Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

# GSA-Net: gated scaled dot-product attention based neural network for reading comprehension

Xiang Ma & Junsheng Zhang

**To cite this article:** Xiang Ma & Junsheng Zhang (2020) GSA-Net: gated scaled dot-product attention based neural network for reading comprehension, Automatika, 61:4, 643-650, DOI: 10.1080/00051144.2020.1809221

**To link to this article:** https://doi.org/10.1080/00051144.2020.1809221

Published online: 01 Sep 2020.

AUTOMATIKA
2020, VOL. 61, NO. 4, 643–650
https://doi.org/10.1080/00051144.2020.1809221

# GSA-Net: gated scaled dot-product attention based neural network for reading comprehension

Xiang Ma[a] and Junsheng Zhang [b]

[a]Institute of Tourism, Changchun Vocational Institute of Technology, Changchun, People's Republic of China; [b]Research Center for Information Science Theory and Methodology, Institute of Scientific and Technical Information of China, Beijing, People's Republic of China

**ABSTRACT**
Reading Comprehension (RC) is concerned with building systems that automatically answer questions about a given context passage. The interactions between the context and question are very important to locate the correct answer. In this paper, we propose a Gated Scaled Dot-Product Attention based model for RC task. The character-level embedding is incorporated into the word embedding which is helpful to deal with Out-of-Vocabulary (OOV) tokens. The attention distribution is obtained by scaled dot product which captures the interaction between question and passage effectively. Further, self-matching attention mechanism is adopted to resolve the problem of long-distance dependency. These components provides more information for the prediction of the starting and ending position of the answer. We evaluate our method on Stanford Question Answering Dataset (SQuAD) and the results show that different components in the model boost the performance.

**ARTICLE HISTORY**
Received 18 June 2020
Accepted 10 August 2020

**KEYWORDS**
Natural language processing; attention; machine reading comprehension

## 1. Introduction

Question Answering (QA) is the task of retrieving answer to a given question. It is an intelligent search engine based on natural language processing and information retrieval techniques. The user is allowed to ask questions in natural language and the system will return the corresponding answer directly. It has been explored both in the open-domain field [1] and domain-specific settings, such as BioASQ for Biomedical field [2]. The Reading Comprehension task limits the candidate answer in a given passage.

Question Answering techniques have been improved by the promotion of official evaluation and open data set publication. Wang [3] proposes the generative probability model to compute the matching degree of dependency tree between question and answer. Heilman and Smith [4] make use of the conditional random field model to estimate the structural distance between question and answer in dependency tree. Ko [5] puts forward a probability-based ranking model to select the candidate answers, and logistic regression method is used to estimate the probability of the correct answer. Severyn and Moschitti [6] use the SVM tree kernel to learn the shallow syntactic features for classification of question-answer pairs.

The traditional methods have also been applied on biomedical datasets and achieve better results. The OAQA(Open Advancement of Question Answering) system [7] combines the biomedical resources, including domain-specific parsers and entity markers, to retrieve concepts and synonyms. Logistic regression classifiers are used for question classification and candidate answer scoring.

With the publication of large-scale open-domain datasets Stanford Question Answering Dataset (SQuAD) [8, 9], TriviaQA [10], WikiReading [11], Children Book Test [12], etc, the neural network based techniques for QA system have been developed recently [13–16], and they lead to the significant improvement over traditional methods.

The neural network based Question Answering makes difference with the traditional methods. Usually, the neural model is trained by end-to-end way for achieving an answer to the given question and passage. Yu [17] uses Convolutional Neural Networks (CNN) to model the distribution of question and answer. Feng [18] publishes a question-answer data set for the insurance domain and proposes several CNN models based on this data set. Because of the superior performance of the Attention mechanism [19] in the sequence to sequence model, the researchers try to introduce the attention mechanism into the question answering task. With the publication of SQuAD dataset [8, 9], the techniques based on deep learning have been well verified. Wang and Jiang [9] propose two kinds of answer prediction models: Sequence Model and the Boundary Model, and the interaction layer is imported to compute the attention distribution. Seo [15] improves the model of Wang and Jiang [13] by adding the bidirectional attention mechanism. Though these approaches work well

**CONTACT** Junsheng Zhang zhangjs@istic.ac.cn

in Question Answering task, in all but a few cases, such methods have a high requirement for computing resources and have difficulties to answer complex context-dependent questions.

Most question answering systems based on neural networks use interactive attention mechanism or bi-attention mechanism to obtain answers [20]. The existing methods mainly focus on the relationship between the question and the passage, and they pay little attention to the interactive verification between candidate answers. The problem is more obvious in open-domain QA, where a question needs to be answered by considering candidate answers from multiple paragraph. In order to resolve this problem, Wang et al. [21] propose a two-stage process extraction: first extract answer candidates from passages and then select the final answer by combining information from all of the candidates. V-net [22] adopts an end-to-end neural model that enables answer candidates from different passages to verify each other based on their content representations.

