Towards Knowledge Enhanced Language Model for Machine Reading Comprehension

PEIZHU GONG, JIN LIU, (Member, IEEE), YIHE YANG, AND HUIHUA HE

1 College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
2 College of Education, Shanghai Normal University, Shanghai 200234, China

Corresponding author: Huihua He (hehuihua@shnu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61872231 and Grant 61701297.

ABSTRACT Machine reading comprehension is a crucial and challenging task in natural language processing (NLP). Recently, knowledge graph (KG) embedding has gained massive attention as it can effectively provide side information for downstream tasks. However, most previous knowledge-based models do not take into account the structural characteristics of the triples in KGs, and only convert them into vector representations for direct accumulation, leading to deficiencies in knowledge extraction and knowledge fusion. In order to alleviate this problem, we propose a novel deep model KCF-NET, which incorporates knowledge graph representations with context as the basis for predicting answers by leveraging capsule network to encode the intrinsic spatial relationship in triples of KG. In KCF-NET, we fine-tune BERT, a highly performance contextual language representation model, to capture complex linguistic phenomena. Besides, a novel fusion structure based on multi-head attention mechanism is designed to balance the weight of knowledge and context. To evaluate the knowledge expression and reading comprehension ability of our model, we conducted extensive experiments on multiple public datasets such as WN11, FB13, SemEval-2010 Task 8 and SQuAD. Experimental results show that KCF-NET achieves state-of-the-art results in both link prediction and MRC tasks with negligible parameter increase compared to BERT-Base, and gets competitive results in triple classification task with significantly reduced model size.

INDEX TERMS Machine reading comprehension, knowledge graph embedding, BERT, capsule network.

I. INTRODUCTION

With the rapid development of deep learning and increasing availability of datasets (e.g., SQuAD, RACE) [1], [2], machine reading comprehension (MRC) has achieved remarkable advancements in the last few years. A common approach is to use recurrent neural networks (RNNs) to process the given text and the question in order to predict or generate the answers. RNNs, such as Long Short-Term Memory (LSTM) units [3], are effective to encode the context semantics and capture the long-term dependency. With the recurrent connection, the word in the given passage or question also contains the information of its neighbors. Besides, attention mechanism [4] is widely used with RNNs so as to promote interaction between the question and the given passage. The attention mechanism imitates the process of people’s repeated thinking while doing reading comprehension, which captures the semantic relevance of words in questions and passages. Therefore, many studies have used attention-based RNNs [5] as backbone and achieved promising results. However, existing methods can only obtain the semantic information of a given text, lacking common knowledge, which will limit the understanding of words or cause deviations.

In the process of human reading comprehension, people may utilize world knowledge when the question cannot be answered simply by contextual reasoning. The external knowledge plays such an important role that it is considered to be the biggest gap between MRC and human reading comprehension. Knowledge graph (KG) [6] raised by Google in 2012, uses an effective way to organize and manage massive amount of information, which can provide rich structured knowledge facts for better language understanding. Recently, introducing structured knowledge in KG into MRC has become a new trend and several studies centered on knowledge-based machine reading comprehension (KBMRRC) [7] have been proposed. Zhang et al. [8] utilized both large-scale textual corpora and KGs to train an enhanced
language representation model (ERNIE), which achieves significant improvements on various knowledge-driven tasks. Liu et al. [9] proposed a knowledge-enable language representation model (K-BERT) with knowledge graph (KGs), which introduces soft-position and visible matrix to avoid knowledge noise to a large extent. In the former, the algorithm TransE which is used for extracting and encoding structured knowledge, has a poor ability to obtain semantic information, while in the latter, the sentence tree to limit the impact of knowledge greatly increase the complexity of information fusion.

In this paper, we take advantage of structural characteristics of the triples in KGs and incorporates knowledge graph representations into MRC to overcome the shortcomings of the above methods. There are two challenges lies in the road of this knowledge integration: (1) Structured Knowledge Encoding: although the structure of triples (head entity, relation, tail entity) in KGs can well sort out facts, it cannot be directly used for downstream tasks. So how to effectively extract and encode its related informative facts in KGs for language representation models is an important problem (2) Knowledge Noise Issue: the intervention of irrelevant knowledge or synonyms may be misleading and divert the sentence from its correct meaning, hence how to design a reasonable fusion mechanism is another challenge.

To tackle these challenges, we propose a two-stage model KCF-NET. For knowledge extraction, we first convert the triples in KG into vector representations, and design CapsBERT, a novel module based on CapsNet to mine the global relationship between entities. The global semantic information got from CapsBERT is more comprehensive and richer, compared with that in previous work which only retrieves information of related entities. For knowledge fusion, we propose a new fusion mechanism, adopting the idea of gating, which can balance the weight distribution of the meaning of the original text and external knowledge to limit the impact of external knowledge in a simple form.

In summary, main contributions of this paper are as follows:

- To enhance knowledge expression in relational triples, we design a novel knowledge extraction module consisting of BERT and CapsNet which captures the horizontal relationship and vertical relationship of the triple matrix, containing features between and in the triples.
- To balance the weight of knowledge and context, a novel knowledge fusion structure based on multi-head attention mechanism is proposed, which utilizes gating units to preserve the original information and avoid knowledge noise.
- In order to accelerate the convergence of CapsNet and avoid overfitting, we design an initialization mechanism based on the similarity between capsules to optimize the dynamic routing algorithm.

The remaining of this article is structured as follows. The related works are discussed in section 2. Section 3 presents KCF-NET and its details. Finally, we evaluate the experimental results with datasets in section 4 and draw a conclusion in section 5.

II. RELATED WORK

A. MACHINE READING COMPREHENSION

Machine reading comprehension is a task to measure the extent to which the machine understands natural language by reading a document and answer questions. Compared with traditional tasks such as word segmentation, named entity recognition and syntactic analysis, MRC usually needs longer chapters and deeper semantic information, which requires comprehensive use of text representation, retrieval, reference resolution and reasoning. In order to accurately predict the answer, many efforts have devoted to extract the correlation between the context and question. Hermann et al. [10] proposed an attention-based neural network that calculates the similarity between the context and question. Weston et al. [11] introduced an end-to-end memory network, which can explicitly store long-term memories and also have an easy access to reading memories. In this way, the model can understand the context and question more deeply by multiple turns of interaction. Dhingra et al. [12] proposed GA reader, which utilize gate mechanism to decide the extent to which question information affects context words when updating the context representations.

However, due to the limited knowledge contained in the context, there is still a long way to go before the machine truly understands text. Knowledge-based machine reading comprehension (KBmrc), as a new trend, makes MRC tasks much closer to real-world application. KBMRC differs from MRC mainly in inputs. In addition to sequence of the context and question, related knowledge extracted from knowledge base is necessary in KBMRC. Long et al. [13] propose a new task, which provides additional entity description extracted from knowledge base as external knowledge to help entity prediction. Miller et al. [14] utilize Key-Value Memory Networks to find out relevant external knowledge and store it in memory slots as key-value pairs. Then keys are used to match with the query to generate relevant knowledge representations. Wang & Jiang [15] propose a data enrichment method with semantic relations in WordNet. For each word in the context and question, they try to find out the positions of passage words, which are regarded as explicit knowledge to assist answer prediction. Pan et al. [16] propose a novel encoder-decoder supplementary architecture which transfers knowledge learned from machine comprehension to the sequence-to-sequence tasks to deepen the understanding of the text. In conclusion, KBMRC has demonstrated its advantages in exploiting knowledge for question answering, which is no longer restricted to the given context.

