Received March 28, 2019, accepted June 16, 2019, date of publication August 13, 2019, date of current version October 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930407

Enhancing Machine Reading Comprehension With Position Information

YAJING XU[1], WEIJIE LIU[1], GUANG CHEN[1], BOYA REN[2], SIMAN ZHANG[3], SHENG GAO[1], AND JUN GUO[1]

[1]Pattern Recognition and Intelligent System Laboratory, Beijing University of Posts and Telecommunications, Beijing 100000, China
[2]The National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100000, China
[3]DOCOMO Beijing Communications Laboratories Company Ltd., Beijing 100000, China

Corresponding author: Weijie Liu (liuweijie@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61300080, and in part by the DOCOMO Beijing Communications Laboratories Company Ltd.

ABSTRACT When people do the reading comprehension, they often try to find the words from the passages which are similar to the question words first. Then people deduce the answer based on the context around these similar words. Therefore, the position information may be helpful in finding the answer rapidly and is useful for reading comprehension. However, previous attention-based machine reading comprehension models typically focus on the interaction between the question and the context representation without considering the position information. In this paper, we introduce the position information to machine reading comprehension and investigate the performance of the position information. The position information is experimented in three different ways: 1) position encoder; 2) attention mechanism; and 3) position mapping embedding. By experimenting on TriviaQA dataset, we have demonstrated the effectiveness of position information.

INDEX TERMS Attention mechanism, machine comprehension, position information.

I. INTRODUCTION

As a task of automatic question answering, machine reading comprehension is usually defined as a task to answer a corresponding question based on the some natural language documents. If a machine reading comprehension model can obtain a good score from predicting the right answer, we believe that the model is capable of understanding the given context.

Position information is important in many tasks of natural language processing (NLP), such as machine translation, entity relation extraction and question answering. While doing human reading comprehension, people usually locate the section which is close to question word first. Table 1 is an example of machine reading comprehension. There are about thousands of words in the whole document, and through the relevant search of question words, we can first locate a excerpt of corresponding document shown in Table 1. In this excerpt, the answer "Cliff Thorburn" is also found surrounded the question words "snooker player" and "The Grinder" ("nickname" is also a key word if we associate it with question word "known as"). Thus we can find that question answering task usually requires only a small portion of the context which is close to question words.

The research of machine reading comprehension [2]–[4] has achieved a rapid progress with the release of large-scale datasets like SQuAD [5], TriviaQA [6], CNN/Daily Mail [7] and Children's Book Test [8]. Generally, for answering a particular question, there is a lot of redundant and irrelevant context information in long document, which will cause some noise and weaken the performance of the model. Suffered from this problem, attention mechanism [9], [10] is used to enable the model to focus on target area within a context paragraph that is most relevant to answer the question.

Up till the present, most of the machine reading comprehension models focus on how to obtain the implications between the question and document with attention mechanism, but they seem to neglect the words position information. In [12], it assumes that those words which are closer to query words are more likely to be the related terms of the query topic. The effectiveness of the position information is proved in [12]–[14]. Our previous work [1] states that the method of focusing on words in specific position coincides with the thought of attention mechanism. So integrating position information into attention mechanism is a natural and useful way.

The associate editor coordinating the review of this manuscript and approving it for publication was Huiji Gao.

TABLE 1. An example of machine reading comprehension from TriviaQA.

Qustion: Which snooker player was simply known as 'The Grinder'? **Answer:** Cliff Thorburn
Excerpt: In 1983, Thorburn became the first player to compile a maximum break at the World Championship. ***Cliff Thorburn*** is one of two snooker players inducted into Canada's Sports Hall of Fame, the other being George Chenier. The slow, determined style of play earned ***Cliff Thorburn*** the nickname "The Grinder".

In this paper, we investigate the effectiveness of position information. After analyzing the role of position information in machine reading comprehension, we investigate three methods of introducing position information: position encoder, position mapping embedding and positional attention-based model. Both position encoder and position mapping embedding encode the position information directly into the text representation, and allow the model to take advantage of the position information automatically. But the positional attention-based model is in a more complex way. The model can be divided into the following steps: (i) We introduce the external knowledge Paraphrase Database (PPDB) [15] and expand the question key words from PPDB as the preparation to take full advantage of position information; (ii) Based on our assumption that if a context word is closer to the question words in context, the model should pay more attention to this word. Therefore, some kernel functions are selected to measure the proximity between the center word of context and other words; (iii) We calculate the position influence with the extended words set of question and kernel functions, then use attention mechanism to incorporate position influence and then compute the attended representation. We also use query-aware attention to obtain the attended query representation; (iv) We use these attended representations to infer the answer by pointer network. After this, to explore the effect of different implementation methods of position information on model performance, we also conducted the experiments within position embedded and position mapping methods.

In the experiments, in order to explore the performance of different implementation methods of position information, we evaluate our method on TriviaQA. The experimental results show that the position information can guide machine reading comprehension.

The work in this paper is an extension of our previous work [1], which proposed a position-based attention mechanism on machine reading comprehension. Based on this, this paper further studies the application and influence of position information, and the previously proposed model is one of the method to utilize position information. At the same time, this paper has carried on the experiment and the analysis to the various aspects of the position influence, explained how to use the position information reasonably. Our contributions are summarized as follows:

- We investigate the effectiveness of position information in machine reading comprehension tasks. Several different methods which introduce position information are experimented and analyzed. The experiments show that the introduction of position information can help the model to answer question.
- We experiment on TriviaQA dataset, and the results prove that the method of using prior hypothesis and attention mechanism is an appropriate application of position information in machine reading comprehension tasks, which is also more effective t han other direct characterization methods. To explore more details of the positional attention, we analyze the influence of σ value on position information and visualize it for more specifically.

