Multi-paragraph Reading Comprehension with Token-level Dynamic Reader and Hybrid Verifier

1st Yilin Dai, 2st Qian Ji, 3st Gongshen Liu*, 4st Bo Su*
School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University,
Shanghai, China
{lydai1108, jeicy_good, lgshen, subo}@sjtu.edu.cn

***Abstract*—Multi-paragraph reading comprehension requires the model to infer answers of arbitrary user-generated questions by reasoning cross-passage information. Previous work usually generates answer by directly employing a pointer network to predict the start and end position of the answer. However, span-level reading is insufficient since intermediate words may matter more. In this paper, we propose a novel unified network that includes a selector, a Token-level dynamic reader, and a Hybrid verifier (TH-Net). The core of token-level dynamic reader is a gate mechanism which dynamically selects important intermediate words according to boundary words. We decide the reader score from each token being both the boundary and the content. Moreover, we adopt a hybrid network verifier considering semantic answer-answer and entailment question-answer relationships to robust the model in case of being fooled by adversarial answers. Our experiments on SQuAD-document, SQuAD-open, and Trivia-wiki datasets show significant and consistent improvement as compared to other baselines and achieve the state-of-the-art performance on two of them.**

***Index Terms*—reading comprehension, TH-Net, token-level dynamic reader, hybrid verifier**

(a)	(b)
Question: What system did Tesla recommend to Niagara Falls in 1893?	**Question:** What are the main sources of primary law?
Paragraph 1: […], Tesla advised Adams that **a two-phased system** would be the most reliable and that there was a Westinghouse system to light incandescent bulbs using two-phase alternating current.	**Paragraph1:** European Union law is a body of treaties and legislation, which have direct effect or indirect effect on the laws of European Union member states. The three sources of European Union law are **primary law, secondary law and supplementary law**.
Paragraph 2: The acquisition of a feasible AC motor gave Westinghouse a key patent in building **a completely integrated AC system**, but the financial strain of buying up patents and […].	**Paragraph2:** The main sources of primary law are the **Treaties establishing the European Union**. Secondary sources include **regulations and directives** which are based on the Treaties. Secondary sources include regulations and directives which are based on the Treaties.
Paragraph 3: Based on Tesla's advice that they could build **a complete AC system** at the Columbian Exposition, a contract for building a two-phase AC generating system was awarded to Westinghouse Electric.	**Paragraph 3:** The primary law of the EU consists mainly of **the founding treaties**, the "core" treaties being **the Treaty on European Union (TEU) and the Treaty on the Functioning of the European Union (TFEU)**.
Original BERT prediction: a completely integrated AC system	**Original BERT prediction:** the founding treaties
Correct answer: a two-phased system	**Correct answer:** Treaties establishing the European Union

Fig. 1. Two examples of question and paragraphs from SQuAD-document (ID: 56e0812c231d4119001ac217 and 5725b7f389a1e219009abd5e), with candidate answers in a bold font and keywords underlined. (a) A question suffered from boundary similarity. (b) A question suffered from content similarity.

I. Introduction

Machine reading comprehension (MRC) models empower machines to answer user-generated questions by comprehending textual data. In real-world scenarios, passage may be extended and include both relevant and irrelevant content. Since multi-paragraph MRC task being more applicable, our method focuses on addressing the challenges coming with document-level data instead of single-paragraph data.

Existing multi-paragraph MRC models can be divided into two basic approaches: pipelined methods and unified methods. Both typically consist of a paragraph selector for choosing relevant paragraphs from document, a paragraph reader for extracting answers from chosen paragraphs, and an answer verifier for ruling out noisy answers. In pipelined approaches [1], [2], these three components are considered separate and trained independently, but high-quality upstream outputs may not necessarily benefit downstream modules. Unified methods directly apply the model to the input and return the answer with the highest score [3]. Three components share the same contextualized text representation and optimize simultaneously in a joint learning method [4], [5]. This paper adopts unified model to avoid inconsistent performance across different components. Recently, some work replaces word and character embeddings [6] with outputs from pre-trained language models (LMs) to get deeper word representations [7], [8], [9]. Our model follows this approach, but is fine-tuned in training.

Among three modules mentioned above, paragraph reader undoubtedly plays the most pivotal part because of the following reasons. Paragraph selector can be regarded as a binary classification task which can show good results by a linear network. To some extent, it exists to fulfill the document-level data input requirements [10] because ruling out irrelevant paragraphs at first can avoid problems of out-of-memory (OOM) in paragraph reading. Answer verifier also has impact on the performance because unified models without verifier [11], [12] can be easily fooled by adversarial examples [13]. However, paragraph reader remains fundamental to MRC models.

To sequence generating task which specifically selects only a member of the input sequence as the output, pointer network [14] is normally adopted to avoid generating words not known as prior. In MRC task, we usually apply it after question and context embeddings to predict the probability of each token being the start and the end of answer [15], [16]. Although many deep neural networks such as Bi-LSTM [4] and Bi-GRU [5] that can extract deep query-aware context information have brought considerable progress in the performance of reader, few has considered the direct application of pointer network

*Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

to span-extracting task. Only boundary word scores are not enough to measure the full validity and legitimacy of predicted answers. We found that wrong answers may have the same boundary words but different intermediate words with correct ones under many circumstances, which is shown in Fig. 1(a). So we consider taking answer content into consideration in paragraph reader can raise model performance.