B. KNOWLEDGE REPRESENTATION

Knowledge graph, such as WordNet, Freebase, and YAGO [17]–[19], composed by millions of facts about real-world entities and relations in the form of triples, are valuable
resources for various NLP tasks. However, the structure of triples cannot be directly transformed into semantic expression. To this end, knowledge graph embedding models have been proposed to learn vector representations for entities and relations in KGs. Bordes et al. [20] proposed TransE, inspired by the interesting translation-invariant phenomenon of lexical semantics and syntactic relations in the word vector space. TransE regards the relationship in the knowledge base as some kind of translation vector between entities. However, because of the simple structure, it is stretched when dealing with complex relationships. To break through the limitations in dealing with the complex relationships of 1-N, N-1, and N-N, Wang et al. [21] modeled relational triples on hyperplanes and Lin et al. [22] built entity and relation embeddings in separate spaces. More fine-grained models were proposed one after another such as TransD, TransG, and KG2E [23]–[25]. In addition, Ji et al. [26] proposed TranSparse to address the issue of multiple relation semantics. Recently, deep learning approaches have demonstrated breakthrough performance. Nguyen et al. [27] proposed ConvKB, a convolutional neural network (CNN)-based model for KG completion. Yao et al. [28] employed BERT for knowledge graph representation and achieved state-of-the-art results in triple classification, link prediction and relation prediction. Zhong et al. [29] treated triples in knowledge graphs as textual sequences and proposed a novel network based on Transformer. However, the method mentioned above does not take the characteristics of structured knowledge in triples into account. In this paper, CapsNet is combined with BERT to enhance knowledge expression.

C. KNOWLEDGE FUSION

In recent years, enhancing pre-trained language model [30] with external knowledge for downstream tasks has become a new trend. Meanwhile, how to combine knowledge representation with the characteristics of specific task has become an immense challenge. Yang et al. [31] took advantage of external knowledge bases to improve recurrent neural networks for MRC. To effectively integrate background knowledge with information from currently processed text, Yang et al. employed an attention mechanism with a sentinel to adaptively decide whether to attend to external knowledge and which information is useful. Zhang et al. [32] utilized both large-scale textual corpora and KGs to train an enhanced language representation model named ERFIE. ERFIE consists of two stacked modules: T-encoder and K-encoder. The former is responsible for capturing basic lexical and syntactic information from the input tokens, and the latter use TransE to obtain extra token-oriented knowledge information and integrate it into textual information through GELU function. Mihaylov et al. [33] proposed KN-Reader, encoding external commonsense knowledge as a key-value memory. By calculating the attention between the knowledge and context representation, KN-Reader selects the candidate that has the highest attention score as answer. Wang & Jiang [34] designed two types of attention mechanisms. The first one, mutual attention, is aimed at fusing the question representations into the passage so as to obtain the question-aware passage representations; the second one, self-attention, is aimed at fusing the question-aware passage representations into themselves so as to obtain the final passage representations. Yang et al. [35] proposed KT-NET, which employs an attention mechanism to adaptively select desired knowledge from KBs, and then fuses selected knowledge with BERT to enable context- and knowledge-aware predictions. However, most previous methods retrieve the relevant knowledge base before encoding it into the MRC, which can only refer to KB locally. In this paper, we first learn the overall knowledge graph embeddings, and then integrate the knowledge with the context. We design a novel fusion structure that can control the influence of knowledge on reading comprehension and is easier to train.

III. METHOD

In this work we consider the span extraction task in MRC. Given a passage with m tokens $P = \{P_1, P_2, \ldots, P_m\}$ and a question with n tokens $Q = \{Q_1, Q_2, \ldots, Q_n\}$, our goal is to predict an answer A, which is constrained as a continuous span in the passage, donated as $A = \{P_s, \ldots, P_e\}$, where $1 \leq s \leq e \leq m$.

As depicted in Fig. 1, KCF-NET can be divided into embedding layer, fusion layer and output layer. The previous work in the literature only converts the triples in KG into a vector representation. When related entities are mentioned in the article, the semantic vector is used as external knowledge for direct accumulation, leading to deficiencies in both knowledge extraction and knowledge fusion. To take full advantage of facts in KGs which contains rich language patterns, we propose a novel module CapsBERT as a pre-processing step, which fine-tunes pre-trained BERT [36] and utilizes a capsule neural network (CapsNet) [37] for knowledge graph representation. BERT is a state-of-the-art contextual language representation model built on a multi-layer bidirectional Transformer encoder, which is able to capture complex linguistic phenomena. Sabour et al. [38] first introduced CapsNet that employs vector neurons to capture entities in images. Extracted features will be sent from capsules in a layer to those in the next layer by a powerful dynamic routing mechanism, which could encode the intrinsic spatial relationship of triples between the part and the whole. Different from the traditional application of CapsNet, we use capsules to model the entries at the same dimension in the entity and relation embeddings. Thus, each capsule can encode more characteristics in the embedding triple to enhance its knowledge expression.

Given a question and passage, KCF-NET first splits the input into sentences. Then it employs, in turn, (1) a BERT embedding layer containing two encoding methods which compute context-aware representation and knowledge graph representation respectively for the reading text; (2) a fusion layer to integrate context information with external knowl-
edge, so as to enable rich interactions between them; (3) an output layer to predict the final answer.

A. EMBEDDING LAYER

The embedding layer is designed as two branches, using BERT encoders to model reading text and facts in KGs respectively, denoted as context-BERT and knowledge-BERT.

As depicted in Fig. 2, context-BERT takes passage $P$ and question $Q$ as input, and computes a context-aware representation for each token. Specifically, given a passage with $m$ tokens $P = \{P_1, P_2, \ldots, P_m\}$ and a question with $n$ tokens $Q = \{Q_1, Q_2, \ldots, Q_n\}$, we first pack them into a sequence of length $m + n + 3$ i.e.,

$$\pi = [< \text{CLS}>, Q, < \text{SEP}>, P, < \text{SEP}>]$$  \hspace{1cm} (1)

where $<\text{SEP}>$ is the token separating $Q$ and $P$, and $<\text{CLS}>$ is the token for classification. For each token $\pi_i$ in $\pi$, we construct its input representation as:

$$H_i^0 = t_i + s_i + p_i$$  \hspace{1cm} (2)

where $t$, $s$, $p$ are the token, segment and position embeddings respectively, $i$ represents the $i$-th sample in the training batch, $0$ represents the input of the BERT layer. Tokens in $Q$ share a same segment embedding $\delta_p$ and tokens in $P$ a same segment embedding $\delta_q$. Such input representations are then fed into $L$ successive Transformer encoder blocks, i.e.,

$$H_i^l = \text{Transformer}(H_i^{l-1}), \hspace{1cm} l = 1, 2, \ldots, L$$ \hspace{1cm} (3)

The final hidden states $o^{\text{BERT}} = \{H_i\}_{m+n+3} \in \mathbb{R}^{n_s \times d_w}$ are taken as the output of this layer, where $n_s$ is the length of the sentence.

