RikiNet: Reading Wikipedia Pages for Natural Question Answering

Dayiheng Liu[♠†] **Yeyun Gong**[†] **Jie Fu**[♢] **Yu Yan**[†]
Jiusheng Chen[♡] **Daxin Jiang**[‡] **Jiancheng Lv**[♠] **Nan Duan**[†]
[♠]College of Computer Science, Sichuan University [†]Microsoft Research Asia
[♢] Mila [‡]Microsoft Search Technology Center Asia [♡]Microsoft AI and Research
losinuris@gmail.com

Abstract

Reading long documents to answer open-domain questions remains challenging in natural language understanding. In this paper, we introduce a new model, called RikiNet, which reads Wikipedia pages for natural question answering. RikiNet contains a dynamic paragraph dual-attention reader and a multi-level cascaded answer predictor. The reader dynamically represents the document and question by utilizing a set of complementary attention mechanisms. The representations are then fed into the predictor to obtain the span of the short answer, the paragraph of the long answer, and the answer type in a cascaded manner. On the Natural Questions (NQ) dataset, a single RikiNet achieves 74.3 F1 and 57.9 F1 on long-answer and short-answer tasks. To our best knowledge, it is the first single model that outperforms the single human performance. Furthermore, an ensemble RikiNet obtains 76.1 F1 and 61.3 F1 on long-answer and short-answer tasks, achieving the best performance on the official NQ leaderboard[1].

1 Introduction

Machine reading comprehension (MRC) refers to the task of finding answers to given questions by reading and understanding some documents. It represents a challenging benchmark task in natural language understanding (NLU). With the progress of large-scale pre-trained language models (Devlin et al., 2018), state-of-the-art MRC models (Ju et al., 2019; Yang et al., 2019; Lan et al., 2019; Zhang et al., 2019; Liu et al., 2019) have already surpassed human-level performance on certain commonly used MRC benchmark datasets, such as SQuAD 1.1 (Rajpurkar et al., 2016), SQuAD 2.0 (Rajpurkar et al., 2018), and CoQA (Reddy et al., 2019).

Recently, a new benchmark MRC dataset called Natural Questions[2] (NQ) (Kwiatkowski et al., 2019) has presented a substantially greater challenge for the existing MRC models. Specifically, there are two main challenges in NQ compared to the previous MRC datasets like SQuAD 2.0. **Firstly**, instead of providing one relatively short paragraph for each question-answer (QA) pair, NQ gives an entire Wikipedia page which is significantly longer compared to other datasets. **Secondly**, NQ task not only requires the model to find an answer span (called short answer) to the question like previous MRC tasks but also asks the model to find a paragraph that contains the information required to answer the question (called long answer).

In this paper, we focus on the NQ task and propose a new MRC model called **RikiNet** tailored to its associated challenges, which **R**eads the **Wiki**pedia pages for natural question answering. For the first challenge of the NQ task mentioned above, RikiNet employs the proposed Dynamic Paragraph Dual-Attention (DPDA) reader which contains multiple DPDA blocks. In each DPDA block, we iteratively perform dual-attention to represent documents and questions, and employ paragraph self-attention with dynamic attention mask to fuse key tokens in each paragraph. The resulting context-aware question representation, question-aware token-level, and paragraph-level representations are fed into the predictor to obtain the answer. The motivations of designing DPDA reader are: (a) Although the entire Wikipedia page contains a large amount of text, one key observation is that most answers are only related to a few words in one paragraph; (b) The final paragraph representation can be used naturally for predicting long answers.

[1]Till our submission time, 29 Nov. 2019. We refer readers to https://ai.google.com/research/NaturalQuestions/leaderboard for the latest results.

[2]NQ provides some visual examples of the data at https://ai.google.com/research/NaturalQuestions/visualization.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6762–6771
July 5 - 10, 2020.

We describe the details of DPDA reader in § 3.1.

For the second challenge, unlike prior works on NQ dataset (Alberti et al., 2019b; Pan et al., 2019) that only predict the short answer and directly select its paragraph as long answer, RikiNet employs a multi-level cascaded answer predictor which jointly predict the short answer span, the long answer paragraph, and the answer type in a cascaded manner. Another key intuition motivating our design is that even if the relevant documents are not given, humans can easily judge that some questions have no short answers (Borschinger et al., 2019). Take this question as a motivating example:"What is the origin of the Nobel prize?" The answer should be based on a long story, which cannot be easily expressed in a short span of entities. Therefore we also feed the question representation into the predictor as an auxiliary prior to answer type prediction. The details will be given in § 3.2.

On the NQ test set, our single model obtains 74.3 F1 scores on the long-answer task (LA) and 57.9 F1 scores on the short-answer task (SA) compared to the published best single model (Alberti et al., 2019a) results of 66.8 F1 on LA and 53.9 F1 on SA. To the best of our knowledge, RikiNet is the first *single* model that outperforms the single human performance (Kwiatkowski et al., 2019) on both LA and SA. Besides, our ensemble model obtains 76.1 F1 on LA and 61.3 F1 on SA, which achieves the best performance of both LA and SA on the official NQ leaderboard.

