Enhancing Machine Comprehension Using Multi-Knowledge Bases and Offline Answer Span Improving System

Feifei Xu, Wenkai Zhang, Haizhou Du, Shanlin Zhou

School of Computer Science and Technology, Shanghai University of Electric Power, China
xufeifei1983@hotmail.com, wkzhang@mail.shiep.edu.cn, haizhou.du@shiep.edu.cn,
zhoushanlin@mail.shiep.edu.cn

Abstract

Machine Reading Comprehension (MRC) is a challenging but meaningful task in natural language processing (NLP) that requires us to teach a machine to read and understand a given passage and answer questions related to that passage. In this paper, we present a rich knowledge-enhanced reader (RKE-Reader), a hierarchical MRC model that employs double knowledge bases with an NER system as its knowledge enhancement unit. Besides, we are the first to propose an offline answer-imporving method to help model to determine the uncertain answer without extra online training process. Our experimental results indicate that on most datasets, the RKE-Reader significantly outperforms most of the published models that do not have knowledge base, especially on datasets that need commonsense reasoning. And the ablation study also reflects that external knowledge bases and answer-selecting unit do make a positive contribution in the entire model.

Keywords: Machine reading comprehension, Knowledge Bases, Knowledge enhanced system, Answer improvement mechanism

1 Introduction

As one of the important works of natural language processing (NLP), the goal of machine reading comprehension is to train machines to comprehend text. MRC can be classified as a natural language understanding (NLU) task. NLU aims to train models not only to learn to represent, but also to understand, and eventually, to generate the target output.

Figure 1 and Figure 2 are the most recent development status of MRC and its relevant fields. Figure 1 reflects that an increasing number of literatures about contextualized language models (CLMs), MRC and knowledge enhanced systems have been published. MRC is becoming a hot topic in NLU in recent years. Also, as the Figure 2 shows, researchers are making efforts to add a variety of components such as improved attention mechanisms and word embeddings to improve their performances in NLU tasks(we often call them "ensemble models"). It can be concluded from the chart that improved attention mechanism and knowledge graph are widely used in kind of NLU tasks, like recommendation [1]-[2] and QA systems. We can also find that we have more than 200 published papers about MRC in the past 3 years, but a minority of them adopt knowledge bases in their works.

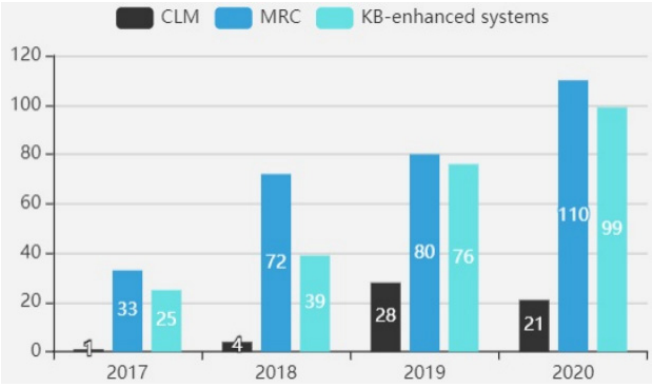

Figure 1. The number of papers published about CLM, MRC and KB-enhanced systems in recent 4 years

The early MRC models, consist of long-short term memory networks and classic attention mechanisms that do not contain knowledge bases, which have an obvious flaw: they often give a high attention score to a specific entity mentioned in the question and try to match that exact entity to words in the given passage without recognizing variants or similar words that may be used in place of that entity (e.g., past tense). Since they cannot match the exact entity to similar words in the given passage, they can easily get confused and output a wrong or incomplete answer.

In passage A of Table 1, since the keyword "year" is mentioned in the question, and "in" exists in both passage and question, we can easily find the answer "2012" in the passage, the attention mechanism of the model finds it easily as well. However, in passage B, according to the question, we need to find a kind of "food" in the text, but the entity "food" does not exist

*Corresponding Author: Wenkai Zhang; E-mail: wkzhang@mail.shiep.edu.cn
DOI: 10.53106/160792642021092205013

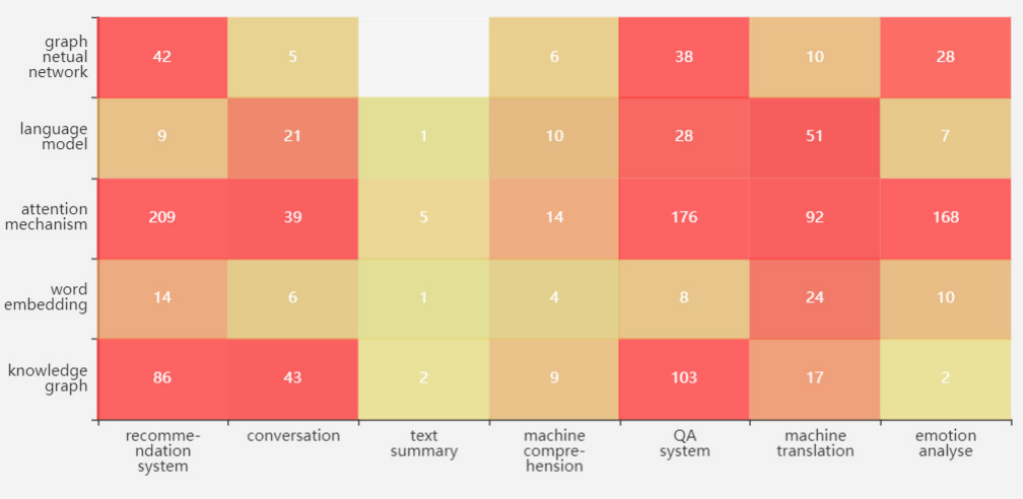

Figure 2. The number of papers in NLU research area in recent 3 years (counted by keywords)

Table 1. Three QA examples in common MRC datasets

PassageA	**In** early 2012, NFL Commissioner Roger Goodell stated that the league planned to make the 50th Super Bowl "spectacular" and that it would be "an important game for us as a league".
Question	**In** which year did Roger Goodell call Super Bowl 50'an important game for us as a league'?
Answer	2012
PassageB	Korean cuisine is probably best known for kimchi, a side dish which uses a distinctive fermentation process of preserving **ingredients**, most commonly cabbage.
Question	In Korea, which **food** was commonly used when making "kimchi"?
Answer	cabbage
PassageC	**Rollo** ultimately settled Normandy and parts of the Atlantic coast included Danes, Norwegians, Norse Gaels, Orkney Vikings, possibly Swedes, and Anglo-Danes from the English Danelaw under Norse control...
Question	**Who** upon arriving gave the original Viking settlers a common identity?
Answer	**Rollo**

in the passage. Instead, we have the synonym, "ingredients" and its sub-concept, "cabbage". If we have a knowledge base in our model, the model can easily find the relation "food – RelatedTo – ingredients" or "cabbage – IsA –food", which helps it to predict the answer "cabbage" quickly and correctly.

