 on Test set¹, as shown in Table 2. Our model outperforms most of the previous approaches. Comparing to the best-performing systems, our model has a simple architecture and is an end-to-end model. In fact, among all the end-to-end models, we achieve the best F1 scores. We believe that the performance of the U-Net can be boosted with an additional post-processing step to verify answers using approaches such as (Hu et al. 2018).

Ablation Study

We also do an ablation study on the SQuAD 2.0 development set to further test the effectiveness of different components in our model. In Table 3, we show four different configurations.

First, we remove the universal node U . We let the negative examples focus on the plausible answer spans instead of focusing on the universal node U . This results in a loss of 2.6% F1 score on the development set, showing that the universal node U indeed learns information about whether the question is answerable.

We also tried to make the universal node U only attached to the passage representation when passing the attention layer. Our results showed that when node U is shared, as it is called ‘universal’, it learns information interaction between the question and passage, and when it is not shared, the performance slightly degraded.

As for the approaches to encode the representations, we pass both the question and passage through a shared BiLSTM. To test the effectiveness of this, we ran the experiment using separate BiLSTMs on embedded question and passage representations. Results show that the performance dropped slightly, suggesting sharing BiLSTM is an effective method to improve the quality of the encoder.

After removing the plausible answer pointer, the performance also dropped, indicating the plausible answers are useful to improve the model even though they are incorrect.

After removing the answer verifier, the performance dropped greatly, indicating it is vital for our model.

Lastly, we run a test using a more concise configuration. In the second block (multi-level attention) of the U-Net, we do not split the output of the encoded presentation and let it pass through a self-attention layer. The bidirectional attention is removed. In this way, our model uses only one unified

¹<https://rajpurkar.github.io/SQuAD-explorer/>

Model	Dev		Test	
	EM	F1	EM	F1
End-to-end Model				
BNA* (Rajpurkar, Jia, and Liang 2018)	59.8	62.6	59.2	62.1
DocQA (Rajpurkar, Jia, and Liang 2018)	65.1	67.6	63.4	66.3
FusionNet++ (Huang et al. 2017)	-	-	66.6	69.6
SAN (Liu et al. 2017)	-	-	68.6	71.4
VS ³ -Net	-	-	68.4	71.3
U-Net	70.3	74.0	69.2	72.6
Ensemble Model				
FusionNet++ (ensemble)	-	-	70.3	72.6
SAN (ensemble)	-	-	71.3	73.7
U-Net (ensemble)	-	-	71.5	75.0
Pipeline Model				
RMR+ELMo+Verifier (Hu et al. 2018)	72.3	74.8	71.7	74.2
Human	86.3	89.0	86.9	89.5

Table 2: Evaluation results on the SQuAD 2.0 (extracted on Sep 9, 2018). * means the BiDAF (Seo et al. 2016) with No Answer.

Configuration	EM	F1	Δ EM	Δ F1
U-Net	70.3	74.0	-	-
no node U	67.9	71.4	-2.4	-2.6
no share U	69.7	73.5	-0.6	-0.5
no concatenate P & Q	69.0	72.8	-1.3	-1.2
no plausible answer pointer	69.6	72.9	-0.7	-1.1
no classification	63.5	68.5	-6.8	-5.5
Self-Attn Only	69.7	73.5	-0.5	-0.5

Table 3: Comparison of different configurations for our U-Net model.

representation of the question and passage at all time. We simply pass this representation layer by layer to get the final result. Compared to the bi-attention model, the F1-score decreases 0.5%.

Multi-task Study

We also run an experiment to test the performance of our multi-task model. We select different losses that participate in the training procedure to observe the performance affected by answer boundary detect or classification.

Table 4 shows the performance. Here we use EM^* and $F1^*$ to represent the EM and F1 score when the classification is not part of the task, which makes it very much like the task in SQuAD 1.1.

Loss	EM^*	$F1^*$	Classification Acc.
\mathcal{L}	75.3	84.8	80.2
\mathcal{L}_{AV}	-	-	67.1
\mathcal{L}_A	77.2	85.1	-

Table 4: Multi-task performance on the development set.

To test our classifier performance, we do not use backward propagation over the loss of answer boundary detection and simply run a classification task. Results (the first two rows in Table 4) show that there is a large gain when using the multi-task model. The answer boundary detection task helps the encoder learn information between the passage and question and also feed information into the universal node, therefore we can use a summarized representation of the passage and question as well as the universal node to distinguish whether the question is answerable, i.e., help improve classification.

For the answer boundary detection task, we find that the multi-task setup (i.e., the classification layer participates in the training process) does not help its performance. Since the classifier and pointer layer shared the encoding process, we originally expected that classification information can help detect answer boundaries. But this is not the case. We think this is also reasonable since distinguishing whether the question is answerable is mainly focusing on the interactions between the passage-question pair, so once the question is predicted as answerable or not, it has nothing to do with the answer boundaries. This is consistent with how human-beings do this classification task.