2 Preliminaries

Before we describe our model in detail, we first introduce the notations and problem formalization. Our paper considers the following NQ (Kwiatkowski et al., 2019) task: Given a natural question q, a related Wikipedia page p (in the top 5 search results returned by the Google search engine), the model outputs a paragraph within the Wikipedia page p as the *long answer* which contains enough information to infer the answer to the question, and an entity span within the long answer that answers the question as the *short answer*. Also, the short answer of the 1% Wikipedia page is "yes" or "no", instead of a short span. Both long answers and short answers can be `NULL` (*i.e.*, no such answer could be found).

Given a natural question q and its paired Wikipedia page p, we tokenize them with the 30,522 wordpiece vocabulary as used in (Devlin et al., 2018). Following (Alberti et al., 2019b; Pan et al., 2019), we generate multiple document spans by splitting the Wikipedia page with a sliding window. Then, we obtain multiple 6-tuple training instances (q, d, c, s, e, t) for each NQ data pair (q, p), where q and d are wordpiece IDs of question with length n and document span with length m, $c \in \mathbb{S}$ indicates the paragraph index of the long answer where $\mathbb{S}$ is the set that includes all paragraph indexes (*i.e,* all long answer candidates) within d, $s, e \in \{0, 1, ..., m-1\}$ are inclusive indices pointing to the start and end of the short answer span, and $t \in \{0, 1, 2, 3, 4\}$ represents the five answer types, corresponding to the labels "NULL" (no answer), "SHORT" (has short answer), "LONG" (only has long answer), "YES", and "NO".

For each tuple (q, d, c, s, e, t) of the data pair (q, p), RikiNet takes d and q as inputs, and jointly predicts c, s, e, t. Finally we merge the prediction results of every tuple to obtain the final predicted long answer, short answer, and their confidence scores of the data pair (q, p) for evaluation.

3 Methodology

We propose the **RikiNet** which **R**eads the **Wiki**pedia pages for natural question answering. As shown in Fig. 1, RikiNet consists of two modules: (a) the **dynamic paragraph dual-attention reader** as described in §3.1, and (b) the **multi-level cascaded answer predictor** as described in §3.2.

3.1 Dynamic Paragraph Dual-Attention Reader

Dynamic Paragraph Dual-Attention (DPDA) reader aims to represent the document span d and the question q. It outputs the context-aware question representation, question-aware token-level document representation, and paragraph-level document representation, which will be all fed into the predictor to obtain the long and short answers.

3.1.1 Encoding Question and Document Span

We firstly employ a pre-trained language model such as BERT (Devlin et al., 2018) to obtain the initial question representation $Q_0 \in \mathbb{R}^{n \times h}$ and the initial document span representation $D_0 \in \mathbb{R}^{m \times h}$, where h is the hidden size. Similar to (Devlin et al., 2018), we concatenate a "[CLS]" token, the tokenized question q with length n, a "[SEP]" token, the tokenized document span d with length m, and a final "[SEP]" token. Then we feed the resulting sequence into the pre-trained language model.

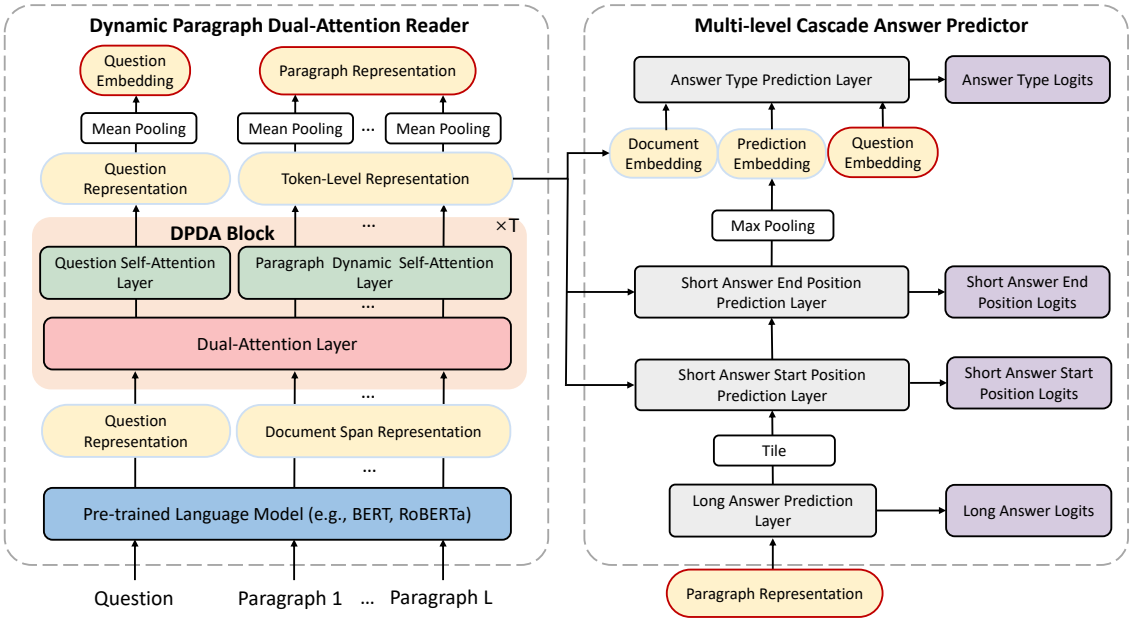

Figure 1: Overview of RikiNet framework.