B. CapsBERT

As depicted in Fig. 3, CapsBERT takes advantage of the structural characteristics of the triples for further knowledge extraction, which is designed as a pre-training step to get knowledge-BERT. CapsBERT is mainly optimized from two aspects. One is to improve the integrity of the knowledge graph, and the other is to better acquire local and global knowledge from triples.

Similar to context-BERT, the input of CapsBERT consists of three parts, namely token embedding, segmentation embedding and position embedding. The token embedding sequence can be symbolized as

$$T = [[\text{CLS}], Tok_{h1}, \ldots, Tok_{hm}, [\text{SEP}], Tok_{r1}, \ldots, Tok_{rn}, [\text{SEP}], Tok_{t1}, \ldots, Tok_{tk}, [\text{SEP}]]$$ \hspace{1cm} (4)

the first token of every input sequence is always a special classification token [CLS]. The head entity is represented as a sentence containing tokens $Tok_{h1}, \ldots, Tok_{hm}$, where $m$ is the length of the head entity. The relation is represented as

FIGURE 1. Overall architecture of KCF-NET (left), with the CapsBERT module illustrated (right). KCF-NET consists of three parts: embedding layer, fusion layer and output layer. Among them, embedding layer includes context embedding and knowledge graph embedding. The fusion layer will balance the weight of context representation and knowledge graph representation through highway module and attention mechanism. The output of KCF-NET is the start point and end point of the answer.

FIGURE 2. The input of the Context-BERT consists of questions and passages, both questions and passages have token, segment and position embeddings.
output. The output will be fed into the corresponding capsule.

to convert all features to a uniform size and concatenate as vector maps of different sizes, we use maximum pooling of the offset.

\[ o \]

the convolution kernel matrix, \( \mathbf{w} \). Different tokens in the same position have a same position embedding.

In addition, we employ CNN and CapsNet to enhance its knowledge expression ability. We use a variety of convolution kernels of different sizes to extract local features of different angles from the triple vector output by the knowledge-BERT. Capsule layer can encode the inherent spatial relationship between a part and a whole constituting viewport invariant knowledge that automatically generalizes to novel viewpoints.

A sentence containing tokens \( \text{Tok}_1, \ldots, \text{Tok}_n \), where \( n \) is the length of the relation. The tail entity is represented as a sentence containing tokens \( \text{Tok}_1, \ldots, \text{Tok}_k \), where \( k \) is the length of the tail entity. Different elements separated by \([\text{SEP}]\) have different segment embeddings, the tokens in sentences of head and tail entity share the same segment embedding \( \hat{a} \), while the tokens in relation sentence have a different segment embedding \( \hat{b} \). Different tokens in the same position have a same position embedding.

In addition, we employ CNN and CapsNet to enhance its knowledge expression ability. We use a variety of convolution kernels of different sizes to extract local features of different angles from the triple vector output by the knowledge-BERT. In this way, not only the overall characteristics of the triple, but also the local relationships between the items within the triple can be captured. The convolution process can be expressed as:

\[
f_{ij} = \text{ReLU}(\text{BatchNorm} \left( \sum_{v=1}^{k} w_v \cdot o_{v, j-i+1}^{\text{BERT}} + b \right))
\]

where \( w_v \) represents the value of the \( v \)-th column in the convolution kernel matrix, \( o_{v, j-i+1}^{\text{BERT}} \) represents the value of the \( i \)-th row, \( j-v+1 \) column in the output triple matrix of the BERT encoding layer, \( k \) represents the column width of the convolution kernel matrix and \( b \) represents the magnitude of the offset.

Since convolution kernels of different sizes can get feature vector maps of different sizes, we use maximum pooling to convert all features to a uniform size and concatenate as output. The output will be fed into the corresponding capsule.

We have designed two capsule layers to capture the relationship between triples. The first capsule consists of \( \eta \) capsules. The dimension of each capsule is \( 1 \times d_v \), where \( d_v \) is the dimension of the final hidden state of the knowledge-BERT coding layer. The second capsule layer is composed of one capsule, the dimension of the capsule is \( 1 \times d_{\text{cap}} \), whose output is the score the triple gets, predicting whether the triple is valid or not. The parameters in two capsule layers are updated by dynamic routing algorithm. However, the uniform distribution in original CapsNet ignores the differences among underlying capsules and increase the iterative calculation. To alleviate this problem, we propose a simi-dynamic routing algorithm. The simi-dynamic routing algorithm enhances the upward transfer of similar feature vectors, in the meanwhile, weakens the transfer of dissimilar feature vectors. Therefore, we use the similarity of the underlying capsule as the initial value, which makes the capsule network converge faster:

\[
similiar_j = \sum_i \hat{u}_j \ast \hat{u}_j^T
\]

\[
= \sum_i \begin{bmatrix} \hat{u}_{j1} \\ \hat{u}_{j2} \\ \vdots \\ \hat{u}_{ji} \end{bmatrix} \ast \begin{bmatrix} \hat{u}_{ji}^T \\ \hat{u}_{ji}^T \\ \vdots \\ \hat{u}_{ji}^T \end{bmatrix}
\]

Optimized dynamic routing algorithm is as followed:

Finally, we define the scoring formula of the triple as:

\[
f(h, r, t) = ||e||
\]

where \( e \) is the vector output of the second capsule layer, whose length is used as a score to predict whether the triple \((h, r, t)\) is valid or not. We compute a cross-entropy loss as a
Algorithm 1 Simi-Dynamic Routing

1: procedure Routing (u, r, l)
2:  
3:  for all capsule i in layer l and capsule j in layer (l+1):
4:      \( k_j \leftarrow \sum_i \hat{u}_j \ast \hat{u}_j^T \)
5:  
6:  for r iterations do
7:      for all capsule i in layer l:
8:         \( q_i \leftarrow \exp(k_i) \sum_{n_{cap}} \exp(k_i) \)
9:  
10:  for all capsule j in layer (l+1):
11:      \( s \leftarrow \sum q_i \hat{u}_i \)
12:  
13:  for all capsule j in layer (l+1):
14:      \( e \leftarrow \frac{||s||^2}{1 + ||s||^2} s \)
15:  
16:  for all capsule i in layer l and capsule j in layer (l+1):
17:      \( k_i \leftarrow k_i + \hat{u}_i \ast e \)
18:  
19: return e

training target:

\[ L = \sum_{(h, r, t) \in (T \cup T')} \ln(1 + \exp((-\theta) \cdot f(h, r, t))) \]

\[ \theta = \begin{cases} 
1, & (h, r, t) \in \mathbf{b}
-1, & (h, r, t) \in \mathbf{b}' 
\end{cases} \quad (8) \]

where \( \mathbf{b} \) represents the positive sample of the knowledge graph triple, and \( \mathbf{b}' \) represents the negative sample of the knowledge graph triple.