To obtain answers to complex questions, reviewing after reading documents for further reasoning is necessary. This can be realized by multi-round reasoning mechanism, which attempts to combine the information of questions with the new information extracted from previous iterations ([23–25]). Gated-attention reader [26] uses multiplicative interactions between the query embedding and intermediate states of a recurrent neural network reader, which is realized by feeding the question encoding into an attention-based gate in each iteration. Cui et al. [27] further propose that question specific attention should be extended to bi-attention mechanism, including both question to document and document to question. ReasonNet [28], which is different from those methods which use fixed iterations, adds a termination module to recognize whether to go on to the next inference or to terminate the reasoning process when the information is sufficient.

To alleviate the problems above, our model allows for significantly more parallelization and even gets higher accuracy for answer prediction. We propose a Gated Scaled Dot-Product Attention based model for Reading Comprehension task, which aims to answer questions in a given context passage. The character-level embedding is incorporated to word embedding which is helpful to deal with Out-of-Vocabulary (OOV) tokens. The attention distribution is achieved by Scaled dot product and self-matching attention mechanism. Finally, the Pointer Network is used to predict the starting and ending position of the answer. The rest of paper is organized as following. Section 2 introduces the hierarchical multi-layer mechanism of the proposed model. Section 3 introduces the different level encoding for question and passage. Section 4 describes the attention-based passage encoding by question interaction. Section 5 is the Pointer Network for answer selection. Section 6 gives the experimental results and Section 7 is the conclusion.

## 2. Hierarchical multi-layer reading comprehension model

We put forward a Gated Scaled Dot-Product Attention based model for RC task, which is represented as a hierarchical multi-layer mechanism shown in Figure 1. It consists of six components.

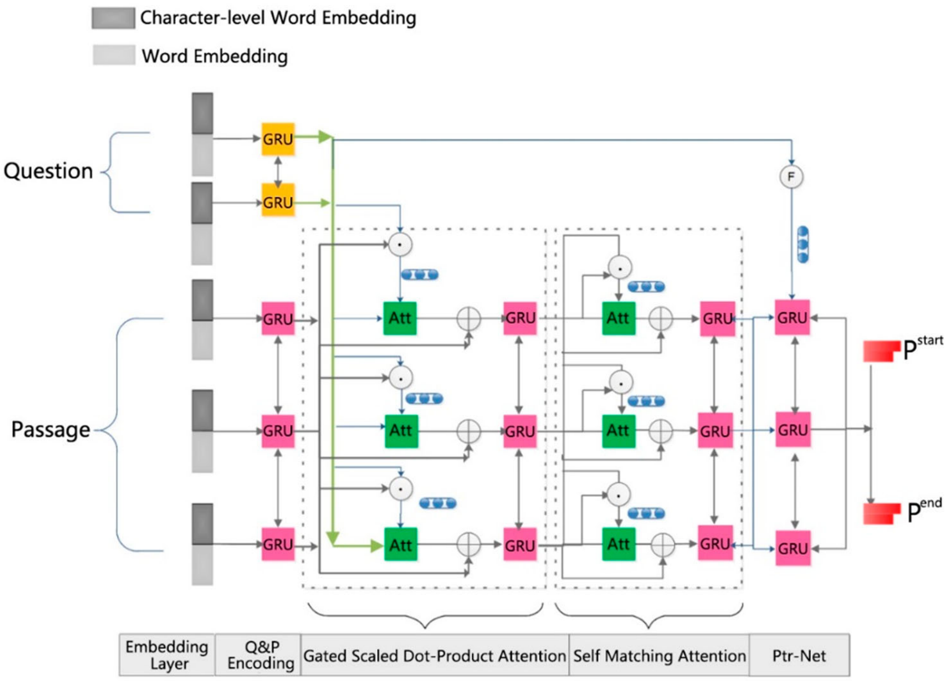

**Figure 1.** Gated Scaled Dot-Product Attention based Module Structure for RC task.

**Character-Level Word Embedding Layer**. Each Character is mapped into a high-dimensional vector space and the character-level word embedding for each word is generated by Bi-GRU.

**Word Embedding Layer.** The word-level vector is concatenated with character-level vector. The distributed matrix representation of each word in question and passage is generated by the two-layer Highway Network.

**Question and Passage Encoding Layer.** It utilizes contextual cues from surrounding words to refine the embedding of each word in question and passage.

**Gated Scaled Dot-Product Attention Layer.** The representation of each word in the candidate passage is encoded by the conjunction of question-aware feature vector.

**Self-matching Attention Layer.** The representation of passage is enriched by matching itself with the output representation of the previous layer. It captures the important cues with long-distance.

**Pointer Network for Answer Selection.** For each question, predict the starting position and ending position of the answer in the passage by Pointer Network.

## 3. Question and passage encoding

### 3.1. Character-level embedding layer

The character-level embedding layer is responsible for mapping each word into a high-dimensional vector space, which has been shown to be helpful to deal with Out-Of-Vocabulary (OOV) tokens.