In answer verifier, most modules rule out noisy candidate answers which have lower correlation degree with the question. It is obvious that higher their relevance is, higher the possibility of the answer being correct. Another observation can be utilized is that many candidate answers tend to contain the same words and look similarly, as is shown in Fig. 1(b). The correct answer has evidence content that can match both relationships, while others cannot, so we think additionally mining deeper semantic information between candidate answers themselves may also help determine the final answer.

Problems mentioned above both arise from the situation when the model needs to differentiate similar candidate answers, so we propose token-level dynamic reader and hybrid verifier in vanilla unified MRC models to avoid boundary and content similarity. In token-level dynamic reader, we combine the probability of each token being the answer content with span-level prediction. In specific, we adopt a gate mechanism to automatically extract semantic information of intermediate tokens and select important ones. We then replace the representations of these tokens with the self-attention output performed over themselves. After that, another linear network is followed to predict whether each word should be contained in the answer. Finally, we generate candidate answers based on the sum of answer start, between, and end scores. To make this answer between score play an appropriate role, we introduce an auxiliary weighted between loss to help it fuse with span-level reader better. In hybrid verifier, we investigate a novel architecture which effectively extracts semantic information between candidate answers. This semantic relationship is combined with entailment information between each candidate answer and the question to choose the correct answer.

This paper proposes a novel unified MRC network including a paragraph selector, a Token-level dynamic paragraph reader, and a Hybrid answer verifier (TH-Net) detailed in Fig. 2. Our model is evaluated on SQuAD-document, SQuAD-open, and Trivia-wiki datasets, and achieves the state-of-the-art (SOTA) performance on two of them. The main contributions can be summarized as follows:

- Unified approach is adopted to raise the overall performance of three components by sharing the same contextual embeddings. These three components are initialized by pre-trained LM and optimized simultaneously in training procedure.
- We introduce token-level dynamic reader which decides candidate answer scores by both boundary and intermediate tokens. This strategy proves to be very effective in paragraph reading because it can better tackle with the problem of insufficient representation of answers chosen by span-level reader.

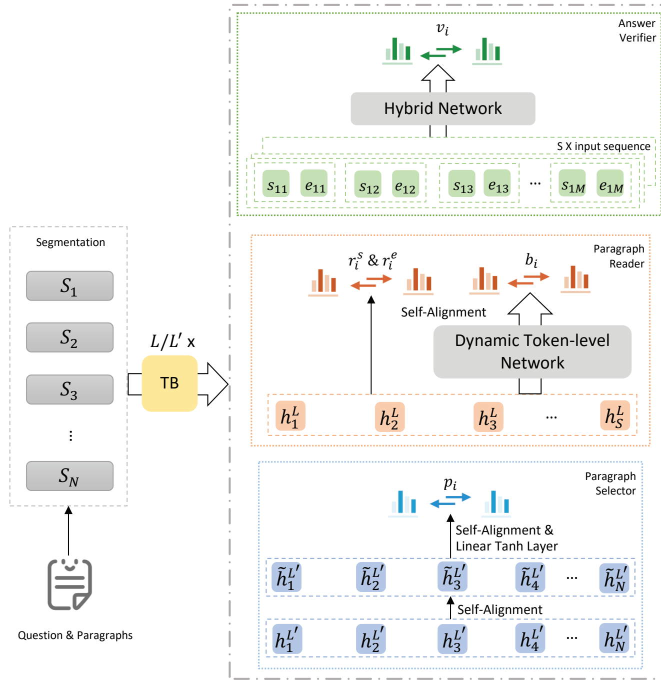

Fig. 2. Architecture overview of TH-Net.

- In answer verifying, we adopt a hybrid network which combines correlated semantic relationship between candidate answers with entailment relationship between question and answer. This mechanism also raises the performance of our verifier.

II. Methods

The overall framework of TH-Net is demonstrated in Fig. 2 where one question and several paragraphs are given as the input and the final answer is returned. TH-Net is initialized by pre-trained LM and fine-tuned during training, with three major modules.

A. Segmentation and Encoding

This layer encodes the input sequence by several transformer blocks and computes a deep and context-aware representation for each token. Before encoding, we concatenate all the input paragraphs to a new document and split it to several segments. Specifically, we produce N paragraph segments using a sliding window of length l and stride r over the new document following [17]. Here we define the *input sequence* as obtained by packing each paragraph P_i and its corresponding question Q, with length $L_x = L_p + L_q + 3$, i.e.,

$$S_i = [< CLS >, Q, < SEP >, P_i, < SEP >] \quad (1)$$

where L_p and L_q are the length of input paragraph and question. Following [18], we take token $< CLS >$ for classification but will not be used in our paper and $< SEP >$ separating question and paragraph. The input representation for the j^{th} token in sequence S_i is constructed as:

$$h_{ij}^0 = s_{ij}^{tok} + s_{ij}^{pos} + s_{ij}^{seg} \quad (2)$$

where s_{ij}^{tok}, s_{ij}^{pos}, and s_{ij}^{seg} are the token, position, and segment embeddings separately. In detail, tokens with the same position share the same position embedding. Besides, all the tokens in question Q and paragraph P_i share the same segment embedding respectively. The input sequence is then fed into L successive transformer encoder blocks to generate deep and context-aware representations. The output for the j^{th} token in in sequence S_i is shown in (3). For the details of transformer block, we refer readers to [19].