3.1.2 Dynamic Paragraph Dual-Attention Block

As shown on the left in Fig. 1, DPDA reader contains multiple Dynamic Paragraph Dual-Attention (DPDA) blocks. The first block takes Q_0 and D_0 as the inputs. The outputs $Q_{(t)}$ and $D_{(t)}$ of the t-th block are then fed into the next block. Each block contains three types of layers: the dual-attention layer, the paragraph dynamic self-attention layer, and the question self-attention layer. The last DPDA block outputs the final question and document representations. We describe them in detail now.

Dual-Attention Layer To strengthen the information fusion from the question to the paragraphs as well as from the paragraphs to the question, we adapt a dual-attention mechanism, which has been shown effective in other MRC models (Xiong et al., 2018; Seo et al., 2017; Xiong et al., 2017). We further tweak it by increasing the depth of attention followed by a residual connection (He et al., 2016) and layer normalization (Ba et al., 2016).

In particular, the t-th block first calculates a similarity metric $L_{(t)} \in \mathbb{R}^{m\times n}$ which is then normalized row-wise and column-wise to produce two attention weights: $A^Q_{(t)} \in \mathbb{R}^{m\times n}$, across the document for each token in the question; and $A^D_{(t)} \in \mathbb{R}^{n\times m}$, across the question for each token in the document,

$$
\begin{aligned}
L_{(t)} &= D_{(t-1)} Q_{(t-1)}^\top \in \mathbb{R}^{m\times n}, \\
A^Q_{(t)} &= \mathrm{Softmax}\left(L_{(t)}\right) \in \mathbb{R}^{m\times n}, \\
A^D_{(t)} &= \mathrm{Softmax}\left(L_{(t)}^\top\right) \in \mathbb{R}^{n\times m}.
\end{aligned}
$$

Similar to (Xiong et al., 2017; Seo et al., 2017), we obtain the question-aware representation of the document by

$$
\begin{aligned}
\bar{Q}^C_{(t)} &= \left(D_{(t-1)}^\top A^Q_{(t)}\right)^\top \in \mathbb{R}^{n\times h}, \\
\bar{D}^C_{(t)} &= \left(A^D_{(t)}\right)^\top \left[Q_{(t-1)}; \bar{Q}^C_{(t)}\right] \in \mathbb{R}^{m\times 2h},
\end{aligned}
$$

where $[\cdot;\cdot]$ denotes concatenation. We also obtain the context-aware question representation in a dual way:

$$
\begin{aligned}
\bar{\bar{D}}^C_{(t)} &= \left(Q_{(t-1)}^\top A^D_{(t)}\right)^\top \in \mathbb{R}^{m\times h}, \\
\bar{Q}^C_{(t)} &= \left(A^Q_{(t)}\right)^\top \left[D_{(t-1)}; \bar{\bar{D}}^C_{(t)}\right] \in \mathbb{R}^{n\times 2h}.
\end{aligned}
$$

We finally apply the residual connection and layer normalization to both the question and the document representations with the linear transformations.

$$
\begin{aligned}
D^C_{(t)} &= \mathrm{LayerNorm}\left(D_{(t-1)} + \bar{D}^C_{(t)} W^D_{(t)}\right) \in \mathbb{R}^{m\times h}, \\
Q^C_{(t)} &= \mathrm{LayerNorm}\left(Q_{(t-1)} + \bar{Q}^C_{(t)} W^Q_{(t)}\right) \in \mathbb{R}^{n\times h},
\end{aligned}
$$

where $W^D_{(t)} \in \mathbb{R}^{2h\times h}$ and $W^Q_{(t)} \in \mathbb{R}^{2h\times h}$ are trainable parameters in the dual-attention layer of the t-th block. The document representation $D^C_{(t)}$ will be fed into the paragraph dynamic self-attention layer to obtain the paragraph representation. The question representation $Q^C_{(t)}$ will be fed into the question self-attention layer to get the question embedding.

Question Self-Attention Layer This layer uses a transformer self-attention block (Vaswani et al., 2017) to further enrich the question representation:

$$Q_{(t)} = \text{Transformer}\left(Q^C_{(t)}\right) \in \mathbb{R}^{n\times h},$$

where the transformer block consists of two sub-layers: a multi-head self-attention layer and a position-wise fully connected feed-forward layer. Each sub-layer is placed inside a residual connection with layer normalization. After the last DPDA block, we obtain the final question embedding $\boldsymbol{q} \in \mathbb{R}^h$ by applying the mean pooling,

$$\boldsymbol{q} = \text{MeanPooling}\left(Q^C_{(T)}\right) \in \mathbb{R}^h,$$

where T denotes the number of the DPDA blocks. This question embedding $\boldsymbol{q}$ will be further fed into the predictor for answer type prediction.