Additionally, we may encounter a sentence, like passage C, which contains many unknown words (UNKs) that do not exist in the given vocabulary file. In this passage, "Rollo", "Normandy", "Nores", "Gaels", "Orkney", and "Danelaw" are all UNKs, and the question requires machine to find a "person" who is mentioned in the paragraph. Traditional MRC models often fail to distinguish which "UNK" is the name of "target person" and which is a common entity name. If we have a recognition component that can inform the model about which "UNK" is the name of a person and which "UNK" is the name of a place before encoding the sentence, we believe that the model can correctly and accurately identify the right entity.

To tackle the problems mentioned in the demo passages, we present RKE-Reader. "RKE" refers to "Rich Knowledge-Enhanced". Compared to traditional MRC models, we mainly focus on teach machines "how to think" and "how to express". We add two external knowledge bases(KBs): WordNet3.0 [3], ConceptNet5 [4] and a name recognition (NER) component to help the model with its comprehension tasks. Then, we proposal to use NER system as an auxiliary component to distinguish some entities in the text, such as the name of a person, a certain place and so on. We also give a novel algorithm about how to select a better answer from a group of backup answers generated by model. Our experimental result shows that the MRC model buffers with rich knowledge significantly outperforms other traditional single or ensemble models on most of the datasets, especially datasets that require machines to make reasoning decisions.

Our main contributions are as follows:

- We present double knowledge bases with NER system as the knowledge enhance- ment unit, which scale is larger than any knowledge modules in published works as we know.
- We are the first to present an offline answer-imporving system, which contains an answer extract unit and a contextualize analyse unit, to help the model determine the uncertain answers without any extra training steps.
- We propose a rich-knowledge enhanced model for MRC, which consists of knowledge enhancement unit and answer-selecting unit mentioned above. The result of our experiments shows that our model significantly outper- forms most of the published ordinary MRC models.

- We find that external KBs do enhance the stability of models and make models hard to be influenced by distract options. The positive effect that knowledge enhancement unit brings is proved in our ablation study.

2 Related Work

2.1 General MRC Model Structure

Ensemble model, which consists of a series of improved components, is widely used in research, and especially application of NLP and some other fields like combining neural network to improve the stability of service prediction [5], enhancing efficiency of task scheduling with the combination of multi cost-aware algorithms [6] and so on. From the aspect of MRC tasks, the universal ensemble model of MRC mainly contains following main parts: embedding and encoding unit, interaction unit and prediction unit.

The task of the embedding unit is to convert words in the passage to their corresponding vector representations. Before the contextualized language models (CLMs) were proposed, researchers used traditional LMs like one-hot labels or distributed word representations such as Word2Vec [7] and GloVe [8]. The fatal problem of the common n-gram language model is that it fails to represent the relationships between words. Even worse, as the size of model's vocabulary grows, the scale of the word vector rises sharply, which becomes an obstacle in the calculation process. Only when CLMs had been developed, NLU works including MRC were able to make a series of remarkable achievements in a short period.

For encoding and interaction units, some of the previous works use convolutional neural networks (CNNs) [9] or recurrent neural networks (RNNs) [10] with a transformer [11] encoder. They try to extract the correlation between context and question and find the most relevant part about the question in the passage. In recent works, bidirectional attention [12] has been proposed to calculate query-to-context attention scores and context-to-query attention scores simultaneously. In this paper, we adopt a bidirectional self-attention mechanism that considers multiple interactions between context and question to locate the most relevant part as the output.

In general, the attention mechanism focuses on the important points of current input and ignores other unimportant factors. In the transformer model, we have two attention concepts: self-attention and context attention. Self-attention means that the input sequence is exactly the same as the output sequence. The model calculates the attention score of itself to explore the most important part of the sequence. Context attention is also called encoder–decoder attention. It often appears in the translation model to calculate the attention score between each word in the encoder sequence and in the decoder sequence. In recent works, many improved attention mechanisms were invented by researchers, such as sparse self-attention [13], Graph neural network attention [14] for NLP tasks, and some others for computer vision tasks. The attention mechanism in NLU is widely used in translation, relational reasoning, abstract generation and MRC tasks.

2.2 MRC Datasets

Currently, MRC datasets can be divided into the following four main categories:

Multi-choice datasets have the same format as the multi-choice reading comprehension tasks found in the homework of junior or senior high school students, like RACE [15]. We use the accuracy score to measure the model's performance on multi-choice datasets.

Blank-filling datasets are as the name suggests. All of the passages in the dataset have many blanks that appear in various paragraphs. The machine needs to comprehend the context and tries to find a word to fill in the blank. BookTest [16] is a large-scale and typical blank-filling dataset. The evaluation metric of this dataset is also accuracy score.

Span-extraction datasets need machines to predict a continuous word sequence shown in the passage as the answer to the given question. We often use a start-position index and an end-position index as the range of the output answer. A series of famous span-extraction datasets were presented by investigators in recent years, such as SQuAD versions 1.1 and 2.0 [17-18], "AddSent-SQuAD", which adds distraction sentences on the normal SQuAD dataset published by [19], ReCoRD [20], NewsQA [21] and so on. The most common evaluation metrics are exact-match rate (EM) and F1-score(F1). In this work, we perform our experiments on six span-extraction datasets (SQuAD 1.1 and 2.0, AddSent-SQuAD, ReCoRD, Google Natural Questions [22] and Quoref [23]).

Free-answering datasets are those in which the answers are not successive in the passage or for which some mathematics or commonsense reasoning must be performed to determine the final answer. MS MARCO [24], DuReader [25] and DROP [26] all belong to this category. Table 2 is the detailed information of the above datasets.

Table 2. Detailed information of some Free-answering datasets

Dataset	Region&Source	Size	Published Time
MS MARCO	English MS user logs	8.8 M	2017
DuReader	Chinese Documents from search engine	1M	2017
DROP	English Manually created	96K	2019

Since all of the answers in these datasets are also provided by humans, we often use BLEU and ROUGE scores instead of the EM score to evaluate the model.