II. RELATED WORK

A machine reading comprehension system based on a deep neural model usually requires a large amount of data as a support. Benefiting from the introduction of many large datasets, machine reading comprehension neural model made rapid progress in recently. There are some manually labeled datasets like [16], [17], which are of high quality but limited by a small number and difficult to train complex models. The task of cloze style machine reading comprehension is to predict the missing word in a passage, making it easier to automatically generate large datasets from a large amount of existing natural text. But the construction of datasets for extractive machine reading comprehension is more complicated and difficult. For extractive machine reading comprehension, WikiReading [18], MS Macro [19] are constantly being introduced. [5] releases the Stanford Question Answering (SQuAD) dataset, which orders of magnitude may not be larger than all previous datasets, but it still enjoys an exceedingly huge quantity. Along with its extraordinary qualities, it drives to the culmination in a natural QA task. After that, TriviaQA [6] provides more realistic questions with a larger order of magnitude.

Supported by these large datasets, neural models have been developed. Due to the long-term reasoning involved, attention mechanism component has become popular and important in neural machine comprehension models. Previous works [2], [9], [20], [21] usually adopte the deep neural model with attention mechanism, and explore a lot of how to improve the attention mechanism. AoA Reader [22] calculates the similarity matrix with two-way attention and focus on each other between the query and the passage. BiDAF [9] uses a bi-directional attention to capture the interaction effect between query and context, and calculates the query-to-context attention and context-to-query attention. In the previous works [20], [23], machine reading comprehension task is regarded as the segment of the answer, that is, the position at which the beginning and the end of the answer were predicted. The boundary-based pointer network [2] is then implement for boundary model.

In information retrieval, position information is effective in some tasks. [24] proposes positional language model for document retrieval and proximity heuristic. In [12], it assumes

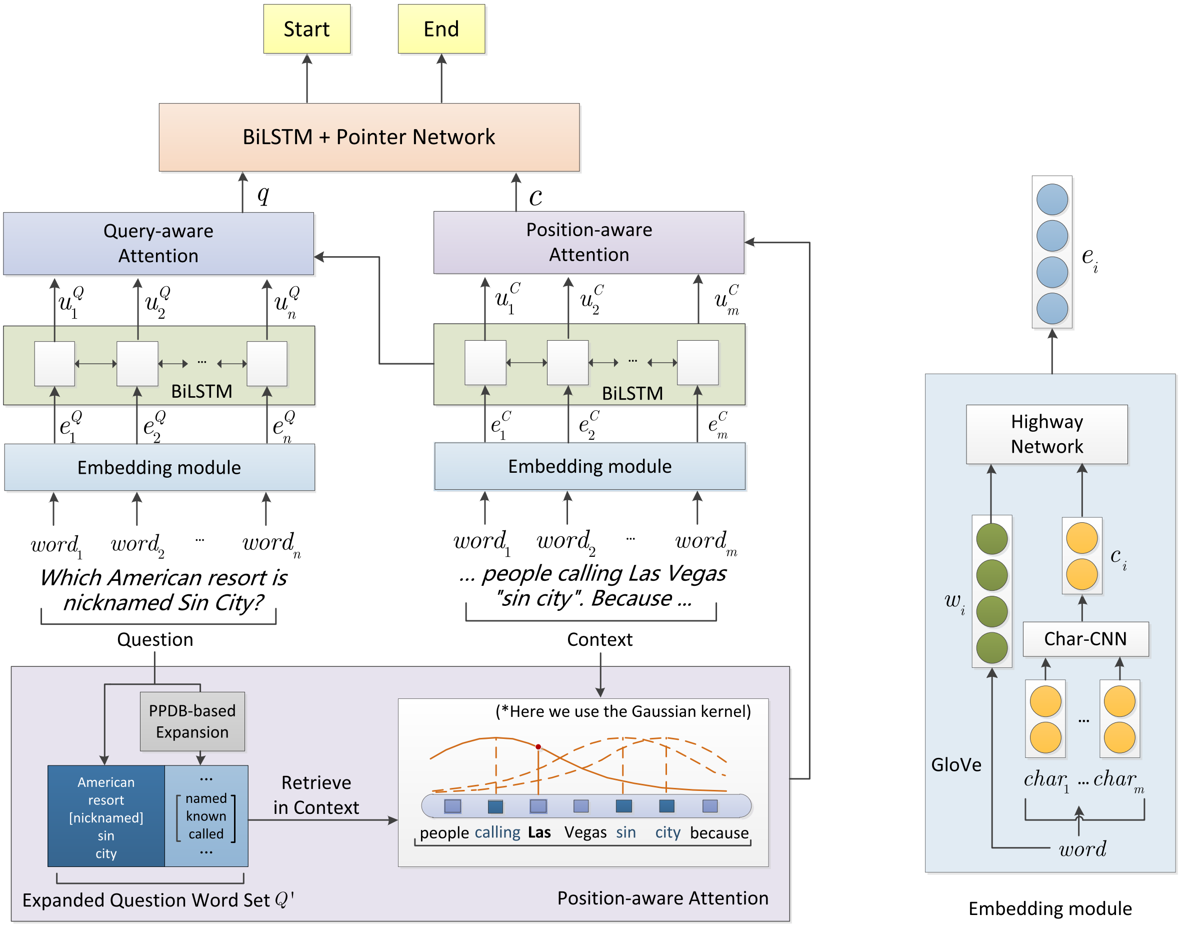


FIGURE 1. The left is overall framework of our position-aware attention-based machine comprehension model, and the embedding module is on the right. As given with the question-passage-answer triple example, we specially show how the question words influence the word "*Las*" with position information. We first get the expansed words of "*nicknamed*", such as "*named*", "*known*", "*called*", etc. After retrieving in context, we then use a Gaussian kernel based on word "*calling*" to get its positional influence to "*Las*" (Question words "*sin*" and "*city*" work in the same way). Attention mechanism is then used to model the position influence into context, and use it with question to infer the answer.

