Multi-strategy Knowledge Distillation Based Teacher-Student Framework for Machine Reading Comprehension

Xiaoyan Yu[1,2,3], Qingbin Liu[1,2(✉)], Shizhu He[1,2], Kang Liu[1,2], Shengping Liu[5], Jun Zhao[1,2], and Yongbin Zhou[3,4]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[2] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{xiaoyan.yu,qingbin.liu,shizhu.he,kliu,jzhao}@nlpr.ia.ac.cn
[3] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[4] School of Cyber Security, Nanjing University of Science and Technology, Nanjing, China
zhouyongbin@njust.edu.cn
[5] Beijing Unisound Information Technology Co., Ltd., Beijing, China
liushengping@unisound.com

Abstract. The irrelevant information in documents poses a great challenge for machine reading comprehension (MRC). To deal with such a challenge, current MRC models generally fall into two separate parts: evidence extraction and answer prediction, where the former extracts the key evidence corresponding to the question, and the latter predicts the answer based on those sentences. However, such pipeline paradigms tend to accumulate errors, i.e. extracting the incorrect evidence results in predicting the wrong answer. In order to address this problem, we propose a **M**ulti-**S**trategy **K**nowledge **D**istillation based **T**eacher-**S**tudent framework (**MSKDTS**) for machine reading comprehension. In our approach, we first take evidence and document respectively as the input reference information to build a teacher model and a student model. Then the multi-strategy knowledge distillation method transfers the knowledge from the teacher model to the student model at both feature and prediction level through knowledge distillation approach. Therefore, in the testing phase, the enhanced student model can predict answer similar to the teacher model without being aware of which sentence is the corresponding evidence in the document. Experimental results on the ReCO dataset demonstrate the effectiveness of our approach, and further ablation studies prove the effectiveness of both knowledge distillation strategies.

Keywords: Machine reading comprehension · Knowledge distillation · Evidence sentence

S. Li et al. (Eds.): CCL 2021, LNAI 12869, pp. 209–225, 2021.
https://doi.org/10.1007/978-3-030-84186-7_14

1 Introduction

Machine reading comprehension (MRC) is a task that enables machines to read and understand natural language documents to answer questions. Since it well indicates the ability of machines in interpreting natural language as well as having a wide range of application scenarios, it has attracted extensive attention from academia and industry over the recent years. Prevailing MRC datasets define their tasks as either extracting spans from reference documents to answer questions, such as SQuAD [28] and CoQA [29], or inferring answers based on pieces of evidence from a given document, which is also referred to as non-extractive MRC, including multiple-choice MRC [16,30], open domain question answering [5] and so on.

Current MRC faces the significant challenge of the irrelevant information in documents causing negative impact on answer predicting. Therefore, our aim is to engage the model to focus on evidence sentences in documents and using them to answer corresponding questions accurately. To illustrate, consider the example shown in Fig. 1 (adapted from the ReCO dataset [35]). In this sample document, only the evidence sentences have a significant impact on predicting the answer; the other sentences are irrelevant information that may confuse the model and preventing it from focusing on the evidence sentences, thus affecting the correctness of answer predicting.

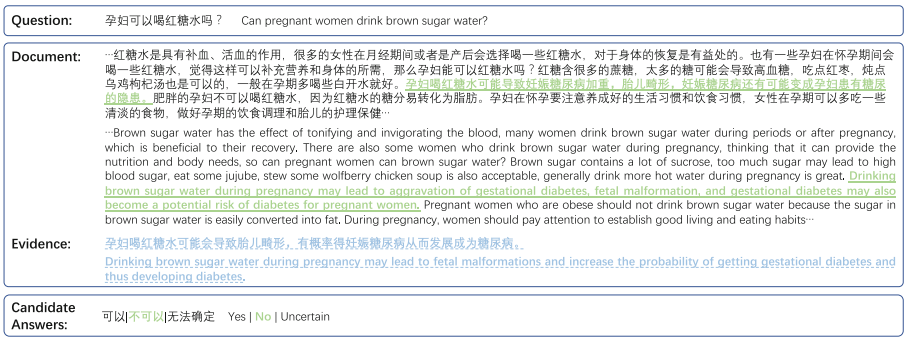

Fig. 1. Example of multiple-choice machine reading comprehension. The sentence in green is the evidence sentence for answering the given question in this document, which is of great importance. Other sentences contain irrelevant information, while potentially negatively affecting the answer prediction. The sentence in blue is the evidence obtained by manual annotation (summarized or paraphrased by the annotator). (Color figure online)