$$h_{ij}^{l} = \textbf{TransformerBlock}(h_{ij}^{l-1}), l = 1, 2, \ldots, L \quad (3)$$

B. Paragraph Selector

In multi-paragraph MRC, the golden answer usually comes from one paragraph or even a small set of sentences [20], so we annotate which paragraph contains the answer in a distantly supervised setup. Here, we introduce paragraph selector to select top S paragraphs. Since reading and generating several candidate answers for each paragraph may cause OOM problems, so we use the hidden states of the first L' transformer blocks $h_i^{L'} = \{h_{ij}^{L'}\}_{j=1}^{L_x}$ ($h_{ij}^{L'} \in \mathbb{R}^h$, h refers to the hidden state dimension) as the input of paragraph selector. For each input sequence, we self-align the deep semantic representation learnt afore-mentioned to obtain a weighted sequence vector $\widetilde{h}_i^{L'} \in \mathbb{R}^h$ followed by a linear projection with activation function for a selector score $p_i \in \mathbb{R}$ as:

$$\mu_i = \mathrm{softmax}(w_\mu^\top h_i^{L'}) \quad (4)$$

$$\widetilde{h}_i^{L'} = \sum_{j=1}^{L_x} \mu_{ij} h_{ij}^{L'} \quad (5)$$

$$p_i = w_p^\top \tanh(W_p \widetilde{h}_i^{L'}) \quad (6)$$

where trainable parameters w_μ and w_p are vectors; W_p is a bilinear projection matrix matches two vectors in the same space. We then normalize p_i and optimize the following objective function:

$$\mathcal{L}_{PS} = -\sum_{i=1}^{N} [y_i LS(p_i) + (1 - y_i)(1 - LS(p_i))] \quad (7)$$

where label $y_i = 1$ means S_i contains one golden answer, otherwise $y_i = 0$. Here, $LS(\cdot)$ refers to log_softmax function.

C. Token-level Dynamic Paragraph Reader

As the core module of TH-Net, this layer is designed to comprehend S paragraphs from selector and return M candidate answers for each paragraph with scores calculated on token-level instead of span-level. Different from paragraph selector, we take the output of L transformer blocks as the input since deeper neural networks with attention-mechanism are supposed to capture more complex and useful linguistic phenomena. Following previous work [15], the probability of each word being the answer start position $r_i^s \in \mathbb{R}^{L_x}$ and end position $r_i^e \in \mathbb{R}^{L_x}$ can be easily obtained by applying a linear projection layer with activation function after hidden states $h_i^L \in \mathbb{R}^{L_x \times h}$ as follows:

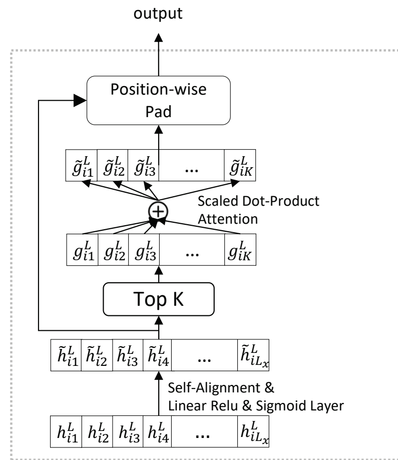

Fig. 3. Architecture of token-level dynamic network in reader.

$$r_i^s = w_s^\top h_i^L, r_i^e = w_e^\top h_i^L \quad (8)$$

where w_s and w_e are vectors to be trained.

In order to make full use of each token, we additionally introduce a token-level dynamic network to obtain answer between score indicating the probability of each word being the content of the answer, detailed in Fig. 3. Our token-level reader is dynamic because it can automatically select important tokens according to the boundary tokens. Here, we define a token *important* if it includes necessary information to decide a candidate answer correct or not. To distinguish different roles of boundary and intermediate tokens play, we use new hidden states $\widetilde{h}_i^L$ obtained by adding two linear projection layers with activation function to h_i^L.

$$\widetilde{h}_i^L = \mathrm{sigmoid}(w_b^\top \mathrm{relu}(W_b^\top h_i^L)) \quad (9)$$

where w_b and W_b are trainable parameters.

A gate mechanism is used to select most important K words $g_i^L \in \mathbb{R}^{K \times h}$ according to $\widetilde{h}_i^L$. K changes along with the length of answer. Attended output matrix $\tilde{g}_i^L$ is obtained by performing scaled dot-product self-attention mechanism over chosen important tokens and position-wise pad is used between $\widetilde{h}_i^L$ and $\widetilde{g}_i^L$ to get the probability of each token being the content of answer $b_i \in \mathbb{R}^{L_x}$:

$$\tilde{g}_i^L = \mathrm{softmax}(\frac{Q_g K_g^T}{\sqrt{h}}) V_g \quad (10)$$

$$b_i = w_g^\top pad[\tilde{h}_i^L; \tilde{g}_i^L] \quad (11)$$

where query $Q_g \in \mathbb{R}^{K \times h}$, $K_g \in \mathbb{R}^{K \times h}$, and value $V_g \in \mathbb{R}^{K \times h}$ are linear projections of g_i^L; w_g is a trainable vector.