We also run the test over SQuAD 1.1 development test to evaluate the performance. Due to a condensed structure, our model achieves an $F1^*$ score of less than 86%, which is not a very competitive score on SQuAD 1.1 test. But as shown above, our model achieves a good score in SQuAD 2.0 test, which shows this model has the potential to achieve higher performance by making progress on both the answer detection and classification tasks.

Overall, we can conclude that our multi-task model works well since the performance of unanswerability classification improves significantly when the answer pointer and answer verifier work simultaneously.

Study on the Different Thresholds of Unanswerability Classification

The output b of the answer verifier is the probability of a question being unanswerable. The smaller the output, the lower the probability of unanswerability is. In SQuAD 2.0, the proportions of unanswerable questions are different in the training and test sets. The default threshold 0.5 is optimized on the training set, but not suitable for the test set. Therefore, it is reasonable to set a proper threshold to manually adapt to the test set.

As mentioned in SQuAD 2.0 paper (Rajpurkar, Jia, and Liang 2018), different thresholds for answerability prediction result in fluctuated scores between answerable and unanswerable questions. Here we show the variation of the F1 score with different thresholds in Figure . The threshold between $[0, 1]$ is used to decide whether a question can be answered. When the threshold is set to 0, all questions are considered as answerable.

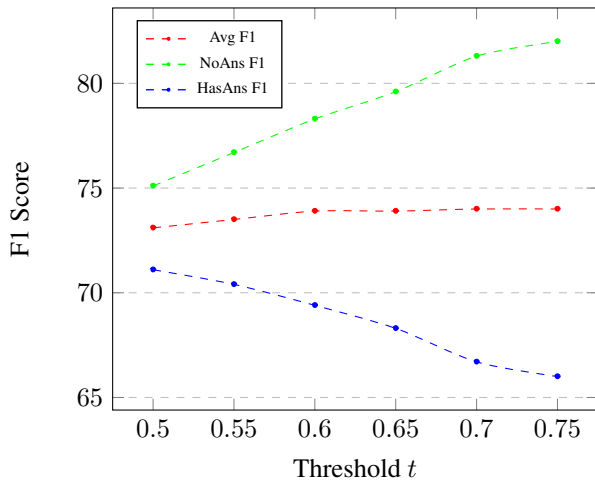


Figure 2: F1 score variation with different thresholds. “NoAns F1” is the recall of unanswerable questions.

As we can see, when the threshold is set to 0.5, F1 score of answerable questions is similar to that of unanswerable questions. When we increase the threshold (i.e., more likely to predict the question as unanswerable), performance for answerable questions degrades, and improves for unanswerable questions. This is as expected. We can see that the overall F1 score is slightly better, which is consistent with the idea from SQuAD 2.0. In addition, we find that for larger thresholds, the variance between EM and $F1$ is narrowed since EM and $F1$ scores for unanswerable questions are the same.

Finally, we set the threshold to be 0.7 for the submission system to SQuAD evaluation.

Related Work

End-to-end Models for MRC

Currently, end-to-end neural network models have achieved great successes for machine reading comprehension (Seo et

al. 2016; Kumar et al. 2015; Sukhbaatar et al. 2015; Cui et al. 2016; Xiong, Zhong, and Socher 2016; Dhingra et al. 2016; Shen et al. 2016; Hu et al. 2017; Wang, Yan, and Wu 2018). Most of these models consist of three components: encoder, interaction, and pointer. The BiLSTM is widely used for encoding the embedded representation. For the interaction, bidirectional attention mechanism is very effective to fuse information of the question and passage. Finally, a pointer network (Vinyals, Fortunato, and Jaitly 2015) is used to predict the span boundaries of the answer. Specifically, in SQuAD test (Rajpurkar et al. 2016), there are approaches to combine match-LSTM and pointer networks to produce boundaries of the answer and employ variant bidirectional attention mechanism to match the question and passage mutually.

In our model, we learn from previous work and develop a condensed end-to-end model for the SQuAD 2.0 task. Different from the previous models, we use a unified representation to encode the question and passage simultaneously, and introduce a universal node to encode the fused information of the question and passage, which also plays an important role to predict the unanswerability of a question.

MRC with Unanswerable Questions

MRC with unanswerable questions is a more challenging task. Previous work Levy et al.; Clark and Gardner (2017; 2017) has attempted to normalize a no-answer score depending on the probability of all answer spans and still detect boundaries at the same time. But the scores of the answer span predictions are not very discriminative in distinguishing whether the question is answerable. Therefore, this kind of approaches, though relatively simple, cannot effectively deal with the answerability of a question.

Hu et al.; Tan et al. (2018; 2018) introduced an answer verifier idea to construct a classification layer. However, this kind of approaches usually has a pipeline structure. The answer pointer and answer verifier have their respective models that are trained separately.

Multi-task models Different from existing work, we regard the MRC with unanswerable questions as a multi-task learning problem (Caruana 1997) by sharing some meta-knowledge. Intuitively, answer prediction and answer verification are related tasks since the underlying comprehension and reasoning of language for these components is the same. Therefore, we construct a multi-task model to solve three sub-tasks: answer pointer, no-answer pointer, and answer verifier.