Previous attempts mainly focused on the pipeline (coarse-to-fine) paradigm [24,36]: first locating or generating the evidence sentences corresponding to the question by an evidence extractor or generator, then the answer is predicted based on it. Unfortunately, such a pipeline paradigm suffers from the problem

of error accumulation. Besides, in real-world scenarios, the evidence supporting the answer to the question is often implicitly present in the document and thus not easily extracted or generated. For instance, 46% of the evidence sentences could not be explicitly found in the documents in the ReCO dataset. Once the evidence extractor or generator gets incorrect evidence, the result obtained by the answer predictor is bound to be wrong.

In this paper, we attempt to engage the model to focus more on the evidence sentences in the document rather than extracting them out. Thus we propose a **M**ulti-**S**trategy **K**nowledge **D**istillation based **T**eacher-**S**tudent framework (**MSKDTS**). In the training phase, we first take evidence and document as the reference information to pretrain a teacher model and a student model, respectively. Then, we incorporate multi-strategy knowledge distillation into the teacher-student framework, which is the student model attempts to produce teacher-like features and predicted answers through feature knowledge distillation and prediction knowledge distillation. Subsequently, in the testing phase, the enhanced student model predicts the answer with only the document (unaware of the evidence sentences). Hence, the whole process obviates the process of explicitly evidence extraction, which naturally circumvents the accumulation of errors in the conventional pipeline paradigm.

Our contributions are summarized as follows:

- We propose a teacher-student framework for MRC to address the issue of irrelevant information in reference documents causing a negative impact on answer inference.
- We propose a multi-strategy knowledge distillation approach in the teacher-student framework, which transfers knowledge from the teacher model to the student model at feature level and prediction level through feature knowledge distillation and prediction knowledge distillation.
- We conducted experiments on the two testing sets of the ReCO dataset, the results demonstrate the effectiveness of our approach, and further ablation experiments prove the effectiveness of both knowledge distillation strategies.

2 Related Work

2.1 Machine Reading Comprehension

The task of machine reading comprehension (MRC) can well indicate the ability of the machine to understand texts. Owing to the rapid development of deep learning and the presence of many large-scale datasets, MRC is under the spotlight in the field of natural language processing (NLP) in recent years. Depending on the format of questions and answers, the MRC datasets can be roughly categorized into cloze-style [10,11], multiple-choice [16,30,35], span prediction [15,28], and free form [9,23]. Lately, new tasks have emerged for MRC, such as knowledge-based MRC [25], MRC with unanswerable questions [13,27,32] and multi-passage MRC [37].

To model human reading patterns, pipeline (coarse-to-fine) paradigm have been proposed [2,19]. These models first extract the corresponding evidence from the document and then predict the answer via such evidence. To train a evidence extractor, current methods are mainly unsupervised methods [14,31], weakly supervised methods [22] and reinforcement learning methods [2]. Besides, Niu et al. [24] proposed a self-training method for MRC with soft evidence extraction, which performs great on several MRC tasks. Moreover, there are supervised methods [8,20] for extractive MRC by automatically generating evidence, which can be adopted in non-extractive MRC by first generating the evidence, then predicting the answer based on it. Last, Wang et al. [35] presents ReCO, a multiple-choice dataset which manually annotated the evidence in the document, which allows training the evidence extractor or generator in a supervised manner.

In order to engage the model focus more on the evidence, while excluding the pipeline paradigm that inevitably leads to error accumulation, we propose a end-to-end teacher-student framework in this paper.

2.2 Knowledge Distillation

Knowledge distillation [12] is an effective means of transferring knowledge over from one model to another by mimicking the outputs of the original model. Knowledge Consolidation Network [1] is proposed to address the problem of catastrophic forgetting in the incremental event detection task by utilizing the knowledge distillation method. To deploy huge neural machine translation models on edge devices, Wu et al. [38] combined layer-level supervision into the intermediate layers of the original knowledge distillation framework. To cope with the problem of performance degradation caused by utilizing lifelong language learning on different tasks, Chuang et al. [3] proposes an approach that assigns the teacher model to first learn the new task and then passes the knowledge to the lifelong language learning model via knowledge distillation.