that the proximity between candidate terms and query terms can be exploited in the process of query expansion, because terms which are closer to query terms are more likely to be related to the topic of query. [14] proposes Term Location (TEL) retrieval model. In machine translation, [32] abandons the neural network and only uses attention mechanism to complete the task, then uses position embedding to improve the model. In relation extraction [25] and sentence similarity [26], the position information is also applied with attention mechanism. In our previous work [1], we introduce a position-aware attention-based model for machine reading comprehension.

In order to take full advantage of position information, we expand the words before the attention component to adapt to the complex and diverse expressions of syntactic dataset of machine reading comprehension tasks. [27] proposes PhraseFinder to build thesauri assist query expansion, [28] expands seed lists by recursively querying for synonyms using WordNet [29], and the work [30] uses lexical and phrasal rules of Paraphrase Database(PPDB) to paraphrase questions by replacing words and phrases in them.

III. POSITION INFORMATION IN MACHINE READING COMPREHENSION

Position information is an essential part of natural language understanding tasks. In most languages, the position of words is related to the grammar and the structure of sentences. One reason why Recurrent Neural Network (RNN) [31] is so popular in NLP is that its architecture can implicitly integrate word order information into the model, which is very useful for many tasks, such as machine translation, sequence annotation and language reasoning.

While doing reading comprehension, we notice that human usually locates the inference range to the neighborhood of the question key words first, and then focus on these subsections to answer. The question related paragraph contains the key information to answer the question. Furthermore, the importance of the word is reduced when the word is far from the question key words. The contexts far from the key words are usually redundant with this question. Therefore, the position information of the question words in the context is playing a guiding role in reading comprehension.

In this paper, we mainly investigate the effectiveness of position information in machine reading comprehension. First, we directly encode the absolute position feature, then we introduce our position attention-based machine reading comprehension model, finally we compare the other position mapping method that can also make use of position information.

A. POSITION ENCODER

There are other ways to introduce position information, such as position embedding. This method directly introduces position information to guide the model at the embedded layer. At the work of [32], the model contains no recurrence and no convolution. However, in natural language tasks such as machine translation tasks, the position information of the forward and backward context plays an important role in the prediction of the model. In order to introduce sequence position information, position encoder is used at embedding layer. In order to compare the effectiveness of our method of introducing position information, we add this component as a contrast experiment.

In our contrast experiment, we add position embedding to the embedding layer module of the baseline model. More specifically, we use cosin and sine function to compute the position embedding matrix:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d}) \tag{1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d}) \tag{2}$$

where d is the dimension of word embedding of model, we keep the dimension of the encoder consistent with the dimension of word embedding d, and then add the two together after this. *pos* is the position and i is the dimension. Then we fix this instead of trainable during training. The work of machine translation [32] find that using trainable position embedding can not improve the performance, but at the same time increases the amount of calculation.

B. POSITION-AWARE ATTENTION IN MACHINE READING COMPREHENSION

As we describe in previous, we should pay more attention to the words in some specific areas. This is consistent with the idea of attention mechanism. We mainly introduce the detail of our position-aware attention-based model that is backward for the inference text paragraph. An overview of our paragraph-level machine reading comprehension model is shown in Figure 1.

1) QUESTION AND PASSAGE ENCODER

After input the neural language sentences of question and passage, we embed each word on character-level and word-level as show in the right side of Figure 1. First, following the previous method [33], we use Char-CNN to generate character embedding ($\{c_i^Q\}_{i=1}^m$ for question and $\{c_i^P\}_{i=1}^n$ for passage, m is the length of question and n is the length of passage) for each word. At the same time, by looking up from pre-trained 300-dimensional Glove word vectors [40], we can obtain the fixed-size word embedding ($\{w_i^Q\}_{i=1}^m$ for question and $\{w_i^P\}_{i=1}^n$ for passage) of each word.

After getting character-level and word-level embedding, we then input the concatenate of these two embeddings through Highway network to get the final representation($\{e_i^Q\}_{i=1}^m$ for question and $\{e_i^P\}_{i=1}^n$passage). The purpose of character embedding is to help model to solving the out-of-vocabulary (OOV) problems, and the goal of using highway network is to calculate the representation feature in different granularity levels.

In order to be benefited from both the forward and backward Long Short-Term Memory Network (LSTM) [35], we use bi-directions LSTM (BiLSTM) to compute the representation u_i^Q for question and u_i^C for passage separately with the question and context representations from the previous layer:

$$\mathbf{u}_i^Q = BiLSTM_Q(\mathbf{u}_{i-1}^Q, \mathbf{e}_i^Q) \tag{3}$$

$$\mathbf{u}_i^C = BiLSTM_C(\mathbf{u}_{i-1}^C, \mathbf{e}_i^C) \tag{4}$$

2) POSITION-AWARE ATTENTION

In this part, we focus on how to apply the position information to enhance the answer prediction. First, we introduce the PPDB-based word expansion, which is the beforehand component that could make model take more advantage of the position information. Then we introduce several kernel function which can measure the word relation of our prior assumption. After that, we implement the attention mechanism to model the position information for the final predict. Our algorithm for calculating position effects is described in Algorithm 1.

a: PPDB-BASED KEY WORD EXPANSION

As shown in the previous section, we notice that people will locate a particular sub-context to answer the question based on the key words of question when doing reading comprehension. Correspondingly, we believe that the context word which is closer to question words should be more important in machine reading comprehension. Therefore, we should select the question words first before we find which context words is closer to these words. If we only use the co-occurrence question words when calculating the position information, there will be many information loss when the question words appear as synonyms or anamorphic phrases.