Question and contextual passage are represented as the word set $Q = \{w_1^q, w_2^q, \ldots, w_m^q\}$ and P $= \{w_1^p, w_2^p, \ldots, w_n^p\}$ respectively. Each word $w_i$ consists of several characters and it is represented as the character-level word distribution matrix $w_i = \{c_1, c_2, \ldots, c_k\}$. The distributed representation of each character $c_i (i = 1, \ldots k)$ is obtained by the pre-trained character vector.[1] Further, Bi-directional Gated Recurrent Unit (Bi-GRU) is used to generate character-level word embedding for each word in question and passage separately.

$$u_i^q = BiGRU(u_{i-1}^q, c_i^q) \quad (i = 1, \ldots k) \tag{1}$$

$$u_i^p = BiGRU(u_{i-1}^p, c_i^p) \quad (i = 1, \ldots k) \tag{2}$$

We use the final hidden state of Bi-GRU $u_k^p$ and $u_k^q$ to represent character-level word embedding which is shown in Figure 2.

### 3.2. Word embedding layer

Word Embedding Layer also maps each word to high-dimensional vector space. We use the pre-trained word vector model Glove [29] to obtain fixed-dimension word vectors $\{e_t^q\}_{t=1}^m$ for question and $\{e_t^p\}_{t=1}^n$ for passage.

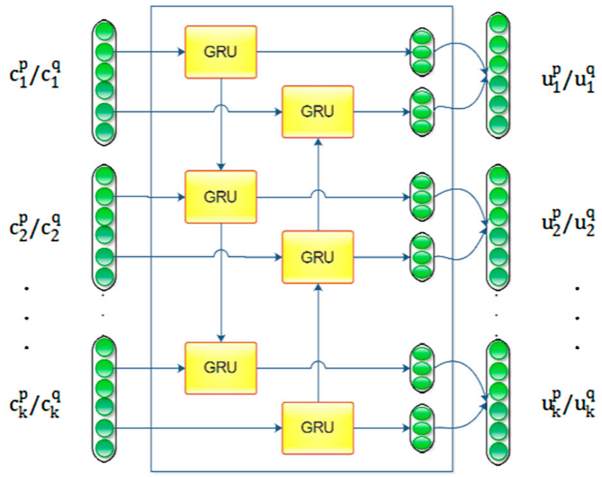

**Figure 2.** Character-level Word Embedding by Bi-GRU.

For each word $w_t$ in question and passage, the character-level vector $u_t^q$, $u_t^p$ and word-level vector $e_t^q$, $e_t^p$ are concatenated respectively, which is represented as Equation 3–4.

$$\mathrm{Q} = \{[e_t^q, u_t^q]\}_{t=1}^m \tag{3}$$

$$\mathrm{P} = \{[e_t^p, u_t^p]\}_{t=1}^n \tag{4}$$

Further, the distributed matrix representation of each word $w_t$ in question and passage is generated by the two-layer Highway Network [30], which are shown in Equation 5–6.

$$\begin{aligned} y_t^q &= ReLU(x_t^q W^x + b^x) \cdot \sigma(x_t^q W^T + b^T) \\ &\quad + x_t^q(1 - \sigma(x_t^q W^T + b^T)) \end{aligned} \tag{5}$$

$$\begin{aligned} y_t^p &= ReLU(x_t^p W^x + b^x) \cdot \sigma(x_t^p W^T + b^T) \\ &\quad + x_t^p(1 - \sigma(x_t^p W^T + b^T)) \end{aligned} \tag{6}$$

Where $x_t^q = [e_t^q, u_t^q] \in R^d$ and $x_t^p = [e_t^p, u_t^p] \in R^d$, $d$ represents the dimension of the concatenated vector. Finally, the question and passage are represented as matrix $\mathrm{Q} = \{y_t^q\}_{t=1}^{m} \in R^{m*d}$ and $\mathrm{P} = \{y_t^p\}_{t=1}^{n} \in R^{n*d}$ respectively.

### 3.3. Contextual encoding for question and passage

Based on the output of previous layer, question representation matrix $\mathrm{Q} = \{y_t^q\}_{t=1}^{m}$ and passage representation matrix $\mathrm{P} = \{y_t^p\}_{t=1}^{n}$, Bi-directional GRU is used to model the temporal interaction between words and they are denoted as Equation 7–8.

$$u_t^{\prime q} = BiGRU(u_{t-1}^{\prime q}, y_t^q) \tag{7}$$

$$u_t^{\prime p} = BiGRU(u_{t-1}^{\prime p}, y_t^p) \tag{8}$$

The distributed representations of the word $w_t$ in question and passage is denoted as $u_t^{\prime q}$ and $u_t^{\prime p}$ separately, and it considers the representation of previous *t-1* words iteratively. The representations of question and passage are defined as $U^q = \{u_1'^q, u_2'^q, \dots u_m'^q\}$ and $U^p = \{u_1'^p, u_2'^p, \dots u_n'^p\}$ respectively.