In a distantly supervised setup, we label all text spans that match the answer text as being correct [5], thus yielding start and end label vectors for each input sequence as $y_i^s \in \mathbb{R}^{L_x}$ and $y_i^e \in \mathbb{R}^{L_x}$. Besides, we also label all the words between

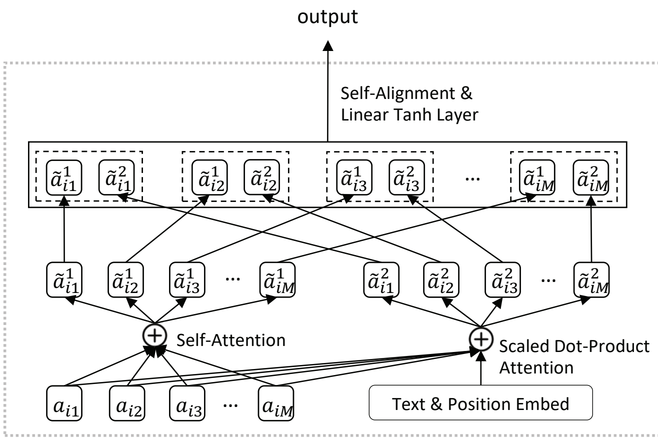

Fig. 4. Architecture of hybrid network in verifier.

the boundary correct to obtain between label vector $y_i^b \in \mathbb{R}^{L_x}$. The sum objective function can be defined as follows:

$$\mathcal{L}_{\text{boundary}} = -\frac{1}{N}\frac{1}{L_x}\sum_{i=1}^{N}\sum_{j=1}^{L_x}[y_{ij}^s LS(r_{ij}^s) + y_{ij}^e LS(r_{ij}^e)] \quad (12)$$

$$\mathcal{L}_{\text{between}} = -\frac{1}{N}\frac{1}{L_x}\sum_{i=1}^{N}\sum_{j=1}^{L_x} y_{ij}^b LS(b_{ij}) \quad (13)$$

$$\mathcal{L}_{PR} = \lambda_1 \mathcal{L}_{\text{boundary}} + \lambda_2 \mathcal{L}_{\text{between}} \quad (14)$$

where λ_1 and λ_2 are two hyper-parameters control the weights of answer boundary function and answer between function.

D. Hybrid Answer Verifier

This module can effectively prune noisy answers out of candidate answers by constructing a hybrid network of combining two models. Model **I** aims to capture the semantic relationship between candidate answers, and model **II** finds the entailment relationship between candidate answer and input sequence.

Model **I** obtains candidate answer representations based on weighted self-aligned vectors which is similar to [21], [22], but utilizes the between-word score. To verify M candidate answers, the m^{th} attended answer representation of i^{th} sequence $\tilde{a}_{im}^1 \in \mathbb{R}^h$ after self-attention is computed via:

$$a_{im} = \frac{1}{e_{im} - s_{im} + 1}\sum_{j=s_{im}}^{e_{im}} (b_i \odot h_i^L)_j \quad (15)$$

$$\gamma_{im,in} = \exp(a_{im}^\top a_{in}) / \sum\nolimits_{j=1}^{M} exp(a_{im}^\top a_{ij}) \quad (16)$$

$$\tilde{a}_{im}^1 = \sum_{j=1}^{M} \gamma_{im,ij} a_{ij} \quad (17)$$

where s_{im} and e_{im} are the start and end index of m^{th} answer for sequence S_i; $\odot$ means position-wise multiplication.

Model **II** captures answer-sequence relationship, thus producing both question and context aware answer representation. We compute the scaled dot product attention of each candidate answer with corresponding input sequence as attention weights and then normalize it to attended vectors as follows:

$$\tilde{a}_{im}^2 = \text{softmax}\left(\frac{Q_a K_a^T}{\sqrt{h}}\right) V_a \quad (18)$$

where query $Q_a \in \mathbb{R}^{L_m \times h}$ is linear projection of the answer representation; key $K_a \in \mathbb{R}^{L_x \times h}$ and value $V_a \in \mathbb{R}^{L_x \times h}$ are linear projections of the input sequence. Here, L_m refers to the maximum candidate answer length.

The verify score $v_i \in \mathbb{R}^M$ is calculated by concatenating the output of both models followed by a linear projection layer. Before concatenating, $\tilde{a}_{im}^2$ will do mean function over the answer length dimension to be the same shape as $\tilde{a}_{im}^1$.

$$v_i^m = w_v \tanh(W_v[\tilde{a}_{im}^1; \tilde{a}_{im}^2])) \quad (19)$$

$$\mathcal{L}_{AV} = -\frac{1}{N}\frac{1}{M}\sum_{i=1}^{N}\sum_{m=1}^{M} y_{im} LS(v_i^m) \quad (20)$$

where w_v and W_v are trainable parameters; $y_i \in \mathbb{R}^M$ is the ground truth label; $[;]$ refers to matrix concatenation.