Paragraph Dynamic Self-Attention Layer This layer is responsible for gathering information on the key tokens in each paragraph. The token-level representation $D_{(t)}$ is first given by:

$$D_{(t)} = \text{Transformer}\left(D^C_{(t)}\right) \in \mathbb{R}^{m\times h}. \quad (1)$$

The difference from the original multi-head self-attention in (Vaswani et al., 2017) is that we incorporate two extra attention masks, which will be introduced later in Eq. (3) and (4). The last DPDA block applies a mean pooling to the tokens within the same paragraph to obtain the paragraph representation $L \in \mathbb{R}^{l\times h}$ as

$$L[i,:] = \underset{\mathbb{L}_j=i}{\text{MeanPooling}}\left(\left\{D_{(T)}[j,:]\right\}\right) \in \mathbb{R}^h, \quad (2)$$

where l denotes the number of paragraph within the document span d (*i.e.*, the number of long answer candidates within the document span d), $L[i,:]$ is the representation of the i-th paragraph, $D_{(T)}[j,:]$ is the representation of the j-th token at last DPDA block, and $\mathbb{L}_j$ indicates the index number of the paragraph where the j-th token is located.

Tokens in the original multi-head attention layer of the transformer self-attention block attend to all tokens. We introduce two attention masks to the self-attention sub-layer in Eq. (1) based on two key motivations: 1) Each paragraph representation should focus on the question-aware token information inside the paragraph; 2) Most of the answers are only related to a few words in a paragraph. For the first motivation, we introduce the paragraph attention mask $\mathcal{M}^L \in \mathbb{R}^{m\times m}$ which is defined as:

$$\mathcal{M}^L[i,j] = \begin{cases} 0, & \text{if } \mathbb{L}_i = \mathbb{L}_j, \\ -\infty, & \text{otherwise.} \end{cases} \quad (3)$$

It forces each token to only attend to the tokens within the same paragraph. Therefore, each paragraph representation focuses on its internal token information after the mean pooling of Eq. (2).

Based on the second motivation, we dynamically generate another attention mask to select key tokens before self-attention. We use a neural network $\mathcal{F}^\Phi_{(t)}$ called scorer with the Sigmoid activation function to calculate the importance score for each token:

$$\Phi_{(t)} = \mathcal{F}^\Phi_{(t)}\left(D^C_{(t)}\right) \in \mathbb{R}^{m\times 1},$$

Then we obtain the dynamic attention mask $\mathcal{M}^\Phi_{(t)} \in \mathbb{R}^{m\times m}$ by selecting top-K tokens[3]

$$\mathcal{M}^\Phi_{(t)}[i,j] = \begin{cases} 0, & \text{if } i \in \mathbb{S}^\Phi_{(t)} \text{ and } j \in \mathbb{S}^\Phi_{(t)} \\ -\infty, & \text{otherwise,} \end{cases} \quad (4)$$

where $\mathbb{S}^\Phi_{(t)} = \underset{k\in[0,m-1]}{\text{argmax-K}}\left(\left\{\Phi_{(t)}[k]\right\}\right)$. Here $\Phi_{(t)}[k]$ denotes the score of the k-th token at t-th block, K is a hyperparameter, and $\mathbb{S}^\Phi_{(t)}$ is the set that includes the index of the selected top-K tokens. This attention mask lets the paragraph representation concentrate on the selected key tokens.

The final scaled dot-product attention weight $A_{(t)} \in \mathbb{R}^{m\times m}$ of the multi-head self-attention sub-layer (Vaswani et al., 2017) in Eq. (1) with two proposed attention masks can be written as:

$$A_{(t)} = \text{Softmax}\left(\mathcal{M}^\Phi_{(t)} + \mathcal{M}^L + \frac{\left(D^C_{(t)} D^{C\top}_{(t)}\right)}{\sqrt{h}}\right).$$

[3]Following Zhuang and Wang (2019), our implementation pads the unselected token representations with zero embeddings and adds the scorer representation with the linear transformation to $D_{(t)}$ to avoid gradient vanishing for scorer training.

3.2 Multi-level Cascaded Answer Predictor

Due to the nature of the NQ tasks, a short answer is always contained within a long answer, and thus it makes sense to use the prediction of long answers to facilitate the process of obtaining short answers. As shown on the right in Fig. 1, we design a cascaded structure to exploit this dependency. This predictor takes the token representation $D_{(T)}$, the paragraph representation L, and the question embedding $\boldsymbol{q}$ as inputs to predict four outputs in a *cascaded* manner: (1) long answer $\rightarrow$ (2) the start position of the short answer span $\rightarrow$ (3) the end position of the short answer span $\rightarrow$ (4) the answer type. That is, the previous results are used for the next tasks as indicated by the notation "$\rightarrow$".

Long Answer Prediction We employ a dense layer $\mathcal{F}^L$ with $\tanh$ activation function as long answer prediction layer, which takes the paragraph representation $L \in \mathbb{R}^{l\times h}$ as input to obtain the long-answer prediction representation $H^L \in \mathbb{R}^{l\times h}$. Then the long-answer logits $\boldsymbol{o}^L$ are computed with a linear layer

$$H^L = \mathcal{F}^L(L) \in \mathbb{R}^{l\times h},$$
$$\boldsymbol{o}^L = H^L W^L \in \mathbb{R}^l,$$

where $W^L \in \mathbb{R}^{h\times 1}$ is a trainable parameter.

Short Answer Prediction Firstly, we use the long-answer prediction representation H^L and the token representation $D_{(T)}$ as the inputs to predict the start position of the short answer. Then the prediction representation of the start position of the short answer will be *re-used* to predict the end position.