## 4. Attention-based passage encoding

### 4.1. Gated scaled dot-product attention layer

An attention mechanism called Gated Attention-based Recurrent Network is proposed to generate new passage representation aligned to question [31]. We adopt the scaled dot-product to implement question-aware passage encoding. Dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

An attention function maps a passage and a set of key-value pairs of question to an output, where the passage(P), keys($Q_{key}$), values($Q_{value}$) and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the passage with the corresponding key of the question. The question-aware passage embedding is achieved by this particular attention mechanism-Scaled Dot-Product Attention [32], which is shown in Figure 3.

The input consists of passage and question's key-value pairs with dimension $d_q$. We compute the dot products of the passage with all keys of question, divide each by $\sqrt{d_q}$, and apply a softmax function to obtain the weights on the values.

The attention function on a set of words in passage is computed simultaneously, packed together into a matrix ***P***. The keys and values are also packed together into matrices $\boldsymbol{Q_{key}}$ and $\boldsymbol{Q_{value}}$ . Given question and passage representation $U^q = \{u_1'^q, u_2'^q, \dots u_m'^q\}$ and $U^p = \{u_1'^p, u_2'^p, \dots u_n'^p\}$, the output attention matrix is computed as:

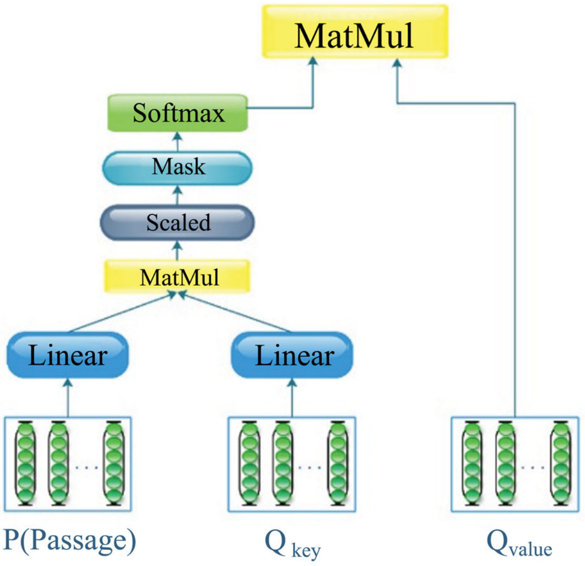

**Figure 3.** Scaled Dot-Product Attention on Question and Passage.

$$C(P, Q_{key}, Q_{value}) = softmax\left(\frac{P \cdot Q_{key}^T}{\sqrt{d_q}}\right) Q_{value} \qquad (9)$$

Here, $P = \sigma(U^p) \in R^{n*d}, Q_{key} = \sigma(U^q) \in R^{m*d}, Q_{value} = U^q \in R^{m*d}$ and $\sigma = ReLU(Wx + b)$ is a non-linear mapping function.

The purpose of attention in RC system is to read the passage by incorporation of question and re-encode the passage by the question-aware information.

To make the attention focused on the important parts of passage that is relevant to question, we add another gate to the input of bi-GRU and it is updated as Equation 10–12. And then the representation of passage is updated as $U^p = \{u_1''^p, u_2''^p, \dots u_n''^p\}$.

$$u_t''^p = biGRU(u_{t-1}''^p, [u_t'^p, c_t]^*) \qquad (10)$$

$$[u_t'^p, c_t]^* = g_t \odot [u_t'^p, c_t] \qquad (11)$$

$$g_t = sigmoid(W_g[u_t'^p, c_t]) \qquad (12)$$

Here, $u_t'^p$ is from the previous encoding layer and it is an additional input into the recurrent network. $c_t$ is the $t$th vector of attention matrix $C$.

### 4.2. Self-matching attention layer on passage

The question-aware passage encoding is generated by Gated Scaled Dot-Product Attention Layer. There still exists a problem that it has limited understanding of the context and misses the important cues outside its surrounding window. Passage context is necessary to infer the correct answer. To address this problem, we use the question-aware distributed representation directly to match the passage itself. It dynamically collects the evidence from the whole paragraph and encodes the evidence relevant to the current passage word. The encoding result $u_t'''^p$ for word $w_t$ in passage is got by Equation 13.

$$u_t'''^p = biGRU(u_{t-1}'''^p, [u_t''^p, C(P, P_{key}, P_{value})_t]) \qquad (13)$$

Here, $P = P_{key} = \sigma(U^p)$ and $P_{value} = U^p$ in a self-attention layer are obtained as the output of the previous layer in the encoder. Each position in the encoder can attend to all positions in the previous layer.

## 5. Pointer network for answer selection

We use pointer networks [33] to predict the starting and ending position of the answer in the passage. The attention-pooling over the question representation is used to generate the initial hidden vector for the pointer network. Given the passage representation $\{u_t'''^p\}_{t=1}^n$, the attention mechanism is utilized to select the starting position $p^{Start}$ and ending position $p^{End}$ in the passage, which can be formulated as following.