E. Joint Training and Prediction

According to the design described above, we train these three modules together as multi-task learning [4] with a joint objective function formulated as follows:

$$\mathcal{L} = \mathcal{L}_{PS} + \mathcal{L}_{PR} + \mathcal{L}_{AV} \quad (21)$$

When predicting the final answer, we first calculate selector score for each input sequence and choose top-S paragraphs. Then for each paragraph, we generate M candidate answers with both boundary score and content score. Content score is calculated by using a dynamic gate mechanism which particularly considers important words in answer span. We also prune noisy candidate answers through a hybrid verifier. Therefore, the final score for m^{th} candidate answer of i^{th} input sequence can be calculated by considering selector score, reader score, and verifier score with different weights:

$$\text{score}_i^m = \eta_1 p_i + \eta_2 (r_i^s + r_i^e + b_i)^m + \eta_3 v_i^m \quad (22)$$

where p_i means the score of i^{th} paragraph containing the answer; $r_i^s + r_i^e$ and b_i refer to the answer boundary and content score respectively; v_i^m represents the confidence of m^{th} answer being correct; η_1, η_2, and η_3 are hyper-parameters control the weights of these three components.

III. Experiments

A. Datasets

We experiment on three well-studied open extractive MRC datasets: SQuAD-document, a variant of SQuAD [23] that includes a collection of crowdsourced questions and a full Wikipedia article for each question; SQuAD-open [24], the same dataset but pair each question with the entire Wikipedia domain; Trivia-wiki [25], a dataset of questions from trivia databases associated with Wikipedia articles by completing a web search of the questions. All datasets use Exact Match (EM) accuracy and (marco-averaged) F1 score as the evaluation metrics.

TABLE I
PERFORMANCE OF TH-NET ON SQUAD-DOCUMENT DATASET. THE RESULTS ARE REPORTED ON THE DEV SET.

Model	SQuAD-document EM	F1
Baseline [15]	60.59	66.87
S-Norm [5]	64.08	72.37
$\text{BERT}_{\text{BASE}}$ [18]	68.32	76.39
$\text{RE}^3\text{QA}_{\text{BASE}}$ [12]	75.71	82.66
TH-Net	**77.51**	**84.29**

TABLE II
PERFORMANCE OF TH-NET ON SQUAD-OPEN DATASET. THE RESULTS ARE REPORTED ON THE DEV SET.

Model	SQuAD-open EM	F1
DS-QA [3]	28.71	36.63
R3 [11]	29.18	37.65
MINIMAL [20]	34.72	42.53
Multi-Step [26]	31.95	39.29
$\text{BERT}_{\text{SERNI}}$ [9]	38.61	46.15
$\text{RE}^3\text{QA}_{\text{BASE}}$ [12]	38.54	47.22
TH-Net	**40.18**	**48.49**

B. Experiment Setups

Data Sampling Before training, we first sample several paragraphs for each question by TF-IDF [5], a traditional information retrieval method measuring the distance between the question and paragraph. It is conducted between the textual metadata of question and paragraphs from the same document, including the document title and main content. Besides, other features such as the recall ratio of the question words from the paragraph are also considered to indicate the relevance. As a result, we can obtain several paragraphs relevant to the question to satisfy the multi-paragraph setting in this paper. For SQuAD-document, we use the top 4 paragraphs, and for Trivia-wiki we use the top 8 because much more instance is given. Besides, we also merge consecutive paragraphs in Trivia-wiki to a maximum of 400 words as in [5].

Implementation Details Our model is initialized by a publicly available uncased base version of BERT, so we set the input sequence length L_x as 384, stride r as 128. We choose the number of transformer blocks for selector L' as 3, and for reader and verifier L as 12. The number of selected paragraphs S and candidate answers M are the key factors to balance the effectiveness and efficiency tradeoff. We choose S=4 and M=10 for the good performance when evaluating on the dev set. We optimize the model by Adam optimizer for finetuning 2 epochs, with the minibatch size as 16 and the initial learning rate as 0.0005. During training, we set the weights of intermediate words λ_2 in (14) as 0.2 and 0.1 for SQuAD-document and Trivia-wiki respectively. During inference, we tune the weights for three major modules, and set η_1 as 1.4, η_2 as 1.2, and η_3 as 1.

Comparison Setting We start with a baseline following the model used in [15]. We also take a pipelined BERT finetuned by datasets sampled in this paper as a direct comparison. Our BERT follows exactly the same design as the original paper [18]. Besides, we further take several top-ranked systems on each dataset as additional comparisons (will be detailed in §4).

IV. RESULTS AND ANALYSIS

A. Experiment results

The results of TH-Net on SQuAD-document and SQuAD-open are summarized in Table. I and Table. II. We can see that by adopting the proposed method, our model achieves 77.51 EM and 84.29 F1, outperforming the previous SOTA methods. Note that the $\text{BERT}_{\text{BASE}}$ has obtained only 76.39 F1, which is 7.9 lower than us and validates the effectiveness of combing selector, reader, and verifier in a unified method to address multi-paragraph MRC task. TH-Net also outperforms RE^3QA which adopts a unified architecture similar to our model by 1.8 EM and 1.63 F1, proving that token-level dynamic reader and hybrid verifier have considerable impact on the performance boost. Besides, we also run experiments on SQuAD-open dataset, and TH-Net surpasses a BERT baseline by 2.34 F1 and RE^3QA by 1.27 F1. The performance boost is not so obvious as in SQuAD-document maybe because questions in open scenario situations are more independent and paying more attention to context-question relationship is not so effective.

TABLE III
PERFORMANCE OF TH-NET ON TRIVIA-WIKI DATASET. THE RESULTS ARE REPORTED ON THE TEST SET.