Since the row-dimension of $D_{(T)} \in \mathbb{R}^{m\times h}$ is different from that of $H^L \in \mathbb{R}^{l\times h}$, we cannot directly concatenate the H^L to $D_{(T)}$. We tile the $H^L \in \mathbb{R}^{l\times h}$ with $\bar{H}^L \in \mathbb{R}^{m\times h}$ along the row-dimension: $\bar{H}^L[i,:] = H^L[\mathbb{L}_i,:] \in \mathbb{R}^h$. Note that $\mathbb{L}_i$ indicates the index number of the paragraph where the i-th token is located. Thus, the model can consider the prediction information of the long answer when predicting the short answer. Similarly, the start and end position logits of the short answer are predicted by,

$$H^S = \mathcal{F}^S\left(\left[\bar{H}^L; D_{(T)}\right]\right) \in \mathbb{R}^{m\times h},$$
$$\boldsymbol{o}^S = H^S W^S \in \mathbb{R}^m,$$
$$H^E = \mathcal{F}^E\left(\left[H^S; D_{(T)}\right]\right) \in \mathbb{R}^{m\times h},$$
$$\boldsymbol{o}^E = H^E W^E \in \mathbb{R}^m,$$

where $\boldsymbol{o}^S$ and $\boldsymbol{o}^E$ are the output logit vectors of the start positions and the end positions of the short answer, $\mathcal{F}^S$ and $\mathcal{F}^E$ are two dense layers with $\tanh$ activation function, and $W^S \in \mathbb{R}^{h\times 1}$, $W^E \in \mathbb{R}^{h\times 1}$ are trainable parameters.

Answer Type Prediction Finally, the predictor outputs the answer type. There are five answer types as discussed in § 2. With the observation that humans can easily judge that some questions have no short answers even without seeing the document, we treat the question embedding $\boldsymbol{q} \in \mathbb{R}^h$ as an auxiliary input for the answer type prediction. Besides, the token representation $D_{(T)}$ and the short-answer prediction representation H^E are also used for that prediction:

$$\boldsymbol{d} = \text{MeanPooling}\left(D_{(T)}\right) \in \mathbb{R}^h,$$
$$\boldsymbol{e} = \text{MaxPooling}\left(H^E\right) \in \mathbb{R}^h,$$
$$\boldsymbol{h}^T = \mathcal{F}^T\left([\boldsymbol{d}; \boldsymbol{q}; \boldsymbol{e}]\right) \in \mathbb{R}^h,$$
$$\boldsymbol{o}^T = \text{Softmax}\left(\boldsymbol{h}^T W^T\right) \in \mathbb{R}^5,$$

where $\boldsymbol{o}^T$ is the logits of the five answer types, $\mathcal{F}^T$ is a dense layer with $\tanh$ activation function, and $W^T \in \mathbb{R}^{h\times 5}$ is a trainable parameter.

Training Loss and Inference For training, we compute cross-entropy loss over the above mentioned output logits, and jointly minimize these four cross-entropy losses as:

$$\mathcal{L} = \mathcal{L}^L + \mathcal{L}^S + \mathcal{L}^E + \mathcal{L}^T.$$

During inference, we calculate the final long-answer score Ψ^L for all the paragraphs within the Wikipedia page based on the long-answer logits $\boldsymbol{o}^L$ and the answer type logits $\boldsymbol{o}^T$. The long-answer score of paragraph c can be written as

$$\Psi^L(c) = \boldsymbol{o}^L[c] + \underbrace{\left(\sum_{t=1}^{4} \boldsymbol{o}^T[t] - \boldsymbol{o}^T[0]\right)}_{\text{answer type score}},$$

where $\boldsymbol{o}^T[0]$ denotes the logits where the answer type is "NULL"(no answer), $\sum_{t=1}^{4} \boldsymbol{o}^T[t]$ denotes the sum of the logits where the answer type is not "NULL". The answer type score can be seen as a bias of each document span in the Wikipedia page. Then we select the paragraph of the highest long-answer score Ψ^L over the entire Wikipedia page as the long answer.

Similarly, the short-answer score of the corresponding span (s, e) is calculate by

$$\Psi^S(s,e) = \underbrace{\left(\boldsymbol{o}^S[s] + \boldsymbol{o}^E[e]\right)}_{\text{answer span score}} + \underbrace{\left(\boldsymbol{o}^T[1] - \boldsymbol{o}^T[0]\right)}_{\text{answer type score}},$$

where $\boldsymbol{o}^T[1]$ denotes the score where the answer type is "SHORT"(has short answer). We select the short answer span which has the highest short-answer score Ψ^S within the long answer as the final short answer. We use the official NQ evaluation script to set two separate thresholds for predicting whether the two types of answers are answerable.

4 Experiments

4.1 Dataset

We focus on the Natural Questions (NQ) (Kwiatkowski et al., 2019) dataset in this work. The public release of the NQ dataset consists of 307,373 training examples and 7,830 examples for development data (dev set). NQ provides a blind test set contains 7,842 examples, which can only be accessed through a public leaderboard submission.