C. FUSION LAYER

The input of the fusion layer consists of two parts. Through the context-BERT, we can get the embedding of passages and questions with rich semantic information \( o^{BERT} \in R^{n_s \times d_w} \), and through the knowledge-BERT module, we can obtain the knowledge graph embedding \( o^{knowledge} \in R^{n_s \times d_k} \) of the entities in the text, where \( n_s \) is the sentence length, \( d_w \) is the output dimension of the context-BERT layer, and \( d_k \) is the output dimension of the knowledge-BERT layer. We integrate them through a nonlinear activation function:

\[ y_i = \sigma \left( w_{el} o^{BERT}_i + w_{el} o^{knowledge}_i + b \right) \quad (9) \]

where \( w_{el} \) and \( w_{el} \) are weight information, \( \sigma \) is activation function like RELU and \( b \) is offset. When the \( i \)-th word is a non-entity part, \( T_e = w_{el} o^{knowledge}_i \) is an all-zero vector, and only \( T_e = w_{el} o^{BERT}_i \) will be calculated. Through the information fusion module, we can get the knowledge fusion encoding \( y = \{ y_i \} \in R^{n_s \times d_w} \).

Different from the previous works, we believe that the original information in the context should be more decisive for question answering than external knowledge. Too much knowledge incorporation may divert the word vector from original sentence meaning. Therefore, in order to preserve the original information and avoid knowledge noise, we employ gating units to regulate the flow of information, including transform gate [39] and carry gate as shown in Fig. 4. The former determines how much original information is transferred to the subsequent module, whereas the latter chooses how much information to carry from the knowledge fusion encoding to the subsequent module. Therefore, a part of the information can quickly pass through a layer without transformation, while the rest needs to go through a nonlinear transformation layer first. The use of the gating structure has two advantages: one is to control the impact of knowledge fusion encoding on machine reading comprehension, and the other is to enable the contextual attention semantic information to skip the knowledge fusion module, which is easier to train:

\[ h_f = \sigma (w_h o^{BERT} + b) \quad (10) \]

\[ h_c = 1 - h_f \quad (11) \]

\[ H = h_f \cdot o^{BERT} + h_c \cdot y \quad (12) \]

FIGURE 4. The input of fusion layer comes from both context-BERT and knowledge-BERT. We adopt highway module to balance the weight of context and external knowledge, together with attention mechanism.

where \( h_f \) represents transform gate and \( h_c \) represents carry gate. Since the information fusion module is calculated based on each word, the fusion vector for each word in the sentence will lack context information between other words. Therefore, A multi-head attention mechanism with residual connection is added to encode the resulting fusion vector.

\[ \tilde{H} = \left[ \tilde{h}_1, \ldots, \tilde{h}_n \right] = \mathcal{C}(h_1, \ldots, h_n) \quad (13) \]

\[ o^{fusion} = \Gamma(\tilde{H} + H) \quad (14) \]
where $\theta$ represents the multi-head attention mechanism, $\beta$ represents batch normalization and $o_{\text{fusion}}$ is the output of the knowledge fusion layer, $o_{\text{fusion}} = \{o_i \in \mathbb{R}^{n_i \times d_i}\}$.

### D. OUTPUT LAYER

Finally, in the output layer, we use dense layer and Softmax to obtain the probability of the answer boundary for each position, as depicted in Fig. 5:

![Diagram of the output layer](image)

**FIGURE 5.** The output of the dense layer and Softmax is the probability of the answer boundary, including the start position and end position.

The probability of each token as the start or end of the answer label is calculated as follows:

$$y^{1} = \frac{\exp(w_1v_i)}{\sum_j \exp(w_1v_j)}$$  \hspace{1cm} (15)

$$y^{2} = \frac{\exp(w_2v_i)}{\sum_j \exp(w_2v_j)}$$  \hspace{1cm} (16)

where $v_i$ represents the output of the knowledge fusion layer $o_{\text{fusion}}$ passes the $i$-th row of the output vector matrix of the fully connected layer, $w_1, w_2$ represent the learned weight parameters, and the loss function is as follows:

$$L = -\frac{1}{N} \sum_{j=1}^{N} (y_1^j \log y_j^1 + y_2^j \log y_j^2)$$  \hspace{1cm} (17)

where $N$ is the size of the batch and $y_1^j, y_2^j$ are the ground truths of the starting point and the ending point.

### IV. EXPERIMENTS

We evaluate our approach from two aspects, namely knowledge expression and reading comprehension ability. Specifically, the knowledge expression ability is mainly to conduct experimental tests on CapsBERT in the task of triple classification and knowledge-BERT module in the task of link prediction. The accuracy of reading comprehension is evaluated on the entire KCF-NET model. This section will be elaborated in the chronological order of the experiment.

#### A. TRIPLE CLASSIFICATION

1) DATASETS

We ran CapsBERT on two widely used benchmark datasets: WN11 and FB13. The former is a subset of WordNet, and the latter is a subset of Freebase. Table 1 provides statistics of the two datasets mentioned above.

| Dataset | #E | #R | Train | Valid | Test |
|---------|----|----|-------|-------|-----|
| WN11    | 38,696 | 11  | 112,581 | 2,609 | 10,544 |
| FB13    | 75,043 | 13  | 316,232 | 5,908 | 23,733 |

2) EVALUATION PROTOCOL

For the triple classification task, we use the accuracy rate to evaluate the classification effect. We compare our CapsBERT with multiple state-of-the-art KG embedding methods including:

- **TransE** [17] used a distributed representation to describe the triples in KGs, which could obtain semantic information through simple mathematical calculations based on spatial invariance.

- **TransG** [21] proposed a variant of TransE, which leveraged a Bayesian non-parametric infinite mixture model to handle multiple relation semantics by generating multiple translation components for a relation.

- **DistMult** [40] proposed a novel approach that utilizes the learned relation embeddings to mine logical rules.

- **ConvKB** [24] proposed a novel use of CNN for KB completion task.

- **AATE** [41] proposed a mutual attention mechanism between relation mention and entity description to learn more accurate textual representations for further improving knowledge graph representation.

- **KG-BERT** [25] fine-tuned BERT model on textual sequences for predicting the plausibility of a triple or a relation.

- **ERNIE** [8] proposed a knowledgeable language representation model, which aggregated both context and knowledge facts in large-scale textual corpora and KGs for predicting both tokens and entities.

- **K-BERT** [9] utilized domain-specific knowledge to train a knowledge-enabled language representation model and introduced soft position and visible matrix to limit the impact of knowledge.

3) QUANTITATIVE EVALUATION

Triple classification aims to judge whether a given triple $(h, r, t)$ is correct or not. Since the model we proposed is based on supervised learning, positive and negative samples are needed to guide the model training. However, the training set of WN11 and FB13 does not contain negative samples, only positive samples. Therefore, we need to perform negative sample filling on these two data sets before the experiment. The negative triple set $d^*$ is simply generated by replacing head entity $h$ or tail entity $t$ in a positive triple $(h, r, t) \in d$ with a random entity $h'$ or $t'$, i.e.

$$d = \{(h', r, t) \mid h' \neq h \land (h, r, t) \in d\} \cup \{(h, r, t') \mid t' \neq t \land (h, r, t) \in d\}$$
Note that a triple will not be treated as a negative example if it is already in positive set $d$. In order to keep the classification effect balanced, we generate negative sample data with the same number of positive samples.

We first pre-trained knowledge graph embeddings, initializing parameters of the BERT encoding layer with models officially released by Google. These models were pre-trained on the concatenation of Books Corpus (800M words) and Wikipedia (2,500 words), using the tasks of masked language model and next sentence prediction.