2.3 MRC Language Models

With the invention of the BERT pretrained language model by [27], an increasing number of researchers find that the transformer with a multi-head attention mechanism model structure can sharply increase the model's ability. Over a short period, RoBERTa [28], ALBERT [29], XLNET [30], ERNIE-1.0 and 2.0 [31-32] and a series of CLMs have been published. Among these excellent models, XLNET solves the discrephancy between train data and test data. It also has the ability of data generating. RoBERTa optimizes BERT's training strategy and mask mechanism. It also uses a massive scale of dataset, which is 10 times as big as BERT's train set, to finetune the language model. While ALBERT aims to train a small scale with high efficiency language model. These CLMs use a tremendous dataset size in pre-training, which makes them more robust and stable.

2.4 KB-MRC

When humans perform reading comprehension tasks, if we encounter a question that cannot be answered directly according to the given text, we will commonly use our background knowledge accumulated through our daily experiences. However, when a machine encounters a similar question, the attention score will become its only basis for evaluation.

Hence, MRC with a knowledge base came into being. When a knowledge graph between word entities has been established, a method for its vector representation is needed to let KBs take part in the concrete calculations. We have some different ways to embed our KBs into vectors, such as KG2E [33], Distmult [34] and so on. In this work, we adopt RotatE [35] to embed our knowledge from WordNet and ConceptNet and GloVe for setting the feature vector by NER result to enhance the self-comprehension ability of our model.

WordNet is a semantic web of English vocabulary that organizes an entity's noun expression, verb expression, synonyms, etc., to an independent net structure called synsets. many researchers, such as KAR [36], KT-NET [37] have picked WordNet as their knowledge base.

ConceptNet contain a large scale of commonsense knowledge closely connected with our daily life. For example, in ConceptNet, we have "golf – relatedTo – eagle" because of "eagle" is a concept of score in golf game. Commonsense knowledge helps the model to distinguish if two entities in the passage are under a same concept.

NELL [38] (Never-Ending Language Learning system) is to be able to develop means of answering questions posed by users in natural language with no human intervention in the process. One of the obvious of NELL knowledge base is its high update frequency, which helps machine to obtain the most recent concepts. Moreover, if the graph's scale of each entity is too large, we need to adopt an reduction algorithm like [39]. In this work.

3 Model

Our model is built of four main parts. The main function and principle of each part are as follows (Figure 3):

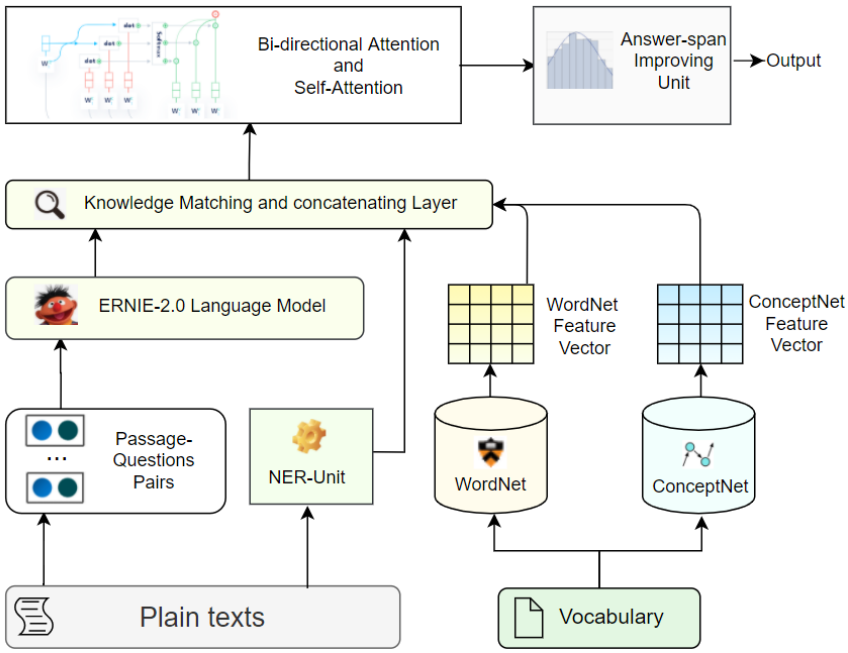

Figure 3. The overview of RKE-Reader

3.1 Embedding Layer

To encode plain texts to embedded vectors, we use ERNIE-2.0 to encode our question–answer pairs. ERNIE is similar to BERT in principle but is more effective in fine-tuning. It can mask a whole word or entity unit while training the model, which is different but better than BERT. ERNIE-2.0 uses the sum of Word, Sentence, Position and Task Embedding as its input. That is:

$$h_i = \sum_e t_i^e \tag{1}$$

where t_i is the ith token, e refers to the embeddings above, and h is the input hidden state. Since the passages and questions are input at the same time, we use a separator, "[SEP]", to distinguish them. Then, the encoded sequences are sent into the multilayer bidirectional transformer. We take the sequence output as the calculated result of this layer. The sequence output is computed by:

$$h_i = \sum_A \boldsymbol{W}_A^{l+1}(\sum_j Attn(h_{i,j}^l)) \tag{2}$$

$$o_i^{l+1} = \boldsymbol{W}_\alpha^{l+1} ACT(\boldsymbol{W}_\beta^{l+1} h_i^{l+1} + b_\beta^{l+1}) + b_\alpha^{l+1} \tag{3}$$

$$Attn_{ij} \propto \exp(k_j^T \boldsymbol{W} o_i^T \boldsymbol{W} o_j) \tag{4}$$

where A refers to the attention head, ACT is the activation function, $\mathbf{W}_A$,$\mathbf{W}_\alpha$, $\mathbf{W}_\beta$,b_α and b_β are all learnable parameters. We give them random values at the beginning of the training process. We use GELU-new [40] as the activation function. $Attn(h_{i,j}^l)$ means the attention score between token i and token j in the l th layer. We assume that we are in the last hidden layer for convenience.

3.2 Knowledge Matching and Concatenating Layer

This layer is the key of "RKE": it contains the output of the NER unit and two knowledge bases:

Named Entity Recognition (NER): Our input tokens need another pre-process step: NER. NER helps us distinguish kinds of named entities such as the names of persons, places, companies or time spans. In this work, we pick up the Stanford University NER system to help us recognize the names of organizations and persons that do not exist in our vocabulary set. The most significant advantage of NER is that it replaces some named entity tokens with a constant symbol instead of the <*UNK*> symbol before the encoding. The following is an example of how NER works in our model. (Table 3)

Table 3. Preprocessed results of the model with and without the NER system

Original text	Luke asks if Obi-Wan is possibly related to a hermit named Ben who lives several miles away in the Dune-Sea area, a vast terrain of sand and rocky canyons
NER preprocessed	[person0] asks if [person1] is possibly related to a hermit named [person2] who lives several miles away in the [misc0] area, a vast terrain of sand and rocky canyons
Normal preprocessed	[UNK] asks if [UNK] is possibly related to a hermit named [UNK] who lives several miles away in the [UNK] area, a vast terrain of sand and rocky canyons

The *MISC* in the table refers to the miscellaneous entities, which can be the name of a movie, video game, certain brand name, etc.