Conclusion and Future Work

In this paper, we regard the MRC with unanswerable questions as multi-task learning problems and propose the U-Net, a simple end-to-end model for MRC challenges. U-Net has good performance on SQuAD 2.0. We first add a universal node to learn a fused representation from both the question and passage, then use a concatenated representation to pass through encoding layers. We only treat question and passage differently during attention interactions. In the rest blocks of

U-Net, we still use the unified representation containing both the question and passage representation. Finally, we train the U-Net as a multi-task framework to determine the final answer boundaries as well as whether the question is answerable. Our model has very simple structure yet achieves good results on SQuAD 2.0 test.

Our future work is to reconstruct the structure of U-Net by replacing the current multi-level attention block with a simpler self-attention mechanism, which we believe can capture the question and passage information, and intuitively is also coherent with the rest of our U-Net model. In addition, we will improve the answer boundary detection performance based on some of the previous successful models. Since our model actually does not achieve very competitive performance in the boundary detection task yet still has a good overall performance on SQuAD 2.0 test, we are optimistic that our U-Net model is potentially capable of achieving better performance. Furthermore, our model has a simple structure and is easy to implement, therefore we believe that our model can be easily modified for various datasets.

Acknowledgement

We would like to thank Robin Jia, Pranav Rajpurkar for their help with SQuAD 2.0 submissions.

References

- [Bahdanau, Cho, and Bengio 2014] Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *ArXiv e-prints*.
- [Caruana 1997] Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.
- [Chen et al. 2017] Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading wikipedia to answer open-domain questions. *CoRR* abs/1704.00051.
- [Clark and Gardner 2017] Clark, C., and Gardner, M. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.
- [Cui et al. 2016] Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2016. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*.
- [Dhingra et al. 2016] Dhingra, B.; Liu, H.; Cohen, W. W.; and Salakhutdinov, R. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*.
- [Hermann et al. 2015] Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, 1684–1692.
- [Hill et al. 2015] Hill, F.; Bordes, A.; Chopra, S.; and Weston, J. 2015. The goldilocks principle: Reading children’s books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.
- [Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- [Hu et al. 2017] Hu, M.; Peng, Y.; Huang, Z.; Qiu, X.; Wei, F.; and Zhou, M. 2017. Reinforced mnemonic reader for machine reading comprehension. *arXiv preprint arXiv:1705.02798*.
- [Hu et al. 2018] Hu, M.; Peng, Y.; Huang, Z.; Yang, N.; Zhou, M.; et al. 2018. Read+ verify: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1808.05759*.
- [Huang et al. 2017] Huang, H.; Zhu, C.; Shen, Y.; and Chen, W. 2017. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *CoRR* abs/1711.07341.
- [Huang, Liu, and Weinberger 2016] Huang, G.; Liu, Z.; and Weinberger, K. Q. 2016. Densely connected convolutional networks. *CoRR* abs/1608.06993.
- [Kingma and Ba 2014] Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.
- [Kumar et al. 2015] Kumar, A.; Irsoy, O.; Su, J.; Bradbury, J.; English, R.; Pierce, B.; Ondruska, P.; Gulrajani, I.; and Socher, R. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.
- [Lee et al. 2016] Lee, K.; Kwiatkowski, T.; Parikh, A. P.; and Das, D. 2016. Learning recurrent span representations for extractive question answering. *CoRR* abs/1611.01436.
- [Levy et al. 2017] Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.
- [Liu et al. 2017] Liu, X.; Shen, Y.; Duh, K.; and Gao, J. 2017. Stochastic answer networks for machine reading comprehension. *CoRR* abs/1712.03556.
- [Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)* 12:1532–1543.
- [Peters et al. 2018] Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proc. of NAACL*.
- [Rajpurkar et al. 2016] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- [Rajpurkar, Jia, and Liang 2018] Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don’t Know: Unanswerable Questions for SQuAD. *ArXiv e-prints*.
- [Seo et al. 2016] Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.
- [Shen et al. 2016] Shen, Y.; Huang, P.-S.; Gao, J.; and Chen, W. 2016. Reasonet: Learning to stop reading in machine comprehension. *arXiv preprint arXiv:1609.05284*.
- [Srivastava et al. 2014] Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R.

2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.
- [Sukhbaatar et al. 2015] Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, 2431–2439.
- [Tan et al. 2018] Tan, C.; Wei, F.; Zhou, Q.; Yang, N.; Lv, W.; and Zhou, M. 2018. I know there is no answer: Modeling answer validation for machine reading comprehension. In *CCF International Conference on Natural Language Processing and Chinese Computing*, 85–97. Springer.
- [Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.
- [Vinyals, Fortunato, and Jaitly 2015] Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. *ArXiv e-prints*.
- [Wang and Jiang 2016] Wang, S., and Jiang, J. 2016. Machine comprehension using match-lstm and answer pointer. *CoRR* abs/1608.07905.
- [Wang et al. 2017] Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 189–198.
- [Wang, Yan, and Wu 2018] Wang, W.; Yan, M.; and Wu, C. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1705–1714.
- [Xiong, Zhong, and Socher 2016] Xiong, C.; Zhong, V.; and Socher, R. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.