4.2 Implementation Details

As discussed in § 2, we generate multiple document spans by splitting the Wikipedia page with a sliding window. Following (Pan et al., 2019; Alberti et al., 2019b), the size and stride of the sliding window are set to 512 and 192 tokens respectively. The average number of document spans of one Wikipedia page is about 22. Since most of the document span does not contain the answer, the number of negative samples (*i.e.,* no answer) and positive samples (*i.e.,* has answers) is extremely imbalanced. We follow (Pan et al., 2019; Alberti et al., 2019b) to sub-sample negative instances for training, where the rate of sub-sampling negative instance is the same as in (Pan et al., 2019). As a result, there are 469,062 training instances in total.

We use Adam optimizer (Kingma and Ba, 2015) with a batch size of 36 for model training. The initial learning rate, the learning rate warmup proportion, the training epoch, the hidden size h, the number of blocks T, and the hyperparameter K are set to 2×10^{-5}, 0.1, 2, 1024, 2, and 256 respectively. Our model takes approximately 24 hours to train with 4 Nvidia Tesla P40. Evaluation completed in about 6 hours on the NQ dev and test set with a single Nvidia Tesla P100.

We use the Google released BERT-large model fine-tuned with synthetic self-training (Alberti et al., 2019a) to encode the document and question as described in § 3.1.1. We also compare the performance of RikiNet which uses the pre-trained RoBERTa $_{\text{large}}$ model (Liu et al., 2019). It should be noted that our RikiNet is orthogonal to the choice of a particular pre-trained language model.

4.3 Main Results

We present a comparison between previously published works on the NQ task and our RikiNet. We report the results of the precision (P), the recall (R), and the F1 score for the long-answer (LA) and short-answer (SA) tasks on both test set and dev set in Tab. 1. The first two lines of Tab. 1 show the results of two multi-passage MRC baseline models presented in the original NQ paper (Kwiatkowski et al., 2019). The third to sixth lines show the results of the previous state-of-the-art models. These models all employ the BERT$_{\text{large}}$ model and perform better than that two baselines. Our RikiNet-BERT$_{\text{large}}$ also employs the BERT$_{\text{large}}$ model, and its single model has achieved a significant improvement over the previously published best model on the test set (LA from 66.8 F1 to 74.3 F1, and SA from 53.9 F1 to 57.9 F1). To the best of our knowledge, this is the first[4] *single* model that surpasses the single human performance (Kwiatkowski et al., 2019) on both LA and SA tasks. We also provide a BERT$_{\text{joint}}$ (Alberti et al., 2019b) + RoBERTa $_{\text{large}}$ (Liu et al., 2019) baseline on NQ, which only replaces the BERT$_{\text{large}}$ in BERT$_{\text{joint}}$ method with RoBERTa $_{\text{large}}$. To be expected, the BERT$_{\text{joint}}$ + RoBERTa $_{\text{large}}$ performs better than original BERT$_{\text{joint}}$. Furthermore, our single model of RikiNet-RoBERTa $_{\text{large}}$ which employs RoBERTa $_{\text{large}}$ model also achieves better performance on both LA and SA, significantly outperforming BERT$_{\text{joint}}$ + RoBERTa $_{\text{large}}$. These results demonstrate the effectiveness of our RikiNet.

Since most submissions on the NQ leaderboard are ensemble models, we also report the results of our ensemble model, which consists of three RikiNet-RoBERTa $_{\text{large}}$ models with different hyper-parameters. At the time of submission (29 Nov. 2019), the NQ leaderboard shows that our ensemble model achieves the best performance on both LA (F1 76.1) and SA (F1 61.3).

[4]The single RikiNet-BERT$_{\text{large}}$ model was submitted to the NQ public leaderboard on 7 Nov. 2019.

	LA Dev			LA Test			SA Dev			SA Test		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DocumentQA (Clark and Gardner, 2018)	47.5	44.7	46.1	48.9	43.3	45.7	38.6	33.2	35.7	40.6	31.0	35.1
DecAtt (Parikh et al., 2016) + DocReader (Chen et al., 2017)	52.7	57.0	54.8	54.3	55.7	55.0	34.3	28.9	31.4	31.9	31.1	31.5
$\text{BERT}_{\text{joint}}$ (Alberti et al., 2019b)	61.3	68.4	64.7	64.1	68.3	66.2	59.5	47.3	52.7	63.8	44.0	52.1
$\text{BERT}_{\text{large}}$ + 4M synth NQ (Alberti et al., 2019a)	62.3	70.0	65.9	65.2	68.4	66.8	60.7	50.4	55.1	62.1	47.7	53.9
$\text{BERT}_{\text{joint}}$ (Alberti et al., 2019b) + $\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019) ‡	65.6	69.1	67.3	-	-	-	60.9	51.0	55.5	-	-	-
$\text{BERT}_{\text{large}}$ + SQuAD2 PT + AoA (Pan et al., 2019)†	-	-	68.2	-	-	-	-	-	57.2	-	-	-
$\text{BERT}_{\text{large}}$ + SSPT (Glass et al., 2019)†	-	-	65.8	-	-	-	-	-	54.2	-	-	-
RikiNet-$\text{BERT}_{\text{large}}$	73.2	74.5	73.9	74.2	74.4	74.3	61.1	54.7	57.7	63.5	53.2	57.9
RikiNet-$\text{RoBERTa}_{\text{large}}$ ‡	74.3	76.4	**75.3**	-	-	-	61.4	57.3	**59.3**	-	-	-
RikiNet-$\text{BERT}_{\text{large}}$ (ensemble)	74.4	76.3	75.4	75.3	75.9	75.6	66.9	53.8	59.6	63.2	56.1	59.5
RikiNet-$\text{RoBERTa}_{\text{large}}$ (ensemble)	73.3	78.7	**75.9**	78.1	74.2	**76.1**	66.6	56.4	**61.1**	67.6	56.1	**61.3**
Single Human (Kwiatkowski et al., 2019)	80.4	67.6	73.4	-	-	-	63.4	52.6	57.5	-	-	-
Super-annotator (Kwiatkowski et al., 2019)	90.0	84.6	87.2	-	-	-	79.1	72.6	75.7	-	-	-