We use BERT-Tiny, BERT-Small and BERT-Base for comparative experiments. In the convolutional layer, we use separable convolution and pooling of different layers to convert the output of BERT with different hidden state dimensions into the input size of capsule network. Since the input of the triple classification task is only the splicing of entities and relationships, we set the maximum length to 20.

As the results obtained in different environments may be different, we use KG-BERT as our benchmark model, where KG-BERT$^*$ means the accuracy rate obtained by training in the local environment. It can be seen from Table 2 that although the proposed model CapsBERT does not obtain the highest accuracy, the overall size of the model is reduced by three to seven times compared with recent work based on BERT, while the average accuracy only decreases by no more than 3% to 89.25%, outperforming the traditional triple classification model.

### TABLE 2. Triple classification accuracy of different methods.

| Method   | Size | WN11 | FB13 | AVG  |
|----------|------|------|------|------|
| TransE   | /    | 75.9 | 81.5 | 78.7 |
| TransG   | /    | 87.4 | 87.3 | 87.4 |
| DistMult | /    | 87.1 | 86.2 | 86.7 |
| AATE     | /    | 88.0 | 87.2 | 87.6 |
| ConvKB   | /    | 87.6 | 88.8 | 88.2 |
| KG-BERT  | 420M | 93.5 | 90.4 | 91.9 |
| KG-BERT$^*$ | 420M | 93.3 | 90.1 | 91.7 |
| K-BERT   | 592M | 91.2 | 90.2 | 90.7 |
| ERNIE    | 750M | 92.2 | 90.2 | 91.2 |
| Our model | 111M | 90.3 | 88.2 | 89.25 |

### B. ABLATION EXPERIMENTS ON CAPSBERT

In order to analyze the impact of different components in CapsBERT on knowledge expression, we designed four different training strategies in the training process, as shown in Fig. 6.

Strategy A: Directly use BERT for triple classification without a capsule network. We selected Google pre-trained BERT-Tiny, BERT-Small and BERT-Base for experiments and fine-tuned them.

Strategy B: Train CapsBERT as a whole.

Strategy C: First train BERT separately and freeze its parameters, and then train the entire CapsBERT, the BERT parameters will not be updated during the overall training process.

Strategy D: First train BERT and then train the entire CapsBERT, that is, the knowledge-BERT layer will be trained twice.

From Table 3, we have observed that original BERT can perform well and achieve an accuracy rate of 93.11%, whereas the accuracy rate of model in Strategy B is only 69.68%, which is almost down to the distribution rate of the experimental data set. From this, we can conclude that knowledge-BERT will not generate appropriate contextual semantic information without pretraining and cause the capsule layer unable to extract spatial information between entities, resulting in poor results. Both Strategy C and Strategy D have worked well. In contrast, strategy C is limited due to freezing parameters. Pre-training and re-training make the triple semantic representation obtained by knowledge-BERT better meet the requirements of the capsule network. Strategy C and strategy D also prove that pre-trained BERT can achieve better results in retraining downstream tasks.

### TABLE 3. Triple classification accuracy of ablation experiments.

| Model     | Strategy A | Strategy B | Strategy C | Strategy D |
|-----------|------------|------------|------------|------------|
| BERT-Tiny | 82.75      | 72.90      | 80.72      | 83.11      |
| BERT-Small| 89.61      | 69.68      | 87.87      | 90.29      |
| BERT-Base | 93.11      | -          | -          | -          |
C. LINK PREDICTION

1) DATASETS
SemEval 2010 Task 8 is used for link prediction. The SemEval 2010 Task 8 dataset includes 8000 training samples and 2717 test samples, each of which includes a text sentence and the category label of the relationship between two entities in the sentence, using \(<e_1>\), \(<e_1>\) and \(<e_2>\), \(<e_2>\) to identify the location of the entity. The relationship distribution of the data set is shown in Fig. 7:

![Figure 7. The distribution of SemEval 2010 Task 8. (CE: Cause-Effect, IA: Instrument-Agency, PP: Product-Producer, CC: Content-Container, EO: Entity-Origin, ED: Entity-Destination, CW: Component-Whole, MC: Member-Collection, MT: Message-Topic).](image)

2) EVALUATION PROTOCOL
For the relationship classification task, we use the F1 score to evaluate. The F1 score takes into account both the accuracy rate and recall rate of the classification model. The calculation formula of each evaluation index is as follows:

\[
P = \frac{TP}{TP + FP} \tag{18}
\]
\[
R = \frac{TP}{TP + TN} \tag{19}
\]
\[
F1 = \frac{2 \times P \times R}{P + R} \tag{20}
\]

where TP represents the number of samples that are correctly predicted in a category, FP represents the number of samples in other categories that are incorrectly predicted as the category, and TN is the number of samples in the category that are incorrectly predicted as other categories.

We compare our model with some other strong baselines:

- **Entity Attention Bi-LSTM [42]** adopted dynamic long short-term memory to extract complex relations between entities.

- **Multi-Attention CNN [43]** proposed a novel convolutional neural network architecture for relation classification, relying on two levels of attention to discern patterns in heterogeneous context.

- **R-BERT [44]** leveraged the pre-trained BERT language model and incorporates information from the target entities to tackle the relation classification task.

- **BERT\_EM+MTB [45]** utilized BERT and distributional similarity for relation learning.

3) QUANTITATIVE EVALUATION
In order to employ knowledge-BERT in CapsBERT, that is, the BERT layer that has been trained by the triple classification task to be used in subsequent machine reading comprehension, we introduce the SemEval 2010 Task8 data set for link prediction tasks to modify its input form. Link prediction aims to predict the relationship category between two entities, showing the model’s ability to analyze the contextual semantic information of the entities in the text, and detect connections between entities.

The experimental results are shown in Table 4. Obviously, the knowledge-BERT (base) obtain the current best F1 score of 91.83, which increases 2.64% compared to BERT. The performance of the knowledge-BERT (small) does not exceed the initial BERT due to the limit of parameters, but compared with the method based on CNN or LSTM, it has made great progress.

![Table 4. The results of link prediction on different methods.](image)

| Model                     | F1 score |
|---------------------------|----------|
| Entity Attention Bi-LSTM  | 85.2     |
| Multi-Attention CNN       | 88.0     |
| R-BERT                    | 89.25    |
| BERT\_EM+MTB              | 89.5     |
| BERT                      | 89.19    |
| Our model (small)         | 88.64    |
| **Our model (base)**      | **91.83**|

In order to visually analyze the experimental results, we draw a confusion matrix graph for the prediction results of each test sample, as shown in Fig. 8. It can be seen that our model can effectively distinguish each relationship category. After joint training with capsule network, knowledge-BERT can not only express the semantic information of the entity itself, but also be able to contact the relationship between the entities. In the subsequent machine reading comprehension tasks, we will integrate knowledge-BERT into KCF-NET.

![Figure 8. Confusion matrix graph for link prediction on knowledge-BERT.](image)

D. MACHINE READING COMPREHENSION

1) DATASETS
SQuAD, regarded as a milestone for MRC, collects 536 articles from Wikipedia, requiring crowd-workers to ask
more than 100,000 questions. The answer to each question is a span from the corresponding passage. In this paper, we choose SQuAD 1.1 since we focus more on answerable questions.