Model	Trivia-wiki Full EM	F1	Trivia-wiki Verified EM	F1
Baseline [15]	40.32	45.91	44.86	50.71
Smartnet [27]	42.41	48,84	50.51	55.90
Re-Ranker [1]	46.94	52.85	62.83	70.68
S-Norm [5]	63.99	68.93	67.98	72.88
SLQA [2]	66.59	70.46	74.83	77.78
TH-Net	**68.55**	**72.77**	**76.45**	**79.38**

We additionally evaluate our model on Trivia-wiki and the result is shown in Table. III. As we can see, TH-Net achieves 68.55 EM and 72.77 F1, outperforming the previous methods. However, the score of 76.45 EM and 79.38 F1 on the verified version is lower than the SOTA performance of 76.7 EM and 79.9 F1 in [26], which implies our model still can be improved.

B. Discussion

Ablation study To get better insight into our model architecture, we conduct an in-depth ablation study on SQuAD-

TABLE IV
COMPARISON OF TH-NET WITH DIFFERENT INDIVIDUAL COMPONENTS ON SQUAD-DOCUMENT AND TRIVIA-WIKI.

Model	SQuAD-document EM	F1	Trivia-wiki EM	F1
Complete Model	**77.51**	**84.29**	**68.55**	**72.77**
- selector	76.07	83.50	67.27	71.64
- token-level reader	76.49	83.81	68.26	72.50
- hybrid-network verifier	76.98	84.05	67.84	72.25
- model **I**	76.80	83.85	68.03	72.30
- model **II**	76.84	83.93	68.10	72.42

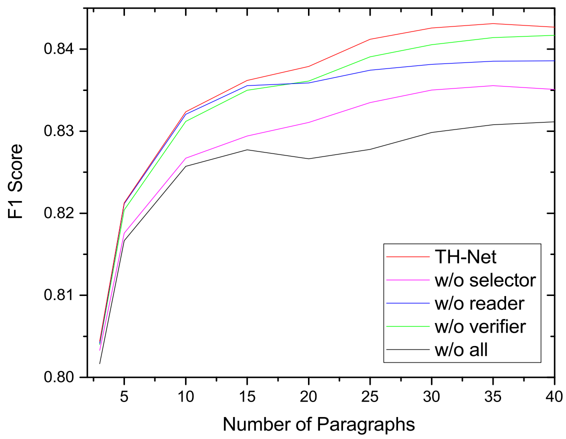

Fig. 5. F1 score on SQuAD-document w.r.t different number of paragraphs.

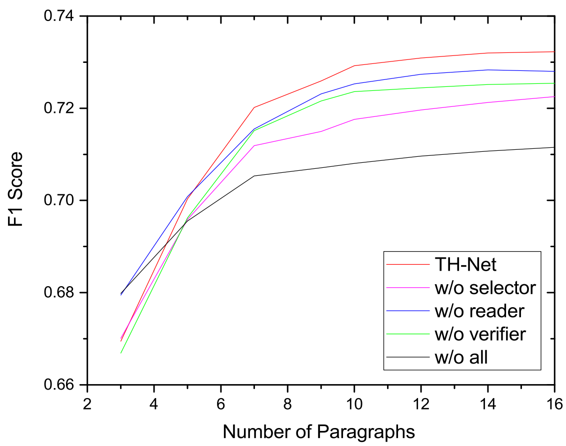

Fig. 6. F1 score on Trivia-wiki w.r.t different number of paragraphs.

document and Trivia-wiki in Table. IV. To ablate selector, we choose paragraphs based on TF-IDF scores. To evaluate token-level dynamic reader, we generate answers only based on span-level scores. The performance of hybrid verifier is shown by selecting answers only considering the selector and reader scores. Ablating selector degrades the performance by 0.79 F1 and 1.13 F1 for SQuAD-document and Trivia-wiki, which proves that selector can prune irrelevant paragraphs effectively and efficiently in the first stage. Removing token-level reader results in a performance drop by 0.48 F1 and 0.27 F1, indicating that the intermediate-word mechanism can improve the performance of span-level reader. Ablating the hybrid verifier, on the other hand, causes little influence on F1 by 0.24 and 0.52 but big on EM by 1.44 and 1.28. This suggests that the hybrid network can elevate the general performance by outputting more precise answers.

Besides, We also measure the performance of our proposed approaches as the model is used to independently process an increasing number of paragraphs, which is shown in Fig. 5 and Fig. 6. We can observe that with more paragraphs to be dealt with, all curves become stable showing it does a passable job at focusing on the correct paragraph. Moreover, the token-level dynamic reader and hybrid verifier do have effect on the model performance boost. In SQuAD-document, reader plays a bigger part, while verifier becomes more important in Trivia-wiki when dealing with more paragraphs. In both datasets, the token-level dynamic reader and hybrid verifier lead to a significant improvement, and selector is even better.

TABLE V
PERFORMANCE OF TH-NET ON SQUAD-DOCUMENT AND TRIVIA-WIKI W.R.T DIFFERENT BOUNDARY AND INTERMEDIATE WORD WEIGHTS.