Table 1: Performance comparisons on the dev set and the blind test set of the NQ dataset. We report the evaluation results of the precision (P), the recall (R), and the F1 score for both long-answer (LA) and short-answer (SA) tasks. We use background color to highlight the column of F1 results. † refers to the works that only provide the F1 results on the dev set in their paper. ‡ refers to our implementations where we only report the results on the dev set, due to the NQ leaderboard submission rules (each participant is only allowed to submit once per week).

4.4 Ablation Study

RikiNet consists of two key parts: DPDA reader and multi-level cascaded answer predictor. To get a better insight into RikiNet, we conduct an in-depth ablation study on probing these two modules. We report the LA and SA F1 scores on the dev set.

Ablations of DPDA Reader We keep the predictor and remove the component of the DPDA reader. The results are shown in Tab. 2. In (a), we remove the entire DPDA reader as introduced in § 3.1 except $\text{BERT}_{\text{large}}$. In (b), (c), and (d), we remove the dual-attention layer, question self-attention layer, and paragraph dynamic self-attention layer as described in § 3.1.1 respectively. In (e) and (f), we remove the paragraph attention mask of Eq. (3) and the dynamic attention mask of Eq. (4) respectively. We can see that after removing the DPDA reader, the performance drops sharply. In addition, the paragraph dynamic self-attention layer has the greatest impact on performance. Moreover, both the paragraph attention mask and dynamic attention mask contribute to the performance improvement.

We also change the hyper-parameter K and the number of blocks T. Results show that the setting of $K = 384$ performs better than $K = 512$ (*i.e.*, no dynamic attention mask), and $K = 256$ performs best. For the number of DPDA blocks T, the model achieves the best performance when $T = 2$.

Ablations of Predictor On the predictor side, we further remove or replace its component and report the results in Tab. 3. In (1) we remove the whole DPDA reader and predictor. In (2), we remove the way of multi-level prediction (*i.e.*, training the model to predict long and short answer jointly) described in § 3.2, and follow the previous work (Alberti et al., 2019b) to directly predict the short answer and then select its paragraph as the long answer. We can see that our multi-level prediction is critical to the long answer prediction. In (3) we only remove the cascaded structure but keep the multi-level prediction, which means that the prediction representations are no longer used as input for other predictions, the performance of both long and short answers drops about 1.0 F1 score. In (4) we change the ordering of cascaded process. That is instead of considering long an-

Setting	LA F1	SA F1
RikiNet-$\text{BERT}_{\text{large}}$ (Full)	**73.9**	**57.7**
(a) - DPDA reader	70.7	55.9
(b) - Dual-attention layer	73.1	56.6
(c) - Question self-attention layer	73.5	57.5
(d) - Paragraph self-attention layer	72.2	56.3
(e) - Paragraph attention mask	73.2	57.1
(f) - Dynamic attention mask	72.9	56.8
RikiNet-$\text{BERT}_{\text{large}}$ ($K = 512$)	72.9	56.8
RikiNet-$\text{BERT}_{\text{large}}$ ($K = 384$)	73.7	57.3
RikiNet-$\text{BERT}_{\text{large}}$ ($K = 256$)	**73.9**	**57.7**
RikiNet-$\text{BERT}_{\text{large}}$ ($K = 128$)	73.7	56.9
RikiNet-$\text{BERT}_{\text{large}}$ ($T = 0$)	70.7	55.9
RikiNet-$\text{BERT}_{\text{large}}$ ($T = 1$)	73.6	57.6
RikiNet-$\text{BERT}_{\text{large}}$ ($T = 2$)	**73.9**	**57.7**
RikiNet-$\text{BERT}_{\text{large}}$ ($T = 3$)	73.5	57.1
RikiNet-$\text{BERT}_{\text{large}}$ ($T = 4$)	73.0	56.9

Table 2: Ablations of DPDA reader on dev set of NQ dataset.