2) EVALUATION PROTOCOL
For machine reading comprehension tasks, the input of the model is the stitching of the passage $P$ and the question $Q$, the output is the start position and end position of the answer, and the text between consecutive positions is intercepted as the predicted answer. There are two indicators to evaluate our model:

(1) Exact Match (EM)
The exact match EM refers to the percentage that the predicted answer exactly matches a certain answer in the correct answer list, that is, whether the predicted answer text is exactly the same as the real answer text.

(2) F1 score
The F1 score index refers to the average coincidence rate between the answers predicted by the model and the correctly marked answers.

We compare KCF-NET with some other strong baselines on SQuAD including:

BiDAF [46] used bidirectional attention flow mechanism to obtain a query-aware context representation without early summarization.

QANet [47] used CNN and self-attention instead of recurrent neural networks to model local and global interactions, respectively.

KAR [48] integrated the general knowledge with the MRC model and used attention mechanisms to improve the robustness to noise.

RM-Reader [49] designed a memory-based answer prediction to gradually increase the reading knowledge and continuously refine the answer span.

SLQA [50] proposed a multiple granularity hierarchical attention MRC model, which performed attention and fusion at different levels.

3) QUANTITATIVE EVALUATION
In KCF-NET, we choose BERT-Base as context-BERT to extract contextual semantic information, and for knowledge-BERT, we use the BERT-Small model that has been trained in CapsBERT, which has 4 layers and 8 self-attention heads. The dimension of the output hidden state vector is 512. In addition, we also set the maximum sentence length of the input to 384. For hyper-parameters, we set Dropout to 0.1, learning rate 5e-5, and batch size to 16.

We use BERT-Base and BERT-Large as baselines for comparative experiments. As can be seen from Table 5, our model has a greater improvement than BERT-Base. In the verification set, EM has increased by 0.6 percentage points and F1 has increased by 0.46 percentage points, indicating the integration of external knowledge can improve the performance of pre-trained language models towards MRC. Compared with BERT-Large, due to the gap in the number of parameters, our model has not been able to obtain better results.

| Model       | EM (Validation) | F1 (Validation) | EM (Test) | F1 (Test) |
|-------------|-----------------|-----------------|-----------|-----------|
| Human       | 80.3            | 90.5            | 82.30     | 91.22     |
| BiDAF       | 77.6            | 86.2            | 80.0      | 86.7      |
| QANet       | 75.9            | 84.1            | 76.2      | 84.6      |
| BERT (Base) | 80.8            | 88.5            | 82.1      | 89.0      |
| BERT (Large)| 84.1            | 90.9            | 85.1      | 91.8      |
| KAR         | 75.7            | 82.8            | 76.1      | 83.5      |
| RM-Reader   | 81.3            | 88.6            | 82.7      | 89.8      |
| SLQA        | 81.5            | 90.2            | 83.2      | 90.1      |
| Our model   | 81.40           | 88.96           | 82.8      | 90.2      |

We use BERT-Base and BERT-Large as baselines for comparative experiments. As can be seen from Table 5, our model has a greater improvement than BERT-Base. In the verification set, EM has increased by 0.6 percentage points and F1 has increased by 0.46 percentage points, indicating the integration of external knowledge can improve the performance of pre-trained language models towards MRC. Compared with BERT-Large, due to the gap in the number of parameters, our model has not been able to obtain better results.

E. ABLATION EXPERIMENTS ON KCF-NET
In order to study the impact of different components on performance of KCF-NET, we design three strategies for further experiments:

Strategy A: Directly use context-BERT for machine reading comprehension without knowledge-BERT and knowledge fusion layer. We choose Google pre-trained BERT-Base for experiments and fine-tune.

Strategy B: Employ context-BERT and knowledge-BERT. Add them directly for fusion.

Strategy C: Use the complete KCF-NET, containing context-BERT, knowledge-BERT and knowledge fusion layer.

Figure 9 shows the model structure of strategy A and strategy B. As can be seen from Table 6, each component of
TABLE 8. The experiment of bert embedding in KCF-NET with different hidden layers, attention heads and hidden state dimension.

| Group | NO | Hidden layer | Attention heads | Hidden state dimension | BERT EM | BERT F1 | KCF-NET EM | KCF-NET F1 | epoch | time  |
|-------|----|--------------|-----------------|------------------------|--------|--------|-----------|-----------|-------|-------|
| A     | 1  | 2            | 2               | 128                    | 25.56  | 37.96  | 32.78     | 45.21     | 3     | 0:33:09 |
|       | 2  | 2            | 4               | 256                    | 40.05  | 51.89  | 46.69     | 58.35     | 3     | 0:42:12 |
|       | 3  | 2            | 8               | 512                    | 53.92  | 65.40  | 57.91     | 69.11     | 3     | 0:59:36 |
|       | 4  | 2            | 12              | 768                    | 57.90  | 68.69  | 60.2      | 70.92     | 3     | 1:28:46 |
| B     | 5  | 4            | 2               | 128                    | 38.49  | 48.97  | 44.93     | 55.93     | 3     | 0:40:44 |
|       | 6  | 4            | 4               | 256                    | 59.59  | 70.00  | 64.43     | 74.40     | 3     | 1:04:10 |
|       | 7  | 4            | 8               | 512                    | 70.66  | 80.26  | 71.85     | 81.08     | 3     | 1:36:42 |
|       | 8  | 4            | 12              | 768                    | 74.00  | 82.53  | 74.23     | 82.94     | 3     | 2:29:50 |
| C     | 9  | 6            | 2               | 128                    | 44.91  | 55.99  | 52.56     | 63.75     | 3     | 0:49:38 |
|       | 10 | 6           | 4               | 256                    | 62.83  | 72.91  | 67.21     | 76.76     | 3     | 1:19:40 |
|       | 11 | 6           | 8               | 512                    | 75.04  | 83.14  | 75.54     | 83.89     | 3     | 2:06:22 |
|       | 12 | 6           | 12              | 768                    | 78.68  | 86.4   | 79.04     | 87.20     | 3     | 3:24:17 |
| D     | 13 | 8           | 2               | 128                    | 48.25  | 58.95  | 55.14     | 66.01     | 3     | 0:55:03 |
|       | 14 | 8           | 4               | 256                    | 62.83  | 72.91  | 71.97     | 80.77     | 2     | 1:41:36 |
|       | 15 | 8           | 8               | 512                    | 75.04  | 83.14  | 77.50     | 85.53     | 2     | 2:08:31 |
|       | 16 | 8           | 12              | 768                    | 78.68  | 86.40  | 80.14     | 87.76     | 2     | 3:20:19 |
| E     | 17 | 10          | 2               | 128                    | 52.18  | 63.11  | 58.48     | 68.96     | 3     | 1:02:33 |
|       | 18 | 10          | 4               | 256                    | 73.62  | 82.54  | 74.47     | 82.99     | 2     | 1:50:22 |
|       | 19 | 10          | 8               | 512                    | 78.32  | 86.12  | 78.36     | 86.13     | 2     | 3:40:57 |
|       | 20 | 10          | 12              | 768                    | 80.23  | 88.33  | 81.49     | 88.58     | 2     | 5:03:10 |
| F     | 21 | 12          | 2               | 128                    | 55.08  | 66.45  | 61.30     | 71.98     | 3     | 1:08:33 |
|       | 22 | 12          | 4               | 256                    | 73.00  | 81.50  | 74.80     | 83.07     | 2     | 2:04:54 |
|       | 23 | 12          | 8               | 512                    | 80.02  | 87.50  | 81.11     | 87.68     | 2     | 4:08:22 |
|       | 24 | 12          | 12              | 768                    | 81.16  | 88.58  | 81.40     | 88.96     | 2     | 5:45:42 |

FIGURE 9. Different components of KCF-NET. (a) without knowledge-BERT and knowledge fusion layer. (b) without knowledge fusion layer.