Adversarial feature learning [6] is a method that renders the student model with comparable feature extraction ability to the teacher model via Generative Adversarial Networks (GANs, Goodfellow et al. [7]). In order to tackle the major challenge faced in event detection, namely ambiguity in natural language expressions, Liu et al. [21] proposed an adversarial imitation based knowledge distillation approach to learn the feature extraction ability from the teacher model. Lample et al. [17] adopts adversarial feature learning to align features extracted from different language auto-encoders for unsupervised neural machine translation.

In our work, we incorporate multi-strategy knowledge distillation (feature level with adversarial feature learning and prediction level) into the teacher-student framework, bringing more attention to the evidence.

3 Methods

Figure 2 demonstrates the overall framework of MSKDTS, which aims to cope with the irrelevant information in documents. MSKDTS is composed of three

major parts, namely, the teacher model, the student model and the knowledge distillation strategies. Documents and evidence sentences are concatenated with queries and candidate answers respectively as the input to the teacher model and the student model. Following encoding the input sequences and predicting the answers, we utilize knowledge distillation to align the features and predictions of the student model to the teacher model.

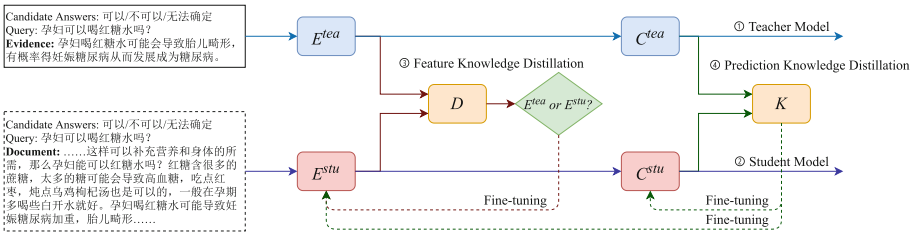

Fig. 2. The overall framework of MSKDTS. The model is composed of six component: the teacher encoder E^{tea}, the student encoder E^{stu}, the discriminator D, the teacher classifier C^{tea}, the student classifier C^{stu} and the prediction knowledge distillation component K. In the training phase, we first take evidence and document as the input reference information to pretrain the teacher model and the student model. Next, E^{stu} and D compete with each other through adversarial imitation strategy. In addition, the probabilities of the answers predicted by C^{stu} and C^{tea} are aligned by a prediction knowledge distillation approach K. In the final testing phase, documents are used as input to the enhanced E^{stu} and C^{stu} for answer prediction.

3.1 MRC Model

Our teacher-student framework mainly oriented towards the multiple-choice MRC problem [16,30,35]. It is composed of a teacher model and a student model, both of which consisting a BERT encoder and a multi-class classifier.

BERT Based Encoder. E^{tea} and E^{stu} are implemented using BERT [4], a multi-layer bidirectional Transformer [34] encoder. Below illustrates several different elements of the input to the BERT encoder and their representations:

- **Document:** A N-token document contains several sentences, distinct parts of which describe different information, denoted as $X^d = \{w_1, w_2, \ldots, w_N\}$, where w_i denotes a word in the document.
- **Evidence:** As the most critical information in inferring the correct answer, the evidence is typically shorter than the document, denoting it by a M-token (where $M \leq N$) sequence as $X^e = \{w_1, w_2, \ldots, w_M\}$, where w_i denotes each word in the evidence sentences.
- **Query:** A L-token query is denoted as $X^q = \{w_1, w_2, \ldots, w_L\}$, where w_i is a word in the query.

– **Alternative Answers:** To predict the correct answer from candidate answers, we denote each candidate answer by A_i. We concatenate A_1 to A_U using [OPT] as the input to the encoder. Note that, in BERT encoder, we use a special token [unused1] as [OPT].

In accordance with the different inputs of the teacher model and the student model, we concatenate these elements above and encode them with the BERT encoder to obtain a context-sensitive representation for the input sequence:

1) For the teacher encoder (E^{tea}), we concatenate candidate answers, query and evidence as input by special tokens of BERT, obtaining an input representation with evidence sentences as reference information. In Fig. 3, we present an example of the input sequence for the teacher encoder.