For each word $w_t$ in passage, predict the corresponding probability to be the starting ($L = Start$) or ending ($L = End$) word of the answer.

$$s_t^L = v^T tanh(W^p u_t^{\prime\prime\prime p} + Wh_L) \quad (\mathrm{L = Start, End})$$

$$a_t^L = \frac{exp(s_t^L)}{\sum_{i=1}^n exp(s_i^L)}$$

$$p^L = argmax(a_1^L, \ldots, a_n^L) \tag{14}$$

We use question representation $U^q = \{u_1^{\prime q}, u_2^{\prime q}, \ldots u_m^{\prime q}\}$, which is obtained from question encoding layer described in section 3.3, to help locate the starting position of answer $p^{Start}$. Here, $h_{Start}$ in Equation 15 is computed as Equation 16 and it is an attention-pooling vector of the question based on the random parameter $v^q$.

$$s_t^q = v^T tanh(W^q u_t^{\prime q} + Wv^q)$$

$$a_t^q = \frac{exp(s_t^q)}{\sum_{j=1}^m exp(s_j^q)}$$

$$h_{Start} = \sum_{i=1}^m a_i^q u_t^{\prime q} \tag{15}$$

Also the attention-based passage representation $\{u_t^{\prime\prime\prime p}\}_{t=1}^n$ is used to predict the ending position of answer by Equation 16. $h_{End}$ represents the last hidden state of the pointer network.

$$h_{End} = biGRU\left(h_{Start}, \sum_{t=1}^n a_t^{Start} u_t^{\prime\prime\prime p}\right) \tag{16}$$

## 6. Experiment and results

### 6.1. Dataset

We evaluate our model on Stanford Question Answering Dataset (SQuAD) V1.1[2] dataset. It consists of questions on a set of Wikipedia articles, where the answer to every question is a segment of text from the corresponding reading passage. With 100,000+ question-answer pairs on 500+ articles, SQuAD is significantly larger than previous reading comprehension datasets (Table 1).

The question-context sample from SQuAD is shown in below.

***Passage:*** *In 1870, Tesla moved to Karlovac, to attend school at the Higher Real Gymnasium, where he was profoundly influenced by a math teacher Martin Sekulic. The classes...*

***Question:*** *Who was tesla's main influence in Karlovac?*

***Answer:*** *Martin Sekulic*

**Table 1.** SQuAD Data Distribution.

| | | Questions-Answer pairs | | |
|---|---|---|---|---|
| #Passage | #Question | #Traing | #Dev | #Test |
| 23,251 | 107,785 | 87,599 | 10,570 | 9,616 |

### 6.2. Experimental setting and evaluation

We use the StanfordNLP tokenizer [34] to preprocess each passage and question. For word embedding, we use pre-trained case-sensitive GloVe embeddings[3] [29] for both questions and passages, and it is fine-tuning during training. All of the out-of-vocab words are represented by zero vectors. The length of hidden vector is set to 128 for all layers. The hidden size used to compute attention score is also set to 128. We apply dropout between layers with a dropout rate 0.5. The model is optimized with Adam with an initial learning rate of 0.5.

BioASQ (http://participants-area.bioasq.org/) is a challenge providing training data for biomedical semantic indexing and question answering task. The 2017 BioASQ training dataset contains 1799 questions, of which there are 413 factoid questions and 486 list questions. These questions have about 20 snippets on average and each of them have 34 tokens long.

Two metrics are utilized to evaluate the model performance on SQuAD and they are Exact Match (EM) and F1 score. EM measures the percentage of the predictions that match any one of the ground truth answers exactly. F1 measures the word overlap between the prediction and ground truth answer.

$$\mathrm{EM} = \frac{\sum_{i=1}^M I(a'_i = a_i^*)}{M} \tag{17}$$

$$\mathrm{Recall} = \sum_{i=1}^M \frac{count(a'_i \cap a_i^*)}{count(a_i^*)} \tag{18}$$

$$\mathrm{Precision} = \sum_{i=1}^M \frac{count(a'_i \cap a_i^*)}{count(a'_i)} \tag{19}$$

$$\mathrm{F1} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{20}$$

Here, $M$ denotes the number of test samples. The predicted answer is represented as $a_i'$ and ground-truth answer is denoted as $a_i^*$.