External Knowledge Bases: To convert relationships between entities to a matrix, a knowledge feature vector is needed before this operation. We follow RotatE as the knowledge graph embedding method: for each synset (triplets) extracted from WordNet, if we use (h, r, t) to represent a relation r between the head entity h and tail entity r, the score function is defined as:

$$f(s) = -\| h \odot r - t \|^2 \tag{5}$$

We give each token a 100-dimensional vector to represent knowledge embedding. The final shape of the WordNet feature matrix is 40,943 × 100.

The process of ConceptNet knowledge extraction is similar to that of WordNet. For a given entity, we use the API of ConceptNet and select at most the top 8 entities and their relations, ranked by the weight of each relation, as the knowledge base. We extract 142,103 triples containing 32,378 entities and 31 relations. The final shape of the ConceptNet feature matrix is 142,103 × 100. We concatenate all the knowledge vectors and the CLM embedding vector by using a similar method of [41].

3.3 Comprehension Layer

The input shape of encoded data in the previous layer is shown in Figure 4.

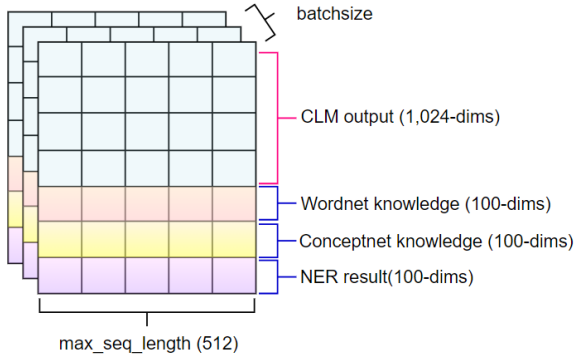

Figure 4. Matrix structure after knowledge injection

Each token in the passage consists of four parts: CLM output, WordNet feature vector, ConceptNet feature vector and NER output. The most important part of MRC is to extract the correlation between context and question properly. For this layer, we follow the trilinear self-attention to measure the similarity between the two input tokens:

$$s_{ij} = \boldsymbol{W}^T(\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_i \odot \boldsymbol{r}_j) \quad (6)$$

$$\boldsymbol{M}_s = (\boldsymbol{M}_{CLM}, \boldsymbol{M}_{WN}, \boldsymbol{M}_{CN}, \boldsymbol{M}_{NER}) \quad (7)$$

$$r_i = softmax(\boldsymbol{W}_0 \boldsymbol{M}_s + \boldsymbol{W}_1 \boldsymbol{M}_s^T + b) \quad (8)$$

where r_i, r_j are the rich-knowledge vectors generated from the previous layer, M is the assembled feature matrix, and W and b are the learnable parameters. We build our self-attention matrix by a softmax function:

$$\boldsymbol{A}_{ij} = \frac{\exp(s_{ij})}{\Sigma_j \exp(s_{ij})} \quad (9)$$

The attention weights between two entities are normalized under the condition of:

$$\sum_j A_{ij}^h = 1 \quad (10)$$

3.4 Output Layer

In this work, we focus on the span extraction tasks of MRC, where context C and question Q are given. The task is to extract a successive subsequence s = $\{t_i, t_{i+1}, ..., t_{i+m}\}$(1 ≤ i ≤ i+m)and i+m <*max_seq_length*, which means that we need to find the maximum probability of an entity and this token can be the start or end position of the answer span. We use a softmax function to calculate the probabilities:

$$p_s = \frac{\exp(\boldsymbol{W}_s^T o_i)}{\Sigma_j \exp(\boldsymbol{W}_s^T o_j)}, \; p_e = \frac{\exp(\boldsymbol{W}_e^T o_i)}{\Sigma_j \exp(\boldsymbol{W}_e^T o_j)} \quad (11)$$

$$Ł_{s,e} = \frac{1}{N}\sum_i^N (\log(p_c^s) + \log(p_c^e)) \quad (12)$$

where o is the attention output obtained from the previous layer, W is the weight matrix. p_s^c, p_e^c refer to the probability of obtaining the correct start and end position, and N is the total number of test cases.

3.5 Answer-span Improving System

In recent works, there are relatively less research about how to improve the given output from a MRC model. [42] introduced an online corrector in training process to let machine fix the incorrect answer span automatically. In this work, we present an offline MRC answer improvement system that has no need to do extra training tasks. It contains two sub components: answer extract unit and contextualize analyse unit. The operating principles of each unit are as follows:

answer extract unit: during the dataset pre-process work, we find that there are minor amount of answers are included in a single word. For example, we have a context like "*Fulham (1:1) Liverpool - Decordova-Reid's sweet finish cancelled out by Salah penalty*", and we want to find the score of "*team Liverpool*" in this match, in most of the MRC works, they often give us "1:1" as the output. However, that is not the exact answer. In this work, we use the word-piece tokenizer to split the single word into word pieces and extract the answer part. In this context, it will give the single number "1" as the answer, which is the correct one.

contextualize analyse unit: practically, we find that the reason some outputs miss the ground truth is that they often have some extra data or a subordinate clause after or before the right answer. For example, we have a ground truth, "*British Broadcasting Corporation*", but our output is "*British Broadcasting Corporation (BBC)*", which has an extra abbreviation that leads to an imperfect answer. We find that more than a half of prediction errors can be traced back to "extra or missing data". The confidence score often lower than 0.500000 in these wrong predictions. For the answers that the model generates correctly, the confidence score tends to be higher than 0.900000. We also classify all the errors into following 3 main categories (Figure 5) and give an example of each category (Table 4).