Bilingual pivoting [36] is one of the most well-known approaches to paraphrasing, which follows the work of the statistical phrase-based translation (SMT) [37]. Based on the assumption that two English strings which can be translated to the same foreign string have the same meaning, PPDB [15] is created as a syntactic paraphrases. Because there are more various representations than ordinary synonym dictionaries, we choose PPDB to obtain some lexical representations of question words, and calculate the query-to-context position influence with expanded word set Q'.

Algorithm 1 Steps for Introducing Positional Attention (With Gaussian Kernel)

```
Input: The set of question word, $Q$; The sequence of context
    words, $C = (c_1, c_2, \ldots, c_l)$, which arranged in word order.
Output: Position influence vector $\mathbf{p_i}$;
Expand the set of question word $Q$ with help of PPDB to
    get the expanded question word set $Q'$;
for each $c_i \in C$ do
    if $c_i \in Q'$ then
        init Array pos_i $\in \mathbb{R}^l$
        $pos\_i_j = K_{Gaussian}(i, j)$
init Array pos $\in \mathbb{R}^l$
for each pos_i do:
    pos $= \sum$ pos_i in each dimension
$\mathbf{p_i} = \mathbf{pos_i P}$
return $\mathbf{p_i}$;
```

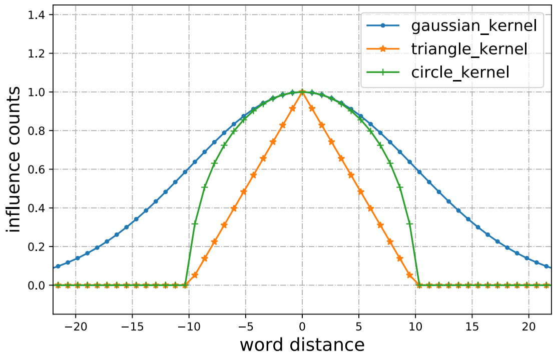


FIGURE 2. Proximity-based kernel functions. We set $\sigma = 10$ for all kernels.

For each question words except stopword, we retrieve from PPDB to obtain the relevant words in lexical and phrasal rules. Therefore, for a question which has a key word set $Q = \{q_i\}_k^{i=1}$ and the extend words set $Q' = \{\{q_i\}_k^{i=1}, \{q_i'\}_k^{i=1}\}$ is then used to calculate the position distance.

b: POSITIONAL INFLUENCE

In this step, corresponding to our previous assumption that the words which is close to the question words are more likely to be the answer, for different distances u we need a function $f(u)$ to measure the following mapping relationships:

$$f(u_1) > f(u_2), when\ |u_1| < |u_2| \tag{5}$$

Following the previous works in information retrieval [14], there are several kernel functions can be used to measure this mapping, such as Gaussian kernel, Triangle kernel and Circle Kernel. Figure 2 shows the distribution curve of these kernel function, and thees kernels are formally shown as below:

$$K_{Gaussian}(u) = exp[\frac{-u^2}{2\sigma^2}] \tag{6}$$

$$K_{Triangle}(u) = \begin{cases} 1 - \left|\frac{u}{\sigma}\right| & u < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$K_{Circle}(u) = \begin{cases} \sqrt{1 - (\frac{u}{\sigma})^2} & u < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where u is the variable that affects the output of kernel function. In our task, we define u as the position distance in the context of two words. The study [12] shows that Gaussian kernel is an effective way in most cases. In order to facilitate the description of the model detail, we use Gaussian kernel to describe later. We model the weight that how a question word influenced a context word as below:

$$K_{Gaussian}(c, q) = exp[\frac{-(p_c - p_q)^2}{2\sigma^2}] \tag{9}$$

where c and q are the context word and question word which occurs in context, p_c and p_q is the absolute position of word c and word q in the context. The normalization parameter σ controls the function distribution curve as shown in Figure 2. The characteristics of the Gaussian kernel make the weight of the context words which are closer to question words boost most, which also measures appropriately to our assumption.

For each context word i, we calculate the position influence of context word j as:

$$pos_i(j) = K_{Gaussian}(i, j)I[j \in Q'] \tag{10}$$

where Q' is the question word set we assigned as the informational word, $I[\cdot]$ is indicator function, when the condition is true, the value is set to 1, otherwise it will be 0. At the same time, to take the word relevance into account, we also try to calculate the position influence by multiplying a similarity weight $Sim(i, j)$ between the question word and the extent relevant word, but it is worthless because of the little boost and a huge cost of time:

$$pos_i'(j) = K(i, j) * Sim(i, j) * I[j \in (Q' - Q)] \tag{11}$$

For each context word of position, we calculate the position embedded vector $\mathbf{p_i}$ with a shared position embedding matrix $\mathbf{P}$:

$$\mathbf{p_i} = \mathbf{pos_i P} \tag{12}$$

c: POSITION-AWARE ATTENTION

After we obtain the position embedded vector, we calculate the attention weight r_i for each hidden state $\mathbf{u}_i^C$ as:

$$r_i = \mathbf{v}^\top tanh(\mathbf{W}_h \mathbf{u}_i^C + \mathbf{W}_p \mathbf{p}_i) \tag{13}$$

$$a_i = \frac{exp(r_i)}{\sum_{j=1}^{n} exp(r_j)} \tag{14}$$

where a_i is the attention weight which can measure the important weight of each context word. We can obtain the attended vector as follow:

$$\mathbf{c}^p = \sum_{i=1}^{n} a_i \mathbf{u}_i^C \tag{15}$$

3) QUERY-CONTEXT ATTENTION:

In order to get the query-context interactive information, a two-directional attention [9] is used. First, we calculate the similarity of each word:

$$s_{ij} = \mathbf{W}_1\mathbf{u}_i^C + \mathbf{W}_2\mathbf{u}_j^C + \mathbf{W}_3(\mathbf{u}_i^C \odot \mathbf{u}_j^C) \tag{16}$$

Then wen can attend question and context representation as below:

$$p_{ij} = \frac{exp(s_{ij})}{\sum_{j=1}^{n} exp(s_{ij})} \tag{17}$$

$$\mathbf{c}_i^q = \sum_{j=1}^{m} p_{ij}\mathbf{u}_i^Q \tag{18}$$

In another direction:

$$m_i = max(a_{i1}, \ldots, a_{im}) \tag{19}$$

$$p_i = \frac{exp(m_i)}{\sum_{j=1}^{n} exp(m_i)} \tag{20}$$

$$\mathbf{q}^c = \sum_{i=1}^{n} p_i\mathbf{u}_i^C \tag{21}$$

Then we concat $\mathbf{u}_i^C$, $\mathbf{c}_i^q$, $\mathbf{q}^c \odot \mathbf{u}_i^C$, $\mathbf{c}^p \odot \mathbf{u}_i^C$ as the final vector, which combines bidirectional information and position information.

4) ANSWER PREDICTION AND TRAINING:

The answers in TriviQA are usually a continuous text fragment, we use Bi-LSTM and pointer networks [2] to predict answer start and end position score. The self-attention is also applied because it can bring the internal interaction. The softmax operation is applied to the start and end scores to produce start and end probabilities. At the training step, we choose the sum of the negative log probabilities of the true start and end tokens by the predicted distributions as the minimize function to optimize our model.

C. POSITION MAPPING EMBEDDING

Before constructing the positional influence vector, we use a shared position embedding matrix to calculate the position influence vector as shown in Formula 10. Instead of this, we also experiment with using a special positional effect that uses Gaussian distribution to map each position, with the idea that the influence of each concrete distance follows the distribution of Gaussian. So we use the Gaussian kernel to map the columns of the initial matrix:

$$\mathbf{P} \sim GaussianKernel(u, \sigma) \tag{22}$$

This approach simply uses the Gaussian distribution to exploit position information, but does not design for each word.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

In this section, we first introduce the machine reading comprehension dataset TriviaQA we used and the auxiliary dataset PPDB for question expansion, and briefly introduce the dataset text pre-processing we utilized before the experiment. After this, two matrices to evaluate the performance of machine reading comprehension model are explained. Finally we introduce the implementation details of the experiment.

TABLE 2. TriviaQA datasets statistics.

Domain	QA pairs	Paragraphs
Wikipedia	77k	138k
Web	95k	662k

1) DATASETS

a: TriviaQA DATASET

TriviaQA dataset includes 95K question-answer pairs, and each question has 6 evidence documents on average [6]. It is authored by trivia enthusiasts and independently gathered evidence documents, which makes the dataset more complex. The contexts of over 650K context-query-answer triples are/ automatically generated from search results of either Wikipedia or Web. The average length of contexts in TriviaQA is about 2895 words [21]. Chapters for inferential answers are divided into two domains according to the source of their acquisition: **Wikipedia** articles and **Web** search results.

Before using the TriviaQA dataset, we need some necessary text pre-processing first. There are usually many small fragments in the documents of the dataset. Questions in the TriviaQA dataset usually contain multiple related passages, which are ranked in the dataset. We find that passages with higher ranking usually contain more information about reasoning answers and answer spans, so when merging paragraphs, we prefer to use those ones. We merge these related paragraphs into documents before using the data, then take a maximum of 500 words per documents and make sure the answers in this document. The QA pairs of the dataset are randomly partitioned into a training set (80%), a development set (10%) and a test set (10%).

2) EVALUATION METRICS

There are two assessment methods in machine reading comprehension tasks: Exact Match (**EM**), the predicted answer span matches the ground truth answer exactly. **F1** score, the overlap ratio between predicted answers and ground truth answer.

3) IMPLEMENTATION DETAILS

At the embed layer, we initialize the word embedding with the 300 dimension Glove word embeddings trained on 840B tokens [40], and obtain the char embedding by using char-CNN [33] with 100 dimension. During the training phase, the model is optimized using AdaDelta optimizer [41] and maintain an exponential decay rate of 0.999. For char-CNN, BiLSTM and linear transformation we adopt dropout rate with 0.2 Considering the large number of TriviaQA datasets

TABLE 3. Result on the TriviaQA.