2) For the student encoder (E^{stu}), analogous to the teacher encoder, except that the reference information is the document rather than the evidence sentence, i.e., candidate answers, query, and document.

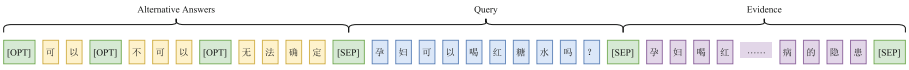

Fig. 3. The BERT input format of the teacher encoder (E^{tea}).

Multi-class Classifier. The softmax classifier is applied as our multi-class classifier, both C^{tea} and C^{stu}, which is used to predict the correct answer from the candidate answers. We take the output of the BERT encoder (E^{tea} and E^{stu}, respectively), i.e., the encoded features, as the input of the classifier. The hidden layer of [OPT] is used as the classification feature f^o for each candidate answer. The multi-class classifier takes these features as input and then computes a prediction probability for each candidate answers as output. The prediction probability $P(A|E, C)$ for each candidate answer is computed as:

$$P(A|E, C) = \text{softmax}(W^o \cdot f^o + b^o) \tag{1}$$

where E is E^{tea} or E^{stu}; C is C^{tea} or C^{stu}; A is the candidate answers; W^o and b^o are trainable parameters of the multi-class classifier (either C^{tea} or C^{stu}).

3.2 Knowledge Distillation Strategies

This section demonstrates in detail of the multi-strategy knowledge distillation in our model, which includes feature-level and prediction-level knowledge distillation, correspondingly, we build a discriminator D and a prediction distiller K. They differ in that the input to D is the extracted feature vector (i.e., the hidden layer of [OPT] mentioned in Sect. 3.1) obtained from either E^{tea} or E^{stu}, whereas the input to K is the logit of the prediction probability from C^{tea} or C^{stu}.

Feature Knowledge Distillation. We adopt an adversarial feature learning approach for feature level knowledge distillation, specifically, we apply a discriminator D, a multi-layer perception (MLP) based binary classifier. It takes the features obtained from E^{tea} and E^{stu} as the input then generates a probability P^D to distinguish the source of the input features. P^D is calculated as:

$$P^D = \text{sigmoid}(W^s(\tanh(W^x f^o + b^x)) + b^s) \tag{2}$$

where sigmoid$(*)$ is the activation function that maps a scalar to a float number between 0 and 1. A well-trained discriminator would output 1 for the features from E^{tea} and 0 for the features from E^{stu}. W^x, b^x, W^s, and b^s are trainable parameters of the discriminator. We use a two-layer MLP to enhance the representativeness of our discriminator.

The detailed training process of D will be elaborated in Sect. 3.3.

Prediction Knowledge Distillation. Apart from adversarial feature learning for feature level knowledge distillation, we propose a prediction level knowledge distillation. It enables the prediction probability of C^{stu} imitates those of C^{tea}, thereby improving its answer prediction ability. We adopt the knowledge distillation method proposed by Hinton et al. [12], whose specific approach in this framework is demonstrated in Sect. 3.3.

3.3 Overall Training Procedure

Our training process can be summarized into two phases, namely the pretraining phase and the fine-tuning phase. The overall training procedure is demonstrated in Algorithm 1.

Algorithm 1. The Overall Training Procedure

Input: Training Data (x, x^*, y)
1: Pretrain the teacher model (E^{tea},C^{tea}), the student model (E^{stu},C^{stu}), and the discriminator (D)
2: Freeze E^{tea} and C^{tea}
3: **repeat**
4: Freeze D
5: Unfreeze E^{stu} and C^{stu}
6: Updata E^{stu} and C^{stu} using Eq.8
7: **if** the remainder of the batch number to k is 0 **then**
8: Unfreeze D
9: Freeze E^{stu} and C^{stu}
10: Update D using Eq.5
11: **end if**
12: **until** convergence
Output: An enhanced student model

The Pre-trainig Phase. In the pre-training phase, we first train the teacher model and the student model using the evidence and the documents as reference information, respectively. Then the discriminator is trained using the outputs of E^{tea} and E^{stu}.