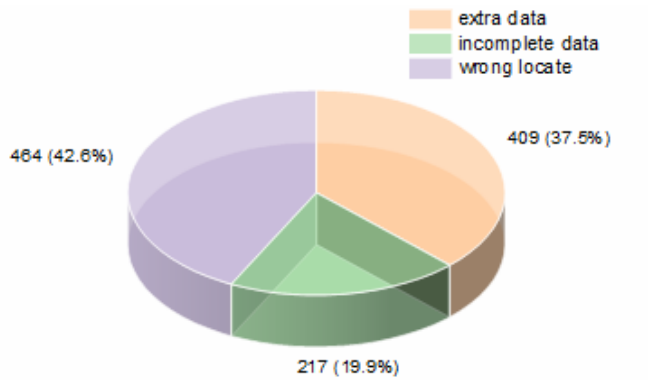

Figure 5. Main reasons of wrong predictions

Table 4. Examples of all type of errors in span extraction MRC tasks

	Type I: extra data
Question:	What was the highest price during the auction?
Output:	One person paid 80,000 euros ($98,000)
Ground truth:	80,000 euros
	Type II: incomplete data
Question:	What was held in the last Sunday?
Output:	Mona Lisa
Ground truth:	the Mona Lisa viewing
	Type III: wrong locate
Question:	What was the total turn volume of the auction?
Output:	It auctioned off a total of 24 items
Ground truth:	The auction raised approximately $2.9 million

We suppose that the reason model gives an "extra data" or an "incomplete data" answer is not it "really don't know the answer". In contrast, it "knows the answer, but fails to express it exactly". If we let our model give the "n-best" answers, we can find that a majority of answers have high degrees of similarity, like Table 18 and Table 19 in the applendix. The n-best answers are often belong to different parts of a whole sentence, and the sentence always contains the ground truth (except "wrong locate" errors). We let our model gives the n-best predictions to analyse if the ground truth of the question that model answered incorrectly is in the following n-1 best answers (Table 5):

Table 5. Theoretical maximum score with n-best predictions

n-size	theoretical max (EM-score)	theoretical Max (F1-score)
1	87.8	94.2
3	92.5	96.2
5	94.2	97.1

From the chart and tables above, we find that more than 94% of the question can be answered perfectly by use the answers from top 5 best predictions. So we reckon that there is a large potential in improving the accuracy of locating the exact answer. We present a solution that if we find that the maximum confidence score of the output is lower than 0.400000, we give each question 5 backup answers instead of just 1 "best" answer, and we use bigram perplexity (ppx) to measure the occurrence of an entity that exists in the corpus by the following equation:

$$ppx(s) = 2^{-\frac{1}{n}\Sigma \log(P(w_i))} \quad \textbf{(13)}$$

where n is the length of sequence and $P(w)$ means the probability of the word w existing under the condition of two words given ahead of w. Perplexity helps us to learn the common formation of an entity that exists in a document, and the result shows that it works well under the condition of stopword selections and punctuation selections. The corpus we use for calculating the perplexity value is BBC news dataset [43], fakenewsdataset [44] and HotPotQA [45]. The corpus contains 100,000 sentences in total. (Note: the content of the corpus has no intersection with our experiment datasets.) Generally, the perplexity value with a detailed description is often lower than the one that is short and simple. Moreover, it can also affect the output by the frequency of the sequence in the corpus, if our corpus contains more "*British Broadcasting Corporation*" than the latter one with *BBC* included, we will tend to choose the answer that does not have the suffix "*(BBC)*". The following table shows how this mechanism works (Table 6).

Table 6. Example of how we use perplexity to fix the answer

Predictions	Confidence/ Rank	log (ppx)/ Rank
construction of military roads	0.347304/#1	10.723868/#4
the construction of military roads	0.199904/#2	10.575973/#2
construction of military roads to the area by Braddock and Forbes	0.148401/#3	11.101646/#5
military roads to the area by Braddock and Forbes	0.085418/#4	10.671533/#3
construction of military roads to the area	0.081130/#5	**10.210134/#1**
Ground Truth: the construction of military roads		

As the table shows, prediction no.1 has one more stopword, "the", than prediction no. 2 does, however, it is not the exact right answer. If we consider this from the aspect of perplexity, we will choose the no. 2 prediction and arrive at the ground truth. The reason that *ppx*(2nd-answer)<*ppx*(1st-answer) is that there is more use of "*the construction of sth*" than the one without the in the corpus. If there are no answers with a confidence score>0.400000, we will consider both the confidence score and perplexity value. We first sort them and choose the answer that performs well on both of two aspects, that is, *min*(Rank(conf) + Rank(ppx)). If we obtain the same calculation result on two or more answers, we choose the one with low perplexity. The final result shows that using the perplexity value to help the model determine the uncertain answer can affect the final F1 score by approximately -0.2 to 0.4.

4 Experiments

4.1 Datasets

In this work, we mainly focus on the span extraction task of MRC. We evaluate our model and kinds of baselines on following datasets.

SQuAD1.1 and SQuAD2.0 are widely used in evaluating MRC models. SQuAD1.1 contains more than 100,000 questions from 500+articles. All of the answers can be found continuously in the passage. SQuAD2.0 not only includes the questions in previous version, but also adds more than 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones.

Addsent-SQuAD is a revision of SQuAD dataset. It has more distracting sentences on both questions and passages. Most of these extra sentences are related to the answer spans.

ReCoRD is a large-scale reading comprehension dataset which requires the ability of commonsense reasoning, which means that some of the answers are existing in a different way in the passage. It needs the

machine to use some external knowledge to analyse the answer.

Google Natural Questions dataset is a banchmark for QA research provide by Google. The corpus of this dataset consists of more than 307,000 articles from the entire Wikipedia. The full size of the dataset is larger than 42GB. We use the simplified version of the dataset, select all passages that has answers and convert them into SQuAD format and train the dataset with the same steps like SQuAD1.1.

Quoref is a MRC dataset that aims to test the ability of understanding long and complex sentences. In Quoref, the questions are derived from paragraphs taken from a diverse set of English Wikipedia articles and are collected using an annotation process that deals with the aforementioned issues in different ways.

4.2 Baselines

We compare our RKE-Reader with baselines as follows under the same hardware condition on every datasets:

BERT: The official evaluation of the SQuAD 1.1 and 2.0 datasets is available on the website. We have also made it runnable in the ReCoRD dataset to obtain more results. We compare two kinds of BERT models: BERT-base and BERT-large, which are mainly different in the number of parameters.

KT-NET: is a KB-MRC model that also has four main components. It uses BERT as the CLM with two knowledge bases—WordNet and NELL —which reached the top spot on the SQuAD Leaderboard when it was published.

RKE-Reader (without KBs): We remove the KB components of our model and try to perform the same task. Thus, we can intuitively discover the ability of the KB-buffered layer in our model (Ablation study). We have performed other baselines that are not listed above on the datasets.

4.3 Implementation Details

The training process is executed on a Nvidia-Tesla V100 (32 GB video memory) GPU machine with an Intel Xeon Gold 6271C CPU with 12 cores and 24 threads. The RAM size of our machine is 32GB. Our model is implemented in PaddlePaddle-GPU version 1.7.0. We use GELU-new as the activation function and Adam as the optimizer in the fitting stage. All learnable parameters are randomly initialized. The detailed configuration of our experiments is presented in Appendix I (Table 16).