KCF-NET has played a certain role, of which the effect of BERT is the most obvious. In contrast, we can conclude that the introduction of external knowledge has a positive effect on reading comprehension, but inappropriate integration mechanisms can mislead the model to make wrong predictions.

F. EFFECTS OF KNOWLEDGE EMBEDDING LAYER

Experiments are also conducted to analyze the influence of different knowledge graph data. We compared the different combinations of WN11, FB13, and SemEval 2010 Task8 as depicted in Fig. 10:

- Strategy A: Train CapsBERT on WN11, train knowledge-BERT on SemEval 2010 Task8 and train KCF-NET on SQuAD1.1.
- Strategy B: Train CapsBERT on FB13, train knowledge-BERT on SemEval 2010 Task8 and train KCF-NET on SQuAD1.1.
- Strategy C: Train CapsBERT on WN11 and FB13, train knowledge-BERT on SemEval 2010 Task8 and train KCF-NET on SQuAD1.1.

It can be seen from Table 7 that different knowledge graph datasets also have a great impact on the knowledge graph embedding. We find that knowledge-BERT trained...
FIGURE 11. Line charts of impact of different hidden layers. With the increase of the hidden state dimension, the EM score and F1 score have been greatly improved, but the larger the hidden state dimension, the smaller the increase.

on WN11 has better performance than that trained on the FB13.

We infer that it may be related with the composition of the KGs. WN11, which consists of several words and corresponds to distinct word facts, is more suitable for constructing word semantic vectors for machine reading comprehension. Besides, the result of Strategy C shows that knowledge graphs can make up for each other to some extent.

G. EFFECTS OF BERT EMBEDDING

In order to study the influence of BERT coding layers with different structures and different parameters on the final result, we have done the further experiments. We take the number of layers as \{2, 4, 6, 8, 10, 12\}, the dimension of output vector as \{128, 256, 512, 768\} and the number of multi-heads as \{2, 4, 6, 8, 12\}, building a total of 48 models. Among them, 24 models are bare BERT, the rest are KCF-NET. The experimental results are shown in Table 8.

According to Table 8 we find that the dimension of hidden state, the number of hidden layers and attention heads all have a positive impact on MRC. However, as the number of the parameters increases, the training time also increases, so it is necessary to balance performance and complexity in actual application. To visually compare different parameters, we first grouped all models according to the number of hidden layers and plotted curves as shown in Fig. 11. Obviously, with the increase of the hidden state dimension, the EM score and F1 score have been greatly improved, but the larger the hidden state dimension, the smaller the increase. In addition, we find that by fusing external knowledge, KCF-NET can
indeed improve the performance of MRC to a certain extent. When the hidden state dimension is small, the improvement is very significant. In most cases, KCF-NET’s EM score is always slightly greater than the F1 score. We believe that this is the positive effect of external knowledge on entity coding, which can make the original fuzzy position prediction more precisely. Since the F1 score itself calculates the degree of similarity, which is a vague indicator, the improvement is not as high as the EM score.

Then, we adjust the groups of different models for further observation. We re-assigned 4 groups according to the number of attention-heads, as shown in Fig. 12. Group (a) contains the model \{1, 5, 9, 13, 17, 21\} in Table 8, group (b) contains model \{2, 6, 10, 14, 18, 22\}, group (c) contains model \{3, 7, 11, 15, 19, 23\} and group (d) contains the model \{4, 8, 12, 16, 20, 24\}. The hidden state dimensions in the four groups are 128, 256, 512, and 768, respectively.

We find that with the increase of the number of hidden layers, the model’s EM score and F1 score have been greatly improved, especially when the number of hidden layers is small. In addition, compared with Fig. 11, when the number of hidden layers increases, the change in the increase of KCF-NET is more volatile than that of BERT, and it is not entirely in a downward trend. We speculate that this is because the number of hidden layers in BERT does not directly affect the information representation of the final word, whereas the knowledge coding layer in KCF-NET directly affects the entity representation, so the impact is more obvious. Overall, this experiment proves that the knowledge coding layer and knowledge fusion layer in KCF-NET have a positive effect on the entire model, especially for models with small parameters. But for BERT with a large number of parameters, more word context information can be learned in the pre-training stage. At this point, it is similar to the original intention of adding external background knowledge in this paper. Therefore, KCF-NET has a limited improvement on BERT with a larger number of parameters, but it still has a positive effect. It also proves the effect of external knowledge on improving the performance of machine reading comprehension.

H. EFFECTS OF TRANSFORMER LAYER

In previous experiments, we have studied the BERT coding layers with different parameters. In this section, the influence of the information carried by different layers in BERT will be revealed.

As shown in the Fig.13, we distinguish the different Transformer layers of BERT-Base in different colors. We design four strategies as follows:

Strategy A: Only use the first layer of BERT-Base to connect MLP and Softmax for output.
Strategy B: Only use the last layer of BERT-Base.
The comparison experiment of different transformer layers in the task of triple classification, and the instantiation parameters are reduced by three-quarters. On the link prediction task, knowledge-BERT obtains the current best F1 score of 91.83. Compared with the performance of BERT-Base on SQuAD, our model has a higher accuracy rate with negligible parameter increase. This work demonstrates the feasibility of further enhancing advanced language models with knowledge from KBs, which indicates a potential direction for future research.

**REFERENCES**

[1] 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. 2383–2392.

[2] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. “RACE: Large-scale ReAding comprehension dataset from examinations,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 785–794.

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

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

[5] J. Liu, Y. Yang, S. Lv, J. Wang, and H. Chen. “Attention-based BiGRU-CNN for Chinese question classification,” J. Ambient Intell. Humanized Comput., pp. 1–12, June 2019, doi: 10.1007/s12652-019-01344-9.

[6] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. “Learning entity and relation embeddings for knowledge graph completion,” in Proc. 29th AAAI Conf. Artif. Intell., 2015. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571

[7] S. Liu, X. Zhang, S. Zhang, H. Wang, and W. Zhang. “Neural machine reading comprehension: Methods and trends,” Appl. Sci., vol. 9, no. 18, pp. 3698, Sep. 2019.

[8] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. “ERNIE: Enhanced machine reading representation with informative entities,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 1441–1451.

[9] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang. “K-BERT: Enabling language representation with knowledge graph,” in Proc. AAAI Conf. Artif. Intell., Apr. 2020, vol. 34, no. 3, pp. 2901–2908.

[10] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. “Teaching machines to read and comprehend,” presented at the 29th Annu. Conf. Neural Inf. Process. Syst., 2015.

[11] J. E. Weston, A. D. Szlam, R. D. Fergus, and S. Sukhbaatar. “End-to-end memory networks,” presented at the 29th Annu. Conf. Neural Inf. Process. Syst., 2015.