λ_1	λ_2	**SQuAD-document** EM	F1	**Trivia-wiki** EM	F1
0.5	0	77.05	84.18	68.50	72.69
0.45	0.1	77.30	84.21	**68.55**	**72.77**
0.4	0.2	**77.51**	**84.29**	68.39	72.45
0.35	0.3	77.06	83.85	68.17	72.21
0.3	0.4	76.69	83.55	67.95	72.06

Effect of token-level dynamic reader We assess whether our reader raise the performance with different weights of intermediate word scores being used, which is detailed in Table. V. We notice that F1 score reaches the peak when the weight of between-word score is 0.2 for SQuAD-document and 0.1 for Trivia-wiki. This phenomenon is possibly caused by the fact that answer instances in Trivia-wiki dataset are much shorter and contain less words, thus making boundary words outperform intermediate words and increasing between-word weight will instead degrade the performance.

Effect of hybrid verifier We discuss the effect of model **I** and model **II** in hybrid verifier separately and report the result also in Table. IV. As we can see, removing model **I** results in a worse performance drop by 0.44 F1 and 0.47 F1 compared to removing model **II** by 0.36 F1 and 0.35 F1 in SQuAD-document and Trivia-wiki respectively. It indicates that semantic relationship between answers matters more than entailment relationship between answer and the input sequence. This occurred probably because the reader has mastered enough semantic information of both question and paragraph, which also validates the effectiveness of our token-level dynamic reader for generating high-quality candidate answers.

Case study We conduct a case study to demonstrate how each module takes effect with the same example discussed in §1 and compare it with BERT. For each candidate answer, we list three scores predicted by the selector, reader, and verifier in Table. VI. In specific, reader scores include scores calculated on both span-level and token-level. For the first question, top-3 ranked candidate answers all begin with “a” and end with “system”, with close selector, boundary reader, and verifier scores. It is very difficult to choose the correct answer without considering the content, but it proves that token-level reader can benefit this kind of question by concentrating on important intermediate words. Although all candidate answers have the

TABLE VI
SCORES PREDICTED BY TH-NET FOR EXAMPLES MENTIONED IN §1, WITH CORRECT ANSWERS IN BOLD.

Question & Candidate Answers	Selector	Reader		Verifier
Q1: What system did Tesla recommend to Niagara Falls in1893?	**Selector Scores**	**Boundary Scores**	**Content Scores**	**Verifier Scores**
A1: a two-phased system	0.512	5.696	**5.332**	2.866
A2: a completely integrated AC system	0.528	5.914	1.256	1.595
A3: a complete AC system	0.490	7.961	3.585	2.431
Q2: What are the main sources of primary law?	**Selector Scores**	**Boundary Scores**	**Content Scores**	**Verifier Scores**
A1: primary law, secondary law and supplementary law	0.763	5.342	9.183	3.072
A2: Treaties establishing the European Union	0.715	6.023	8.081	**4.855**
A3: the founding treaties	0.306	6.106	7.235	0.267

same boundary words, the correct one still can be chosen by determining the key intermediate word "two-phased" outperforming "complete" and "integrated". Similarly, the second candidate answer of the second question is preferred by the verifier component, thus being returned as the final answer. It proves that our hybrid verifier can be effective when the selector and reader module make an incorrect decision among the confusing answer candidates. By taking all the four scores into consideration, our model can correctly predict the answer.

V. RELATED WORK

A. Reading Comprehension Datasets

In the last few years, the SOTA performance in MRC has been rapidly advanced, in no small part because of the creation of many datasets. Earlier cloze-style task [28] requires the system to predict a held out word from a piece of text. Other datasets including SQuAD [23], [29], TriviaQA [25], and WikiReading [30] provide problems under more realistic scenario. However, none of these datasets can fulfill the multi-paragraph requirement in this paper, so we generate examples by retrieving passages for existing questions based on TF-IDF [5]. We choose to work on SQuAD and Trivia datasets considering they are more widely studied.

B. Pre-trained Language Models

More recently, many pre-trained LMs such as BERT [18] and XLNet [31] have caused a stir in NLP. LMs pre-trained on substantial unlabeled data with deep neural networks and attention-mechanism can bring deep and complex linguistic contextual representations of text, which greatly boost the performance of more than twenty language processing tasks including MRC. In this paper, we employ a unified network with three major modules also built upon pre-trained LMs, but in fine-tuned instead of feature-based way.

C. Neural Reading Comprehension

Mainstream MRC architectures including selector, reader, and verifier are realized in pipelined or unified ways [1], [12]. Since document is much longer than question compared with paragraph, we are sure to lose many useful information if we summarize each document into a fixed-sized vector and do attention between the document and question. So it is necessary to first use a selector to extract relevant paragraph content and do comprehension after that. Recent work [32] also adopts a sketchy reader first to judge whether the question is answerable and then an intensive reader to produce candidate answers based on useful paragraphs.

In reader module, studies typically consider reading comprehension task as predicting the answer boundary [5], [15], [16], [33] by leveraging various attention mechanisms to build interdependent presentations of passages and questions. However, calculating answer scores only by boundary words is clearly not sufficient to grasp enough important semantic information. [4] uses another linear network to obtain the content probability when deciding the score of a candidate answer. Our paragraph reader is also realized by combing boundary-word score and intermediate-word score. But differently, the core is a token-level dynamic gate which selects important intermediate tokens according to boundary words and reaches a balance between time, memory, and accuracy.