		Full		Verified	
Model	Domain	EM	F1	EM	F1
Classifier [6]	Wiki	22.46	26.52	27.23	31.37
BiDAF [9]		40.32	45.91	44.86	50.71
Smarnet [38]		42.41	48.84	50.51	55.90
MEMEN [39]		43.16	46.90	49.28	55.83
Our model with:					
-Gaussian Kernel		**43.82**	**49.17**	52.27	58.38
-Triangle Kernel		43.53	49.04	**52.83**	**58.85**
-Circle Kernel		42.91	48.48	51.42	57.55
Classifier [6]	Web	24.64	29.08	27.38	31.91
BiDAF [9]		41.08	47.40	51.38	55.47
Smarnet [38]		40.87	47.09	51.11	55.98
MEMEN [39]		**44.25**	48.34	**53.27**	57.64
Our model with:					
-Gaussian Kernel		43.24	**49.93**	52.87	57.72
-Triangle Kernel		43.05	49.41	51.92	**58.17**
-Circle Kernel		42.64	48.53	51.83	56.90

we evaluated on, we set the hidden state size as 300 and minibath size as 60.

For the hyper-parameter σ of the kernel function, we experiment with a set of values from 5 to 50 in increments of 5 to compare the performance of different σ with different kernel function, and we will analyze it later.

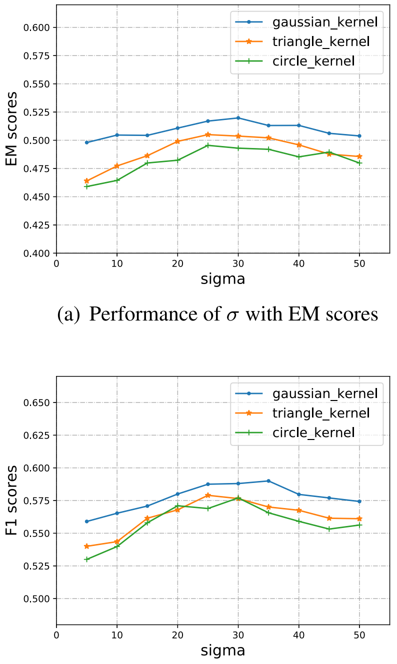
B. MAIN RESULTS

The performance on TriviaQA dataset is given in Table 3. As we describe in previous, the gathered method of TriviaQA makes the dataset closer to the machine reading comprehension task which people usually deal with, thus we mainly evaluate the model on TriviaQA. In order to achieve best performance on dataset, many models adopt complex structures or add many sub-components. In our work, we mainly explore the role of position information in machine reading comprehension model, so we mainly compare it with the baseline of dataset, and then analyze its effects through an ablation study. The baseline model Classifier is follow the work [42], which is to maximize the conditional probability of the correct answer and minimize the conditional probability of the wrong answer when the question is given. The BiDAF model [9] use a LSTM-based model with two-directional context-query attention.

In order to evaluate the performance of different kernel functions, we experiment with Gaussian kernel, Triangle kernel and Circle kernel on TriviaQA separately. As shown in the results from Table 3, the Gaussian kernel function outperforms Triangle kernel and Circle kernel in most cases. Explaining from the principle of these kernel functions, the Triangle kernel and Circle kernel influences may vanish at a distance. In other words, they only focus on the content of a certain window size. But Gaussian kernel still has a subtle influence in the more distant context. With the help of positional attention, our model achieves a 3% improvement compare to baseline model BiDAF.

TABLE 4. Ablation study of position-aware attention and PPDB-base question word expansion.

Model	EM	F1
Baseline (General model with general attention)	40.12	45.93
+ position-aware attention	46.07	53.21
+ position-aware attention + words expansion	47.52	53.61



(a) Performance of σ with EM scores

(b) Performance of σ with F1 scores

FIGURE 3. Performance of the parameter σ of different kernels.

C. EFFECTS OF POSITION-AWARE ATTENTION AND PPDB-BASE QUESTION WORD EXPANSION

In order to specifically analyze the effect of each module, we have done separate experiments on the two parts. The experimental results are presented in Table 4.

As shown in the experimental results, the position-aware position can bring 5% improvement compare with the model without attention. It means that in most case the context of documents complies with the previous assumption that the word which is near to the question words is more likely to be the answer. Therefore, by model this priori knowledge into our approach, model can focus more on these words that are more likely to be the answer. But at the same time, because of the introduce of this priori knowledge, the model may perform worse in the cases that the question which need to infer the answer across several sentences or even paragraphs. In this case, the priori assumption limits the correct attention of the model to the context.

For the part of PPDB-base question word expansion, this improvement is expected and obvious. Because of the diversity of expressions in the machine reading comprehension dataset, and one of the reasons we choose the TriviaQA dataset as the evaluation dataset is that this data collection comes from real web corpus, which leads to many personal, oral and phrasal expressions of the answer paragraph. Our pipeline process is to retrieve the question words in the article first, and then to model the position information based on these question words. Therefore, the expansion of question words will directly affect the performance of the model.

For the current dataset of machine comprehension, even question words may be the same word in many question. With the aid of PPDB-based expansion, the performance of our model can improve by about 1%.

D. INVESTIGATION OF PROPAGATION SCOPE σ

According to the mathematical definition of kernels in the previous section, all three kernels have a hyperparametric σ to turn. This parameter controls the distribution curve of the kernel function. In our task, it represents the change degree of the influence of the relative position between words. Optimized setting of parameters is usually adjusted according to specific scenarios. If we want the influence of position information to be small and wide-ranging, that is, the curve tends to be flat and wide, then σ should be increased appropriately. Ideally, this parameter would be better if it could be fine-tuned to suit a specific individual problem to the desired scenario. In our work, we set up a number of experiments to explore the effects of different lengths on the kernel function.

As can be seen from the Figure 3, the performance of the Gaussian kernel is significantly better than other kernel functions. In addition, when σ is about 25 to 35, the performance of several kernel is nearly the best. This value is mainly related to the data set because it represents the positional characteristics of the data. As a hyper parameter, its superficial meaning indicates the sensitivity of the question to the influence of words in nearby positions. A larger value means that the closer the word is influenced, the farther the word influence is. For Triangle Kernel and Circle Kernel, since the distribution is only meaningful within the range of less than σ, it is equivalent to a sliding window, and the window size is equal to $2\sigma + 1$, that is, only the information of these words is considered. When we limit σ to a smaller value, they are more suffered with this change than Gaussian kernel.