Inspired by the work of [46], we adopt a two-stage fine-tuning strategy to train our model. We split our training process into two stages: warm-up stage and fitting stage. In warm-up stage, we freeze the parameters from contextualized language models, and we also give a relative large batchsize and a high learning rate maximum value to the model. While in fitting stage, we unfreeze the CLM parameters and we use the Adam optimizer with a decline learning rate and a smaller batch size. Detail settings are presented in Table 7.

Table 7. Global settings of our experiments

	warm-up stage	fitting stage
batchsize	32	16
learning-rate	Increase Linear	Decrease Adam
Attention dropout-rate	0.1	0.05
Hidden dropout-rate	0.1	0.05

We use RotatE as the method of building our knowledge graph embeddings. We train the RotatE model with the WN18 dataset, 1,024 as the batch size, 1e-5 as the learning rate and 128 as the negative sample size to generate 100-dimension knowledge embedding. The training set size of WN18 is 141,442, and the validation and test sets are all 5,000 triplets. The results show that the scores of Hit@1, Hit@3 and Hit@10 on RotatE are 0.941, 0.951, and 0.959, respectively.

4.4 Evaluation and Results

We use EM and F1-score, to evaluate our model and other baselines. We find that our model is not only better in accuracy but also slightly faster in the training process. Under the environment of our hardware, our model can finish the entire training task within 10 hours, while the control group generally spends more than 12 hours. The results of our experiments (Table 8 to Table 13) show that knowledge bases play an important role in MRC.

Table 8. Experiments performed on SQuAD2.0 dataset

model & method	ALL		HasAns		NoAns	
	EM	F1	EM	F1	EM	F1
BERT-large (2018)	80.4	82.0	81.3	84.6	79.5	79.5
BIDAF (2018)	59.3	62.3	–	–	–	–
BIDAF+CLM (2018)	63.4	66.2	–	–	–	–
RMR+ELMo +Verifier (2019)	71.7	74.2	–	–	–	–
KT-NET (2019)	81.1	82.9	81.9	85.5	80.3	80.3
SpanBERT [47] (2020)	85.5	88.6	84.5	90.7	86.5	86.5
RKE-Reader	86.5	88.9	87.0	91.7	86.1	86.1
Human Performance	86.9	89.5	–	–	–	–

Table 9. Experiments performed on Google Natural Questions dataset

model & method	EM-score	F1-score
XLNET-single model (2019)	61.88	70.51
RoBERTa-essemble (2019)	79.66	86.13
CorefRoberta-large (2020)	75.80	82.81
RoBERTa-MT (2019)	72.61	80.68
KT-NET (2019)	66.03	77.19
RKE-Reader	**81.51**	**87.24**
Human Performance	93.00	–

Table 10. Experiments performed on SQuAD1.1 dataset

model&method	EM-score	F1-score
BERT-base (2018)	80.8	88.5
BERT-large (2018)	84.1	90.9
BIDAF (2018)	80.0	--
QA-NET (2018)	83.5	86.9
R-NET (2018)	82.1	88.2
KT-NET (2019)	85.1	91.7
KAR (2019)	76.1	84.9
Hierarchical MRC [48] (2020)	73.9	82.4
IMM [49] (2020)	–	89.4
RKE-Reader	**87.8**	**94.2**
Human Performance	82.3	91.2

Table 11. Experiments performed on ReCoRD dataset

model&method	EM-score	F1-score
BERT-large (2018)	56.4	59.1
XLNET-large (2018)	61.1	63.8
SKG-BERT [50] (2019)	70.9	72.2
KT-NET (2019)	71.6	73.6
RKE-Reader	**76.9**	**79.0**
Human Performance	91.3	91.6

Table 12. Experiments performed on QuoREF dataset

model&method	EM-score	F1-score
BERT-large (2018)	64.9	74.2
XLNET-large (2019)	65.0	76.6
DecaProp (2018)	53.1	65.3
SpanBERT (2020)	–	73.6
IMM (2020)	–	79.2
RKE-Reader	**66.9**	**80.1**

Table 13. Experiments performed on Addsent-SQuAD dataset

model & method	EM-score (Addsent)	F1-score (Addsent)	EM-score (Add1sent)	F1-score (Add1sent)
BERT-large	54.30	58.50	60.86	66.92
XLNET-large	57.31	61.44	63.10	69.01
QANet (2018)	-	45.20	-	55.70
KAR (2019)	-	60.10	-	72.30
Hierarchical MRC (2020)	58.30	65.50	65.20	72.50
KT-NET (2019)	62.40	68.18	70.90	75.43
RKE-Reader	**75.40**	**81.21**	**80.22**	**86.21**

From all of our experiments, results of using KBs outperform those without KBs especially in English datasets. In dataset SQuAD 2.0, RKE-Reader has an obvious advantage in the test cases that have an answer. In version 1.1, RKE-Reader performs nearly 6% better than human performance in EM score and 3% better in F1 score, which is an obvious evidence that external KBs can make our model more robust and reliable. In addition, the results of experiments on other datasets are also close to or slightly lower than the best model on their leaderboards.

4.5 Ablation Study

We also conduct a series of ablation study to discover the effect of each component of our model (Table 14), the result shows that KBs and answer-improving system do both make positive effects on comprehension works.

Table 14. Ablation study on each component of our model

model component	EM-score	F1-score
ERNIE-2.0 (single model)	85.5	91.5
RKE-Reader (KB disabled)	86.9	92.9
RKE-Reader (KB enabled)	87.6	94.0
RKE-Reader (KB, answer improve enabled)	**87.8**	**94.2**

Last but not least, Table 15 shows that it is incorrect to think that the more dimensions the KB possesses, the better performance that will be reflected in the model. If we deploy an oversized dimension on embedding our knowledge bases, it will result in a poor performance instead:

Table 15. Different performance on different scale of KBs on dataset SQuAD 1.1

model & method	EM-score	F1-score
RKE-Reader (No-KBs)	86.5	93.1
RKE-Reader (100-D for each KB)	**87.8**	**94.2**
RKE-Reader (200-D for each KB)	87.7	93.9
RKE-Reader (300-D for each KB)	86.5	92.7

5 Conclusions and Future Works

In this paper, we present the RKE-Reader, a rich knowledge enhanced model for MRC. This work is the first attempt to add multiple knowledge bases to the ERNIE CLM to finish MRC tasks, and it's also the first to present an offline answer-improving system to make the final answer span better. The results show that the RKE-Reader outperforms most of the published models in some classic datasets of span-extraction MRC tasks. In the future, we plan to upgrade our KB module, make knowledge integration to merge individual knowledge bases into a large knowledge unit, we believe that it can greatly improve the performance. Then, we plan to upgrade a new CLM model, like ALBERT, modify some parts of our model to make it capable of performing transfer learning, which requires only one model checkpoint to perform all span-extraction tasks under same language. Further, we consider to make our model more flexible so that it can handle free-answering tasks that require a model

not only to find and locate but also to generate and summarize.