### 6.3. Experimental results

In order to evaluate the performance impact by different components in our hierarchical multi-layer model, we give detailed comparison by removing them separately and the results are shown in Table 2. The scores on DevSet are evaluated by the official script.[4]

As it can be seen from Table 2, the performance declines by removing different components in GSA-Net. These components play an important role to select

**Table 2.** Performance impact by different components in GSA-Net.

| | Evaluation | Results |
|---|---|---|
| Model | EM | F1 |
| GSA-Net | 71.1 | 80.1 |
| GSA-Net Without Character-Level Embedding | 69.5(−2.25%) | 78.7(−1.75%) |
| GSA-Net Without Gate | 68.1(−4.21%) | 77.5(−3.25%) |
| GSA-Net Without Self-Matching Layer | 67.6(−4.92%) | 76.8(−4.12%) |
| GSA-Net Without Gate and Self-Matching Layer | 65.2(−8.29%) | 74.8(−6.62%) |

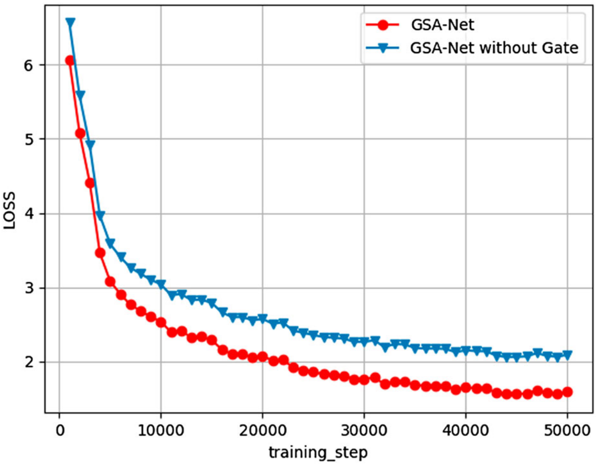

**Figure 5.** The Loss during the Training Process for Answer Selection.

**Table 3.** The performance comparison with different methods.

| | Evaluation | Results |
|---|---|---|
| Models | EM | F1 |
| LR Baseline [8] | 40.0 | 51.0 |
| Dynamic Chunk Reader [35] | 62.5 | 71.2 |
| Match-LSTM with Ans-Ptr [13] | 64.1 | 73.9 |
| Dynamic Co-attention Network [14] | 65.4 | 75.6 |
| RaSoR [36] | 66.4 | 74.9 |
| BiDAF [15] | 68.0 | 77.3 |
| Fine-Grained Gating(ensemble) [37] | 62.4 | 73.0 |
| GSA-Net(Ours) | 71.1 | 80.1 |

the correct answer. Character-level word embedding can handle the Out-of-Vocabulary tokens very well. With the Self-Matching Layer removed, EM and F1 value is lowered by 4.92% and 4.12% respectively. It indicates that the long-distance dependency in the passage can help to locate the correct answer efficiently. In addition, Gate mechanism makes the attention focused on the important parts of passage and boosts the performance.

In order to show the ability of the model for encoding evidence from passage, we draw the alignment of the passage against the question in Gated Scaled Dot-Product Attention Layer. The attention weights are visualized and shown in Figure 4. The darker of the colour, the higher weight value of the word. For example, the answer “Martin Sekulic” in the passage is given more attention to the question. Other words with darker colour, such as “was”, “Kalovac” and “Tesla”, are overlapped with question.

The cross-entropy loss for the RC task in models GSA-Net and GSA-Net without Gate are demonstrated in Figure 5. The loss value decreased gradually with the increasing of training step. It is shown that GSA-Net model has the less loss and also converges faster.

We also compare the performance of our model GSA-Net with other related work on SQuAD as shown in Table 3.

**LR Baseline** [8]**:** A model based on Logistic Regression for RC task, which extracts several types of features for each candidate and computes the unigram/bigram overlap between the sentence and question.

**Dynamic Chunk Reader** [35]**:** A novel neural network model for joint candidate answer chunking and ranking, where the candidate answer chunks are dynamically constructed and ranked in an end-to-end manner.

**Match-LSTM with Ans-Ptr** [13]**:** It proposes two new end-to-end neural network models for machine comprehension task, which combine match-LSTM and Ptr-Net to handle the special properties of the SQuAD dataset.

**Dynamic Co-attention Network** [14]**:** A Model called Dynamic Co-attention Network (DCN) for question answering. The DCN firstly fuses co-dependent representations of the question and the document in order to focus on relevant parts of both. Then a dynamic pointing decoder iterates over potential answer spans.

**RaSoR** [36]**:** A model called RASOR that efficiently builds fixed length representations of all spans in the evidence document with a recurrent network. It explicitly computes embedding representations for candidate answer spans.