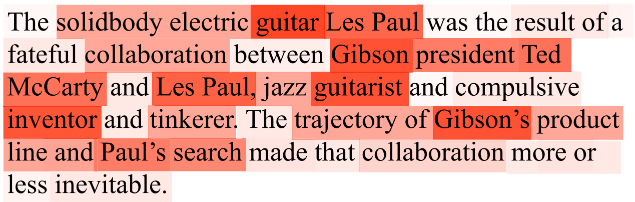
E. CONTRAST EXPERIMENT OF POSITION INFORMATION

We have made comparative experiments and discussed the methods of other kinds of position information. The experimental results are shown in Table 5.

Adding position embedding before contextual layer is a more direct feature method. Previous work is mostly to add relevant information of the neural language when there is no recurrent architecture to bring the word order information. In our experiment, it can not work well because its role may partly repetitive with some part of RNN, which might bringing redundant and less helpful information. This methods simply introduce position information into the representation, but do not design for the characteristics of our proposed prior assumption of position information in machine reading comprehension. And the use of Gaussian position mapping matrix could bring litter improvement since the the initial set up could be change after calculate.

TABLE 5. Effects of Position Embedding and Gaussian Position Matrix. Here we use the model with general attention as baseline model and use our position-aware attention-based model within Gaussian kernel as "Our model".

Method	EM	F1
Baseline model	40.12	45.93
Baseline model with position embedding	40.37	46.08
Our model	47.52	53.61
Our model with Gaussian position matrix	46.93	54.07



The solidbody electric guitar Les Paul was the result of a fateful collaboration between Gibson president Ted McCarty and Les Paul, jazz guitarist and compulsive inventor and tinkerer. The trajectory of Gibson's product line and Paul's search made that collaboration more or less inevitable.

FIGURE 4. The attention heatmap for a context fragment with question "Which guitar innovator and player has a range of Gibson Guitars named after him?".

F. ATTENTION VISUALIZATION

We draw a word-level attention heatmap of one context fragment with question "Which guitar innovator and player has a range of Gibson Guitars named after him?" for intuitive analysis in Figure 4. This fragment comes from a relevant passage of more than 3000 words. Here we leave aside the other redundant information in the context and analyze this text in detail. As we can see, the focus of the model is mainly on several key question words "guitar", "Gibson" and the words around them. With the PPDB word expansion, the word "inventor" also has a higher weight. With our method, the model can reasoning the informational text area in a more direct way and provide better answer.

V. CONCLUSION

In this paper, we explore how to benefit from position information for enhancing the machine reading comprehension. More specifically, we use position encoder, position mapping embedding to introduce position information directly, and we also propose an optimized position-aware attention-based machine reading comprehension model, and introduce external knowledge to take full advantage of position information. By analyzing some details of the use of position information and comparing with these methods to combine position information, our work shows position information could enhance the machine reading comprehension, and attention method is better than others in the use of position information.

REFERENCES

[1] W. Liu, J. Zhao, M. Li, S. Li, and J. Guo, ''Position-aware attention for enhancing the machine comprehension,'' in *Proc. Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, 2018, pp. 20–24.

[2] S. Wang and J. Jiang, ''Machine comprehension using match-LSTM and answer pointer,'' 2016, *arXiv:1608.07905*. [Online]. Available: https://arxiv.org/abs/1608.07905

[3] D. Chen, A. Fisch, J. Weston, and A. Bordes, ''Reading wikipedia to answer open-domain questions,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–10.

[4] B. Dhingra, H. Liu, Z. Yang, W. W. Cohen, and R. Salakhutdinov, ''Gated-attention readers for text comprehension,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–15.

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, ''SQuAD: 100,000+ questions for machine comprehension of text,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1–10.

[6] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, ''TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–11.

[7] Z. Chen, R. Yang, B. Cao, Z. Zhao, D. Cai, and X. He, ''Smarnet: Teaching machines to read and comprehend like human,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[8] F. Hill, A. Bordes, S. Chopra, and J. Weston, ''The goldilocks principle: Reading children's books with explicit memory representations,'' 2015, *arXiv:1511.02301*. [Online]. Available: https://arxiv.org/abs/1511.02301

[9] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, ''Bidirectional attention flow for machine comprehension,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.

[10] S. Sukhbaatar, J. Weston, and R. Fergus, ''End-to-end memory networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.

[11] J. Cheng, L. Dong, and M. Lapata, ''Long short-term memory-networks for machine reading,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1–11.

[12] J. Miao, J. X. Huang, and Z. Ye, ''Proximity-based rocchio's model for pseudo relevance,'' in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2012, pp. 535–544.

[13] J. Zhao, J. X. Huang, and B. He, ''CRTER: Using cross terms to enhance probabilistic information retrieval,'' in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2011, pp. 155–164.

[14] B. Liu, X. An, and J. X. Huang, ''Using term location information to enhance probabilistic information retrieval,'' in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 883–886.

[15] J. Ganitkevitch, B. van Durme, and C. Callison-Burch, ''PPDB: The paraphrase database,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 758–764.

[16] M. Richardson, C. J. C. Burges, and E. Renshaw, ''MCTest: A challenge dataset for the open-domain machine comprehension of text,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 193–203.