References

[1] H. Gao, L. Kuang, Y. Yin, B. Guo, K. Dou, Mining Consuming Behaviors with Temporal Evolution for Personalized Recommendation in Mobile Marketing Apps, *Mobile Networks and Applications (MONET)*, Vol. 25, No. 4, pp. 1233-1248, August, 2020.

[2] X. Yang, S. Zhou, M. Cao, An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews, *Mobile Networks and Applications (MONET)*, Vol. 25, No. 2, pp. 376-390, April, 2020

[3] G. A. Miller, WORDNET: a lexical database for English, *Speech and Natural Language*, Harriman, New York, USA, 1992, pp. 483.

[4] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 2017, pp. 4444-4451.

[5] H. Gao, W. Huang, Y. Duan, The Cloud-edge-based Dynamic Reconfiguration to Service Workflow for Mobile Ecommerce Environments: A QoS Prediction Perspective, *ACM Transactions on Internet Technology (TOIT)*, Vol. 21, No. 1, pp. 1-23, February, 2021.

[6] X. Ma, H. Gao, H. Xu, M. Bian, An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing, *EURASIP Journal on Wireless Communications and Networking,* Vol. 2019, Article No. 249, November, 2019.

[7] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *CoRR*, Vol. abs/1301.3781, September, 2013.

[8] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532-1543.

[9] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, Nevada, United States, 2012, pp. 1106-1114.

[10] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, *CoRR*, Vol. abs/1409.2329, September, 2014.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *Annual Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5998-6008.

[12] M. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi, Bidirectional attention flow for machine comprehension, *CoRR*, Vol. abs/1611.01603, November, 2016.

[13] B. Cui, Y. Li, M. Chen, Z. Zhang, Fine-tune BERT with sparse self-attention mechanism, in Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019, pp. 3546-3551.

[14] S. Zhang, L. Xie, Improving attention mechanism in graph neural networks via cardinality preservation, *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, online event, 2020, pp. 1395-1402.

[15] G. Lai, Q. Xie, H. Liu, Y. Yang, E. Hovy, RACE: Large-scale ReAding comprehension dataset from examinations, *Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September, 2017, pp. 785-794.

[16] O. Bajgar, R. Kadlec, J. Kleindienst, Embracing data abundance: Booktest dataset for reading comprehension, *CoRR*, Vol. abs/1610.00956, October, 2016.

[17] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100, 000+ questions for machine comprehension of text, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas, USA, 2016, pp. 2383-2392.

[18] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for squad, *CoRR*, Vol. abs/1806. 03822, June, 2018.

[19] R. Jia, P. Liang, Adversarial examples for evaluating reading comprehension systems, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark, 2017, pp. 2021-2031.

[20] S. Zhang, X. Liu, J. Liu, J. Gao, K. Duh, B. V. Durme, Record: Bridging the gap between human and machine commonsense reading comprehension, *CoRR*, Vol. abs/1810. 12885, October, 2018.

[21] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, K. Suleman, Newsqa: A machine comprehension dataset, *CoRR*, Vol. abs/1611.09830, November, 2016.

[22] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, S. Petrov, Natural questions: a benchmark for question answering research, *Transactions of the Association for Computational Linguistics*, Vol. 7, pp. 453-466, July, 2019.

[23] P. Dasigi, N. F. Liu, A. Marasovi´c, N. A. Smith, M. Gardner, Quoref: A reading comprehension dataset with questions requiring coreferential reasoning, *CORR*, Vol. abs/1908. 05803, September, 2019.

[24] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, MS MARCO: A human generated machine reading comprehension dataset, *the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016, pp. 1-10.

[25] W. He, K. Liu, J. Liu, Y. Lyu, S. Zhao, X. Xiao, Y. Liu, Y. Wang, H. Wu, Q. She, X. Liu, T. Wu, H. Wang, DuReader: a Chinese machine reading comprehension dataset from real-world applications, *the Workshop on Machine Reading for*

Question Answering, Melbourne, Australia, 2018, pp. 37-46.

[26] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, M. Gardner, DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs, *CoRR*, Vol. abs/1903.00161, April, 2019.

[27] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *CoRR*, vol. abs/1810.04805, October, 2018.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, *CoRR*, vol. abs/1907.11692, July, 2019.

[29] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, *8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020, pp. 1-17.

[30] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, *Annual Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 5754-5764.

[31] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: enhanced language representation with informative entities, *Conference of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019, pp. 1441-1451.

[32] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, ERNIE 2.0: A continual pre-training framework for language understanding, *CoRR*, Vol. abs/1907.12412, November, 2019.

[33] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, *Twenty-Eighth AAAI Conference on Artificial Intelligence*, Qu´ebec City, Qu´ebec, Canada, 2014, pp. 1112-1119.

[34] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, *3rd International Conference on Learning Representations* (*ICLR)*, San Diego, CA, USA, 2015, pp. 1-13.

[35] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, *International Conference on Learning Representations* (*ICLR)*, New Orleans, LA, USA, 2019, pp. 1-18.

[36] C. Wang, H. Jiang, Explicit utilization of general knowledge in machine reading comprehension, *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019, pp. 2263-2272.

[37] A. Yang, Q. Wang, J. Liu, K. Liu, Y. Lyu, H. Wu, Q. She, S. Li, Enhancing pre-trained language representations with rich knowledge for machine reading comprehension, *57th Conference of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019, pp. 2346-2357.

[38] T. Mitchell, E. Fredkin, Never-ending language learning, *IEEE International Conference on Big Data*, Washington, DC, USA, 2014, pp. 1-1.

[39] X. Zhang, H. Yao, Z. Y. Lv, D. Q. Miao, Class-specific information measures and attribute reducts for hierarchy and systematicness, *Information Sciences*, Vol. 563, pp. 196-225, July, 2021.

[40] D. Hendrycks, K. Gimpel, Bridging nonlinearities and stochastic regularizers with gaussian error linear units, *CoRR*, Vol. abs/1606.08415, June, 2016.