Setting	LA F1	SA F1
RikiNet-BERT$_{\text{large}}$ (Full)	**73.9**	**57.7**
(1) - DPDA reader & Predictor	65.9	55.1
(2) - Multi-level prediction	70.9	57.1
(3) - Cascaded structure	73.0	56.7
(4) + S2L cascaded structure	73.6	57.5
(5) - Question embedding	73.4	57.4
(6) - Tanh dense prediction layer	73.2	57.3
(7) + Bi-LSTM prediction layer	73.3	57.4
(8) + Transformer prediction layer	73.5	57.5
(9) + GELU dense prediction layer	73.7	57.6

Table 3: Ablations of multi-level cascaded predictor on dev set of NQ dataset.

swer first and then short answer as described in § 3.2, we consider the cascaded structure of short answer first and then long answer. However, we get slightly worse results in this way. In (5), we remove the question embedding which is used for answer type prediction. It can be observed that the question embedding contributes to performance improvement. In the variants of (6)-(9), we remove the dense prediction layers with Tanh activation function and replace it with Bi-directional Long-Short Term Memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) layers, transformer self-attention blocks, and dense prediction layers with Gaussian Error Linear Unit GELU (Hendrycks and Gimpel, 2016) activation function but neither get better performance.

Overall, both proposed DPDA reader and multi-level cascaded answer predictor significantly improve the model performance.

5 Related Works

Natural Questions (NQ) dataset (Kwiatkowski et al., 2019) has been recently proposed, where each question is paired with an entire Wikipedia page which is a long document containing multiple passages. Although BERT (Devlin et al., 2018) based MRC models have surpassed human performance on several MRC benchmark datasets (Lan et al., 2019; Devlin et al., 2018; Liu et al., 2019; Rajpurkar et al., 2018), a similar BERT method (Alberti et al., 2019b) still has a big gap with human performance on NQ dataset.

There are several recently proposed deep learning approaches for multi-passage reading comprehension. Chen et al. (2017) propose DrQA which contains a document retriever and a document reader (DocReader). Clark and Gardner (2018) introduce Document-QA which utilizes TF-IDF for paragraph selection and uses a shared normalization training objective. De Cao et al. (2019) employ graph convolutional networks (GCNs) for this task. Zhuang and Wang (2019) design a gated token-level selection mechanism with a local convolution. In contrast, our RikiNet considers multi-level representations with a set of complementary attention mechanisms.

To solve the NQ task, Kwiatkowski et al. (2019) adapt Document-QA (Clark and Gardner, 2018) for NQ, and also utilizes DecAtt (Parikh et al., 2016) for paragraph selection and DocReader (Chen et al., 2017) for answer prediction. BERT$_{\text{joint}}$(Alberti et al., 2019b) modifies BERT for NQ. Besides, some works focus on using data augmentation to improve the MRC models on NQ. Alberti et al. (2019a) propose a synthetic QA corpora generation method based on roundtrip consistency. Glass et al. (2019) propose a span selection method for BERT pre-training (SSPT). More recently, Pan et al. (2019) introduce attention-over-attention (Cui et al., 2017) into the BERT model. Pan et al. (2019) also propose several techniques of data augmentation and model ensemble to further improve the model performance on NQ. Although the use of data augmentation and other advanced pre-trained language models (Lan et al., 2019) may further improve model performance, as this is not the main focus of this paper, we leave them as our future work. Our RikiNet is a new MRC model designed tailored to the NQ challenges and can effectively represent the document and question at multi-levels to jointly predict the answers, which significantly outperforms the above methods.

6 Conclusion

We propose the RikiNet, which reads the Wikipedia pages to answer the natural question. The RikiNet consists of a dynamic paragraph dual-attention reader which learns the token-level, paragraph-level and question representations, and a multi-level cascaded answer predictor which jointly predicts the long and short answers in a cascade manner. On the Natural Questions dataset, the RikiNet is the first single model that outperforms the single human performance. Furthermore, the RikiNet ensemble achieves the new state-of-the-art results at 76.1 F1 on long-answer and 61.3 F1 on short-answer tasks, which significantly outperforms all the other models on both criteria.

Acknowledgment

This work is supported by National Natural Science Fund for Distinguished Young Scholar (Grant No. 61625204) and partially supported by the Key Program of National Science Foundation of China (Grant No. 61836006).

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019a. Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.

Chris Alberti, Kenton Lee, and Michael Collins. 2019b. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Benjamin Borschinger, Jordan Boyd-Graber, Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Michelle Chen Huebscher, Wojciech Gajewski, Yannic Kilcher, Rodrigo Nogueira, and Lierni Sestorain Saralegu. 2019. Meta answering for machine reading. *arXiv:1911.04156*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL*.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *ACL*.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *ACL*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *NAACL-HLT*.

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

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, GP Bhargav, Dinesh Garg, and Avirup Sil. 2019. Span selection pre-training for question answering. *arXiv preprint arXiv:1909.04120*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering. *arXiv preprint arXiv:1909.10772*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *TACL*, 7:453–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lin Pan, Rishav Chakravarti, Anthony Ferritto, Michael Glass, Alfio Gliozzo, Salim Roukos, Radu Florian, and Avirup Sil. 2019. Frustratingly easy natural question answering. *arXiv preprint arXiv:1909.05286*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *TACL*, 7:249–266.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

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 *NIPS*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *ICLR*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2018. Dcn+: Mixed objective and deep residual coattention for question answering. In *ICLR*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, and Hai Zhao. 2019. Sg-net: Syntax-guided machine reading comprehension. *arXiv preprint arXiv:1908.05147*.

Yimeng Zhuang and Huadong Wang. 2019. Token-level dynamic self-attention network for multi-passage reading comprehension. In *ACL*.