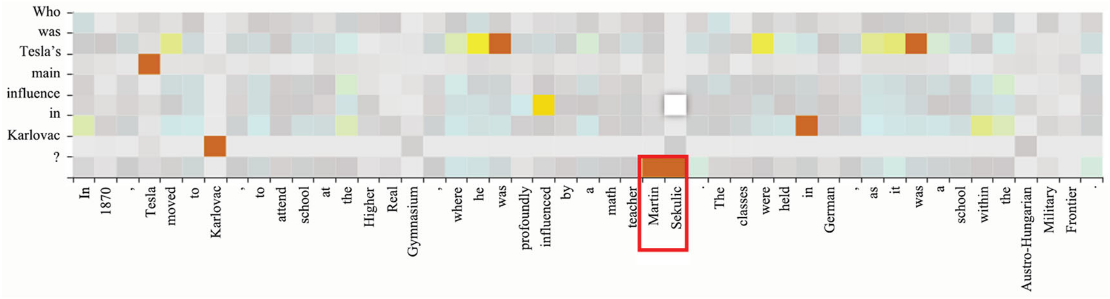

**Figure 4.** Visualization of the Attention Weight in Gated Scaled Dot-product Layer.

**Table 4.** Comparison with competing BioASQ systems.

| | | Top 1 | Top 2 | GSA-Net(Ours) |
|---|---|---|---|---|
| Batch 1 | F1 | 0.33 | 0.34 | 0.34 |
| Batch 2 | F1 | 0.50 | 0.26 | 0.34 |
| Batch 3 | F1 | 0.41 | 0.49 | 0.49 |
| Batch 4 | F1 | 0.29 | 0.38 | 0.30 |
| Batch 5 | F1 | 0.38 | 0.42 | 0.35 |
| Avg | F1 | 0.38 | 0.38 | 0.37 |

**BiDAF** [15]**:** A Model named Bi-Directional Attention Flow (BIDAF) network, a multi-stage hierarchical process that represents the context at different levels of granularity and uses bidirectional attention flow mechanism to obtain a query-aware context representation without early summarization.

**Fine-Grained Gating (ensemble)** [37]**:** A model with fine-grained gating mechanism to combine word-level and character-level representations dynamically. It further extends the idea of fine-grained gating to model the interaction between question and paragraph for reading comprehension.

The results in Table 3 denote that our model GSA-Net performs best in both EM and F1 value. Our method obviously outperforms the baseline and several strong state-of-the-art systems for both single model and ensemble one.

We also compare the performance of our model with the participating systems in BioASQ.[5] For each batch and different question type, the result of the top 2 competing systems and our model are shown in Table 4.

Our model has a strong ability to process the interactive encoding of questions and contexts, which can get high quality question-context alignment representation. Gated mechanism is adopted in our model to solve the problem of propagating dependencies over long distances. Moreover, hierarchical attention mechanisms can locate the segments which are related to the answer step by step, therefore the semantic representation ability of the model is enhanced.

## 7. Conclusion

Machine Reading Comprehension is an important task for natural language understanding. It evaluates the machine's ability to access knowledge and answer questions from the given passage. This paper proposes an end-to-end machine reading comprehension framework, which can well understand questions and relevant fragments for answer prediction. We present a Gated Scaled dot-product Attention based Neural network (GSA-Net) for Reading Comprehension task. The different components in this hierarchical multi-layer model play an important role to locate the correct answer. The gated scaled dot-product attention and self-matching attention mechanism are used to obtain the suitable question-aware representation for passage. Further, the pointer network predicts the answer position effectively. Our model achieves an exact match score (EM) of 71.1% and an F1 score of 80.1% on SQuAD, which outperforms several strong competing systems. This model has little memory requirements, and it performs even better than the models that rely on more computing resources.

Although we have added two attention layers in the proposed model, the interpretability is still not enough, which is a common problem in the deep learning based natural language processing task. In addition, the cross-paragraph reasoning ability of this model needs to be improved and it is important to answer complex question. In the future work, we will try to combine BERT with our method to improve the reasoning ability of the model. Meanwhile, how to learn the prior knowledge of human language expression from large-scale unstructured data and apply it to machine reading comprehension is also a significant goal of our work.

## Notes

1. https://github.com/minimaxir/char-embeddings
2. http://stanford-qa.com/
3. http://nlp.stanford.edu/data/glove.840B.300d.zip
4. http://stanford-qa.com/
5. http://participants-area.bioasq.org/results/5b/phaseB/

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This research work was partially supported by National Natural Science Foundation of China (grant number 71503240) and the National Key Research and Development Program of China (grant number 2019YFA0707201).

## ORCID

*Junsheng Zhang* http://orcid.org/0000-0001-8740-2851

## References

[1] Voorhees EM. Overview of the trec8 question answering track report. Text Retrieval Conf. 2000;99:77–82.

[2] Tsatsaronis G, Balikas G, Malakasiotis P, et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics. 2015;16(1):1–28.

[3] Wang M, Smith NA, Mitamura T. What is the jeopardy model? A quasi-synchronous grammar for qa. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; Prague; 2007. p. 22–32.

[4] Heilman M, Smith NA. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL; Los Angeles, California; 2010. p. 1011–1019.

[5] Ko J, Si L, Nyberg E. A probabilistic framework for answer selection in question answering. Proceedings of

North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Rochester, New York; 2007. p. 524–531.

[6] Severyn A, Moschitti A. Automatic feature engineering for answer selection and extraction. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing; Washington, USA; 2013. p. 458–467.

[7] Zi Y, Yue Z, Nyberg E. Learning to answer biomedical questions: OAQA at BioASQ 4B. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics; 2016. p. 23–37.