[12] B. Dhiung, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov. “Gated-attention readers for text comprehension,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2017, pp. 1832–1846.

[13] T. Long, E. Bengio, R. Lowe, J. C. K. Cheung, and D. Precup. “World knowledge for reading comprehension: Rare entity prediction with hierarchical LSTMs using external descriptions,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 825–834.

[14] A. Miller, A. Fisch, J. Dodge, A. H. Karimi, A. Bordes, and J. Weston. “Key-Value Memory Networks for Directly Reading Documents,” presented at the Conf. Empirical Methods Natural Lang. Process., 2016.

[15] C. Wang and H. Jiang. “Exploring machine reading comprehension with explicit knowledge,” 2016, arXiv:1608.07905. [Online]. Available: http://arxiv.org/abs/1608.07905

[16] B. Pan, Y. Yang, H. Li, Z. Zhao, Y. Zhuang, D. Cai, and X. He. “MAC-NET: Transferring knowledge from machine comprehension to sequence-to-sequence models,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 6092–6102.

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

[18] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. “Freebase: A collaboratively created graph database for structuring human knowledge,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, (SIGMOD), 2008, pp. 1247–1250.

[19] F. M. Suchanek, G. Kasneci, and G. Weikum. “Yago: A core of semantic knowledge,” in Proc. 16th Int. Conf. World Wide Web, Banff, AB, Canada, 2007, pp. 697–706.

[20] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. “Translating embeddings for modeling multi-relational data,” in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 2787–2795.
Z. Zhang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in Proc. AAAI, 2014, pp. 1112–1119.

R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation learning of knowledge graphs with entity descriptions,” in Proc. AAAI Conf. Artif. Intell., 2016, pp. 1–7.

G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, “Knowledge graph embedding via dynamic mapping matrix,” in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process., vol. 1, 2015, pp. 687–696.

H. Xiao, M. Huang, and X. Zha, “TransG: A generative model for knowledge graph embedding,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 2316–2325.

S. He, K. Liu, G. Ji, and J. Zhao, “Learning to represent knowledge graphs with Gaussian embedding,” in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2015, pp. 623–632.

G. Ji, K. Liu, S. He, and J. Zhao, “Knowledge graph completion with adaptive sparse transfer matrix,” in Proc. AAAI Conf. Artif. Intell., 2016, pp. 1–7.

D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, “A novel embedding model for knowledge base completion based on convolutional neural network,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 2, 2018, pp. 327–333.

Y. Yao, C. Mao, and Y. J. A. Luo, “KG-BERT: BERT for knowledge graph completion,” CoRR, vol. abs/1909.03193, 2019. [Online]. Available: https://arxiv.org/abs/1909.03193

P. Zhong, D. Wang, and C. Miao, “Knowledge-enriched transformer for emotion detection in textual conversations,” in Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP), 2019, pp. 165–176.

J. Liu, L. Lin, H. Ren, M. Gu, J. Wang, G. Yuan, and J.-U. Kim, “Building neural network language model with POS-based negative sampling and stochastic conjugate gradient descent,” Soft Comput., vol. 22, no. 20, pp. 6705–6717, Oct. 2018.

B. Yang and T. Mitchell, “Leveraging knowledge bases in LSTMs for improving machine reading,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, Vancouver, BC, Canada: Association for Computational Linguistics, 2017, pp. 1436–1446.

Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “ERNIE: Enhanced language representation with informative entities,” in Proc. 57th Annu. Meeting Assoc. for Comput. Linguistics, 2019, pp. 1441–1451.

T. Mihaylov and A. Frank, “Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2018, pp. 821–832.

C. Wang and H. Jiang, “Explicit utilization of general knowledge in machine reading comprehension,” in Proc. 57th Conf. Assoc. Comput. Linguistics, 2019, pp. 2263–2272.

A. Yang, Q. Wang, J. Liu, K. Lin, Y. Lyu, H. Wu, Q. She, and S. Li, “Enhancing pre-trained language representations with rich knowledge for machine reading comprehension,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 2346–2357.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 1441–1451.

S. Chang and J. Liu, “Multi-lane capsule network for classifying images with complex background,” IEEE Access, vol. 8, pp. 79876–79886, 2020.

S. K. Sahu, P. Kumar, and A. P. Singh, “Dynamic routing using inter capsule routing protocol between capsules,” in Proc. UKSim-AMSS 20th Int. Conf. Comput. Modeling Simulation (UKSim), Mar. 2018, pp. 1–5.

C. Zhang and P. Woodland, “Semi-tied units for efficient gaiting in LSTM and highway networks,” in Proc. Interspeech, Sep. 2018, pp. 1773–1777.

Z. Zhang, F. Zhuang, M. Qu, F. Lin, and Q. He, “Knowledge graph embedding with hierarchical relation structure,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 3198–3207.

B. An, B. Chen, X. Han, and L. Sun, “Accurate text-enhanced knowledge graph representation learning,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 1, 2018, pp. 745–755.

J. Liu, H. Ren, M. Wu, J. Wang, and H.-J. Kim, “Multiple relations extraction among multiple entities in unstructured text,” Soft Comput., vol. 22, no. 13, pp. 4295–4305, Jul. 2018.

L. Wang, Z. Cao, G. de Melo, and Z. Liu, “Relation classification via multi-level attention CNNs,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 1298–1307.

S. Wu and Y. He, “Enriching pre-trained language model with entity information for relation classification,” in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage., Nov. 2019, pp. 2361–2364.

L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, “Matching the blanks: Distributional similarity for relation learning,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 2985–2995.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” 2016, arXiv:1611.01603. [Online]. Available: https://arxiv.org/abs/1611.01603

A. W. Yu, D. Dohan, M. T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, “QANet: Combining local convolution with global self-attention for reading comprehension,” in Proc. 6th Int. Conf. Learn. Represent., 2018. [Online]. Available: https://arxiv.org/abs/1804.09541

C. Wang and H. Jiang, “Explicit utilization of general knowledge in machine reading comprehension,” in Proc. 57th Conf. Assoc. Comput. Linguistics, 2019, pp. 2263–2272.

M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, “Reinforced mnemonic reader for machine reading comprehension,” in Proc. 27th Int. Joint Conf. Artif. Intell., 2018, pp. 4099–4106.

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, 2018, pp. 1705–1714.

PEIZHU GONG received the B.E. degree from Shanghai Maritime University, in 2019, where he is currently pursuing the M.E. degree. His current research interests include deep learning, natural language processing, and machine learning.

JIN LIU (Member, IEEE) received the B.S. degree from Lanzhou University, the M.S. degree from the University of Electrical Science and Technology of China, and the Ph.D. degree from Washington State University. He is currently a Professor with Shanghai Maritime University. His research interests include deep learning, natural language processing, and computer vision. He is a member of IEEE, CAAI, and CCF.

YIHE YANG received the B.E. degree from Shanghai Maritime University, in 2018, where he is currently pursuing the M.E. degree. His current research interests include deep learning, natural language processing, and machine learning.

HUIHUA HE received the Ph.D. degree from Washington State University, in 2007. She is currently an Associate Professor with the College of Education, Shanghai Normal University. Her research interests include ICT uses in education, affective science, social-emotional learning, and related topics. She is also a member of CSE.