Besides, extracting an answer without verifying it may lead to the model be easily fooled by adversarial examples [13] and unable to recover. In response, some work handles this task in different perspectives. Read-and-verify unified network [16] has been proposed for cross-passage answer verification. Besides, candidate answers can also be verified by interacting between reader and verifier [26], adopting hierarchical answer span representations [34], bidirectionally modeling the relationship among passage, question, and candidate answers [35] and so on. Our answer verifier adopts a hybrid network combines the semantic and entailment relationship so as to focus on both local and global information that supports the answer.

VI. CONCLUSION

In this paper, we proposed TH-Net, a novel unified architecture accomplishing multi-paragraph MRC task. TH-Net includes paragraph selector, token-level dynamic reader, and hybrid verifier sharing the same context representations initialized by $\text{BERT}_{\text{BASE}}$ and fine-tuned during training. The proposed approach has the advantages of comprehending paragraphs on token-level effectively and combining semantic information between answers with entailment information between answer and the input sequence. Our method outperforms the pipelined and unified baselines on three challenging datasets: SQuAD-document, SQuAD-open, and Trivia-wiki and achieves the SOTA performance on two of them.

Acknowledgment

This research work has been funded by the National Natural Science Foundation of China (Grant No.61772337), the National Key Research and Development Program of China NO. 2018YFC0832004.

References

[1] S. Wang, M. Yu, J. Jiang, W. Zhang, X. Guo, S. Chang, Z. Wang, T. Klinger, G. Tesauro, and M. Campbell, "Evidence aggregation for answer re-ranking in open-domain question answering," *arXiv preprint arXiv:1711.05116*, 2017.

[2] M. Yan, J. Xia, C. Wu, B. Bi, Z. Zhao, J. Zhang, L. Si, R. Wang, W. Wang, and H. Chen, "A deep cascade model for multi-document reading comprehension," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7354–7361.

[3] Y. Lin, H. Ji, Z. Liu, and M. Sun, "Denoising distantly supervised open-domain question answering," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1736–1745.

[4] Y. Wang, K. Liu, J. Liu, W. He, Y. Lyu, H. Wu, S. Li, and H. Wang, "Multi-passage machine reading comprehension with cross-passage answer verification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1918–1927.

[5] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 845–855.

[6] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[7] A. Yang, Q. Wang, J. Liu, K. Liu, Y. Lyu, H. Wu, Q. She, and S. Li, "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2346–2357.

[8] Z. Zhang, Y. Wu, J. Zhou, S. Duan, and H. Zhao, "Sg-net: Syntax-guided machine reading comprehension," *arXiv preprint arXiv:1908.05147*, 2019.

[9] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, "End-to-end open-domain question answering with bertserini," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 72–77.

[10] Z. Wang, J. Liu, X. Xiao, Y. Lyu, and T. Wu, "Joint training of candidate extraction and answer selection for reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1715–1724.

[11] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, and J. Jiang, "R 3: Reinforced ranker-reader for open-domain question answering," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] M. Hu, Y. Peng, Z. Huang, and D. Li, "Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension," *arXiv preprint arXiv:1906.04618*, 2019.

[13] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2021–2031.

[14] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[15] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

[16] M. Hu, F. Wei, Y. Peng, Z. Huang, N. Yang, and D. Li, "Read+ verify: Machine reading comprehension with unanswerable questions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6529–6537.

[17] D. Hewlett, L. Jones, A. Lacoste, and I. Gur, "Accurate supervised and semi-supervised machine reading for long documents," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2011–2020.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[20] S. Min, V. Zhong, R. Socher, and C. Xiong, "Efficient and robust question answering from minimal context over documents," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1725–1735.

[21] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 188–197.

[22] L. He, K. Lee, O. Levy, and L. Zettlemoyer, "Jointly predicting predicates and arguments in neural semantic role labeling," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 364–369.

[23] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.

[24] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1870–1879.

[25] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1601–1611.

[26] R. Das, S. Dhuliawala, M. Zaheer, and A. McCallum, "Multi-step retriever-reader interaction for scalable open-domain question answering," *arXiv preprint arXiv:1905.05733*, 2019.

[27] Z. Chen, R. Yang, B. Cao, Z. Zhao, D. Cai, and X. He, "Smarnet: Teaching machines to read and comprehend like human," *arXiv preprint arXiv:1710.02772*, 2017.

[28] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in neural information processing systems*, 2015, pp. 1693–1701.

[29] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 784–789.

[30] D. Hewlett, A. Lacoste, L. Jones, I. Polosukhin, A. Fandrianto, J. Han, M. Kelcey, and D. Berthelot, "Wikireading: A novel large-scale language understanding task over wikipedia," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1535–1545.

[31] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, 2019, pp. 5754–5764.

[32] Z. Zhang, J. Yang, and H. Zhao, "Retrospective reader for machine reading comprehension," *arXiv preprint arXiv:2001.09694*, 2020.

[33] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, "Reinforced mnemonic reader for machine reading comprehension," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 4099–4106.

[34] L. Pang, Y. Lan, J. Guo, J. Xu, L. Su, and X. Cheng, "Has-qa: Hierarchical answer spans model for open-domain question answering," *arXiv preprint arXiv:1901.03866*, 2019.

[35] S. Zhang, H. Zhao, Y. Wu, Z. Zhang, X. Zhou, and X. Zhou, "Dual co-matching network for multi-choice reading comprehension," *arXiv preprint arXiv:1901.09381*, 2019.