[8] Rajpurkar P, Zhang J, Lopyrev K, et al. SQuAD: 100,000+ questions for machine comprehension of text. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; Austin, Texas; 2016. p. 2383–2392.

[9] Rajpurkar P, Robin J, Percy L. Know what you don't know: unanswerable questions for SQuAD. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; Melbourne, Australia; 2018. p. 784–789.

[10] Joshi M, Choi E, Weld DS, et al. TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics; Vancouver, Canada; 2017. p. 1601–1611.

[11] Hewlett D, Lacoste A, Jones L, et al. Wikireading: a novel large-scale language understanding task over Wikipedia. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics; Berlin, Germany; 2016; p. 1535–1545.

[12] Hill F, Bordes A, Chopra S, et al. The goldilocks principle: reading children's books with explicit memory representations. Proceedings of the International Conference on Learning Representations; Puerto Rico; 2016.

[13] Wang S, Jiang J. Machine comprehension using match-lstm and answer pointer. Proceedings of International Conference on Learning Representations; 2017.

[14] Xiong C, Zhong V, Socher R. Dynamic co-attention networks for question answering. Proceedings of International Conference on Learning Representations; Toulon, France; 2017.

[15] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension. Proceedings of International Conference on Learning Representations. abs/1611.01603; Puerto Rico; 2017.

[16] Weissenborn D, Wiese G, Seiffe L. Making neural qa as simple as possible but not simpler. Proceedings of the 21st Conference on Computational Natural Language Learning; 2017. p. 271–280.

[17] Yu L, Hermann KM, Blunsom P, et al. Deep learning for answer sentence selection. arXiv:1412.1632 [cs]. 2014;Dec. 2014:arXiv: 1412.1632.

[18] Feng M, Xiang B, Glass MR, et al. Applying deep learning to answer selection: a study and an open task. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU); Scottsdale, AZ; 2015. p. 813–820.

[19] Bahdanau D, Kyunghyun C, Yoshua B. Neural machine translation by jointly learning to align and translate. Proceedings of International Conference on Learning Representations; 2014.

[20] Xiong C, Zhong V, Socher R. Dcn+: mixed objective and deep residual coattention for question answering. Proceedings of the 6th International Conference on Learning Representations, abs/1711.00106; Vancouver, Canada; 2018.

[21] Wang Z, Liu J, Xiao X, et al. Joint training of candidate extraction and answer selection for reading comprehension. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; Melbourne, Australia; 2018. p. 1715–1724.

[22] Wang Y, Liu K, Liu J, et al. Multi-passage machine reading comprehension with cross-passage answer verification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; Melbourne, Australia; 2018. p. 1918–1927.

[23] Sordoni A, Bachman P, Bengio Y. Iterative alternating neural attention for machine reading. Comput Res Repo (CoRR), abs/1606.02245. 2016.

[24] Weissenborn D. Separating answers from queries for neural reading comprehension. Comput Res Repo, abs/1607.03316. 2016.

[25] Kumar A, Irsoy O, Ondruska P, et al. Ask me anything: dynamic memory networks for natural language processing. Proceedings of the International Conference on Machine Learning (ICML); New York, USA; 2016. p. 1378–1387.

[26] Dhingra B, Liu H, Cohen WW, et al. Gated-attention readers for text comprehension. Comput Res Repo, abs/1606.01549. 2016.

[27] Cui Y, Chen Z, Wei S, et al. Attention-over-attention neural networks for reading comprehension. Comput Res Repo (CoRR), abs/1607.04423. 2016.

[28] Shen Y, Huang P-S, Gao J, et al. Reasonet: learning to stop reading in machine comprehension. Proceedings of the 5th International Conference on Learning Representations, abs/1609.05284; Toulon, France; 2017.

[29] Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing; Doha, Qatar; 2014. p. 1532–1543.

[30] Srivastava RK, Greff K, Schmidhuber J. Highway networks. arXiv:1505.00387. 2015.

[31] Wang W, Yang N, Wei F, et al. Gated self-matching networks for reading comprehension and question answering. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics; Vancouver, Canada; 2017. p. 189–198.

[32] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. The Annual Conference on Neural Information Processing Systems; California, USA; 2017. p. 6000–6010.

[33] Vinyals O, Fortunato M, Jaitly N. Pointer networks. Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015; Montreal, Quebec, Canada. 2015. p. 2692–2700.

[34] Manning CD, Surdeanu M, Bauer J, et al. The Stanford corenlp natural language processing toolkit. Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (System Demonstrations); 2014. p. 55–60.

[35] Yu Y, Zhang W, Hasan K, et al. End-to-end reading comprehension with dynamic answer chunk ranking. arXiv preprint arXiv:1610.09996. 2016.

[36] Lee K, Kwiatkowski T, Parikh A, et al. Learning recurrent span representations for extractive question answering. arXiv preprint arXiv:1611.01436. 2016.

[37] Yang Z, Dhingra B, Yuan Y, et al. Words or characters? fine-grained gating for reading comprehension. CoRR abs/1611.01724. 2016.