[17] J. Berant, V. Srikumar, P.-C. Chen, A. V. Linden, B. Harding, B. Huang, P. Clark, and C. D. Manning, ''Modeling biological processes for reading comprehension,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1499–1510.

[18] 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 *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1–11.

[19] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, ''MS MARCO: A human generated machine reading comprehension dataset,'' Tech. Rep., 2016, p. 660. [Online]. Available: https://scholar.google.com.hk/scholar?hl=zh-CN&as_sdt=0%2C5&q=MS+MARCO%3A+A+human+generated+machine+reading+comprehen-+sion+dataset%2C&btnG=

[20] C. Xiong, V. Zhong, and R. Socher, ''Dynamic coattention networks for question answering,'' 2016, *arXiv:1611.01604*. [Online]. Available: https://arxiv.org/abs/1611.01604

[21] W. Wang, M. Yan, and C. Wu, ''Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering,'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 1–10.

[22] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, ''Attention-over-attention neural networks for reading comprehension,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–10.

[23] K. Lee, S. Salant, T. Kwiatkowski, A. Parikh, D. Das, and J. Berant, ''Learning recurrent span representations for extractive question answering,'' 2016, *arXiv:1611.01436*. [Online]. Available: https://arxiv.org/abs/1611.01436

[24] Y. Lv and C. Zhai, ''Positional language models for information retrieval,'' in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2009, pp. 299–306.

[25] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, ''Position-aware attention and supervised data improve slot filling,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 35–45.

[26] Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An, ''Enhancing recurrent neural networks with positional attention for question answering,'' in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 993–996.

[27] Y. Jing and W. B. Croft, ''An association thesaurus for information retrieval,'' in *Intelligent Multimedia Information Retrieval Systems and Management*, vol. 1. 1994. [Online]. Available: https://scholar.google.com.hk/scholar?hl=zh-CN&as_sdt=0%2C5&q=An+association+thesaurus+for+information+retrieval&btnG=

[28] N. Godbole, M. Srinivasaiah, and S. Skiena, ''Large-scale sentiment analysis for news and blogs,'' in *Proc. ICWSM*, vol. 7, 2007, pp. 219–222.

[29] G. A. Miller, ''WordNet: A lexical database for English,'' *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[30] L. Dong, J. Mallinson, S. Reddy, and M. Lapata, ''Learning to paraphrase for question answering,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1–12.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning representations by back-propagating errors,'' *Cogn. Model.*, vol. 5, no. 3, p. 1, 1988.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[33] Y. Kim, ''Convolutional neural networks for sentence classification,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–6.

[34] R. K. Srivastava, K. Greff, and J. Schmidhuber, ''Highway networks,'' 2015, *arXiv:1505.00387*. [Online]. Available: https://arxiv.org/abs/1505.00387

[35] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] C. Bannard and C. Callison-Burch, ''Paraphrasing with bilingual parallel corpora,'' in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, 2005, pp. 597–604.

[37] P. Koehn, F. J. Och, and D. Marcu, ''Statistical phrase-based translation,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, vol. 1, 2003, pp. 48–54.

[38] Z. Chen, R. Yang, B. Cao, Z. Zhao, D. Cai, and X. He, ''Smarnet: Teaching machines to read and comprehend like human,'' 2017, *arXiv:1710.02772*. [Online]. Available: https://arxiv.org/abs/1710.02772

[39] B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai, and X. He, ''Memen: Multi-layer embedding with memory networks for machine comprehension,'' 2017, *arXiv:1707.09098*. [Online]. Available: https://arxiv.org/abs/1707.09098

[40] J. Pennington, R. Socher, and C. Manning, ''Glove: Global vectors for word representation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[41] M. D. Zeiler, ''ADADELTA: An adaptive learning rate method,'' 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701

[42] D. Chen, J. Bolton, and C. D. Manning, ''A thorough examination of the cnn/daily mail reading comprehension task,'' in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1–11.

YAJING XU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2008, where she is currently an Associate Professor with the School of Information and Communication Engineering. Her current research interests include pattern recognition, machine learning, and information retrieval.

WEIJIE LIU received the B.Eng. degree from the Central South University of Forestry and Technology, China, in 2016. He is currently pursuing the M.S. degree with the Beijing University of Posts and Telecommunications. His research interests include machine learning, deep learning, and natural language processing.

GUANG CHEN received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2006, where he is currently an Associate Professor with the School of Information and Communication Engineering. His research interests include information retrieval, and text mining and visualization.

BOYA REN received the M.A. degree from Peking University. She is currently a Research Associate with the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her research interests primarily include international communication and crisis management.

SIMAN ZHANG received the master's degree from The University of Edinburgh, in 2017. She is currently a Researcher with the DOCOMO Beijing Communications Laboratories Company Ltd., whose main research interest is natural language processing.

SHENG GAO received the Ph.D. degree from the Sixth University of Paris, France, in 2011. He is currently an Associate Professor in pattern recognition and intelligent system with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. He is a member of the Big Data Professional Committee of the Chinese Computer Society (CCF), an expert of the National Natural Science Foundation, and a Guest Researcher of the Sixth University of Paris.

JUN GUO received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree from Tohuku Gakuin University, Sendai, Japan, in 1993. He is currently a Professor and the Vice President with BUPT. He has published over 200 articles in the journals and conferences, including *Science*, *Scientific Reports* (Nature), the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition*, *Association for the Advancement of Artificial Intelligence*, *Conference on Computer Vision and Pattern Recognition*, *International Conference on Computer Vision*, and *Special Interest Group on Information Retrieval*. His current research interests include pattern recognition theory and application, information retrieval, content-based information security, and bioinformatics.