[41] Y. Yin, Z. Cao, Y. Xu, H. Gao, R. Li, Z. Mai, QoS Prediction for Service Recommendation with Features Learning in Mobile Edge Computing Environment, *IEEE Transactions on Cognitive Communications and Networking*, Vol. 6, No. 4, pp. 1136-1145, September, 2020.

[42] R. G. Reddy, M. A. Sultan, E. S. Kayi, R. Zhang, V. Castelli, A. Sil, Answer span correction in machine reading comprehension, *Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*, Online Event, 2020, pp. 2496-2501.

[43] D. Greene, P. Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, *23rd International Conference on Machine learning (ICML)*, Pittsburgh, Pennsylvania, USA, 2006, pp. 377-384.

[44] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media, *CoRR*, Vol.abs/1809.01286, September, 2018.

[45] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, C. D. Manning, Hotpotqa: A dataset for diverse, explainable multi-hop question answering, *Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 2369-2380.

[46] K. Sun, D. Yu, J. Chen, D. Yu, C. Cardie, Improving machine reading comprehension with contextualized commonsense knowledge, *CoRR*, Vol. abs/2009.05831, October, 2020.

[47] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, Spanbert: Improving pre-training by representing and predicting spans, *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 64-77, March, 2020.

[48] Z. Wu, H. Xu, Improving the robustness of machine reading comprehension model with hierarchical knowledge and auxiliary unanswerability prediction, *Knowledge-Based Systems*, Vol. 203, Article No. 106075, September, 2020.

[49] X. Liu, K. Liu, X. Li, J. Su, Y. Ge, B. Wang, J. Luo, An iterative multi-source mutual knowledge transfer framework for machine reading comprehension, *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, online event, 2021, pp. 3794-3800.

[50] D. Qiu, Y. Zhang, X. Feng, X. Liao, W. Jiang, Y. Lyu, K. Liu, J. Zhao, Machine reading comprehension using structural knowledge graph-aware network, *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 5895-5900.

Biographies

Feifei Xu is an associate professor of the School of computer science and technology, Shanghai University of Electric Power, China. She holds a Ph.D degree from Tongji University in Pattern Recognition and Intelligence System in Electronic and Information Engineering. She has been engaged in the research of natural language processing, rough set theory and machine learning.

Wenkai Zhang received the B.S. degree in software engineering from Fujian University of Technology, Fujian, China, in 2019. He is currently pursuing the M.S. degree in computer technology at Shanghai University of Electric Power. His research interest includes the natural language understanding and knowledge graph.

Haizhou Du received his Ph.D degree in computer science and technology from Tongji University, Shanghai, China. His research interests include Machine Learning, Data Management, and Distributed System.

Shanlin Zhou received the B.S. degree in information and computing science from Hebei Finance University, Hebei, China, in 2019. She is currently pursuing the M.S. degree in computer technology at Shanghai University of Electric Power. Her research interest includes the neural language generation and knowledge graph.

Applendix

A.1 Experiment Configurations

The detail configuration on each dataset are as follows (Table 16).

Table 16. Detail configurations on each datasets

Dataset	Learning-rate	Epochs
SQuAD-1.1 (2016)	4e-5	3
SQuAD-2.0 (2018)	4e-5	3
SQuAD-Addsent (2017)	4e-5	3
Google Natural Questions (2019)	4e-5	2
Quoref (2019)	3e-5	6
ReCoRD (2018)	3e-5	3

A.2 Case Study

As is shown in Table 17, when RKE-Reader meets an input, it'll start to obtain the CLM output and search all knowledge bases simultaneously. For example, in this table, RKE-Reader finds three relations between the first sentence of passage and the question, that's an important step in finding the answer. From the results of Figure 6, Figure 7, Table 18 and Table 19, we can discover that RKE-Reader gives a high confidence score of 0.988 on correct answer "35", while seq2seq model misses the ground truth and gives an imperfect answer as its second backup answer.

Table 17. Training example from SQuAD1.1 dataset

Text	On 30 July 1891, at the **age** of 35, **Tesla** became a naturalized citizen of the **United States**, and established his South Fifth Avenue laboratory,and later another at 46 E. Houston Street, in New York.
Question	How **old** was **Tesla** when he became a **US** citizen?
KB-info	WordNet:US–United States ConceptNet:old–age NER:Tesla–person
Seq2seq Output	On 30 July 1891
RKE-Reader Output	**35**
Ground Truth	**35**

Table 18. Top 4 predictions generated by RKE-Reader

Answers	Confidence
35	0.988375
at the age of 35	0.009102
the age of 35	0.000603
On 30 July 1891, at the age of 35	0.000124

Table 19. Top 4 predictions generated by Seq2seq model

Answers	Confidence
On 30 July 1891	0.349196
On 30 July 1891, at the age of 35	0.297857
30 July 1891	0.170909
the age of 35	0.103167

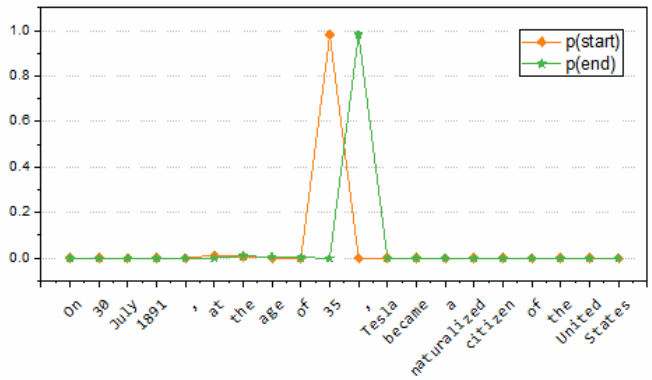

Figure 6. Prediction generated by RKE-Reader

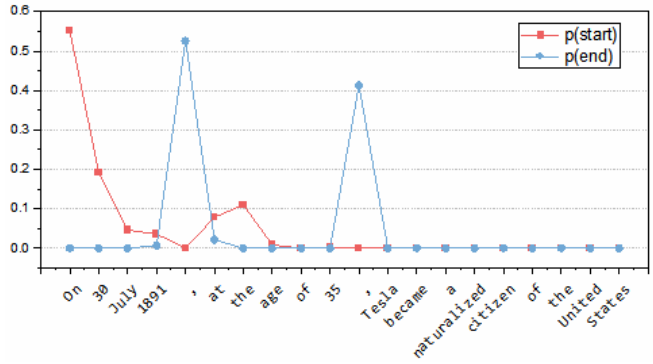

Figure 7. Prediction generated by Seq2seq model