First, we train the teacher model (i.e., concatenating E^{tea} and C^{tea}) which is well aware of the evidence sentences. Its loss function is calculated as:

$$\mathcal{L}^{tea} = -\sum_{i=1}^{U} y_i \log(P(A_i|E^{tea}, C^{tea})) \tag{3}$$

where y_i is the label for the i-th answer A_i.

Then, the student model (i.e., concatenating E^{stu} and C^{stu}) is trained, which is not aware of the evidence, it takes the entire document instead of evidence sentences as reference information, thus implicitly introducing considerable irrelevant information. Its loss function is calculated as:

$$\mathcal{L}^{stu} = -\sum_{i=1}^{Y} y_i \log(P(A_i|E^{stu}, C^{stu})) \tag{4}$$

where y_i is the label for the i-th answer A_i.

In the final step, we keep the parameters of E^{tea} and E^{stu} unchanged to train the discriminator by treating the feature vector obtained from E^{tea} as positive examples (label 1) and those from E^{stu} as negative examples (label 0). In this process, the loss function of training the discriminator is calculated as:

$$\mathcal{L}^{D} = \max_{D} \mathbb{E}_{x\sim X}[\log(D(f^{o,tea}))] + \mathbb{E}_{x^*\sim X^*}[\log(1 - D(f^{o,stu}))] \tag{5}$$

where $f^{o,tea}$ is the features of the teacher encoder and $f^{o,stu}$ is the features of the student encoder.

The Fine-Tuning Phase. In the fine-tuning phase, we aim to enhance the feature extraction ability of E^{stu} and the answer prediction ability of C^{stu}, in other words, in document-only cases, we expect the encoder to ignore the irrelevant information as much as possible, focusing more on the evidence sentences.

To enhance the feature extraction ability of E^{stu}, we employ the pretrained D, which can well distinguish between E^{tea} and E^{stu}, to conduct adversarial training with E^{stu}. The loss of E^{stu} is computed as:

$$\mathcal{L}^{afl} = -y \log(D(f^{o,stu})) \tag{6}$$

where y is the label of the output of E^{stu} given to D during adversarial feature learning. Therefore, in order for E^{stu} to produce features similar to those produced by E^{tea}, we set $y = 1$, i.e., we expect the features extracted by E^{stu} to be recognized by D as those extracted by E^{tea}.

After k batches of fine-tuning E^{stu}, the accuracy of D decreases and fails to distinguish well between the outputs obtained from E^{stu} and E^{tea}, then we

retrain D using the same loss $\mathcal{L}^D$ as in the pretraining phase. Iteratively fine-tune E^{stu} as well as retrain D until the training process converges. The training procedure is shown in Algorithm 1.

As for prediction level knowledge distillation, the output logit of each sample of C^{tea} and C^{stu} are denoted as v and v^*, respectively. The prediction knowledge distillation is calculated as:

$$\begin{aligned} \mathcal{L}^{pkd} &= -\sum_{i=1}^{U} \tau_i(v^*) \log(\tau_i(v)) \\ \tau_i(v^*) &= \frac{e^{v_i^*/\Omega}}{\sum_{j=1}^{U} e^{v_j^*/\Omega}},\ \tau_i(v) = \frac{e^{v_i/\Omega}}{\sum_{j=1}^{U} e^{v_j/\Omega}} \end{aligned} \tag{7}$$

where Ω is a hyper-parameter, which is usually set to be greater than 1 (e.g. $\Omega = 2$) in our experiments to increase the weights of small values; U is the number of classes; $\mathcal{L}^{pkd}$ is designed to encourage the prediction of the student model to match the prediction of the teacher model.

In the fine-tuning phase, the total loss of the student model is:

$$\mathcal{L}^{stu\_all} = \mathcal{L}^{stu} + \alpha \mathcal{L}^{afl} + \beta \mathcal{L}^{pkd} \tag{8}$$

where α and β are two hyper-parameters. If α and β are very large, the model will focus more on learning knowledge from the teacher model, rather than the ground-truth labels. Noting that, the parameters of D, E^{tea}, C^{tea} are kept unchanged while fine-tuning the components of the student model.

After completing the two knowledge distillation approaches above, we obtained an enhanced student model that has successfully learned the knowledge of the teacher model and is able to predict accurate answers using only the documents as reference information.

4 Experiments

4.1 Datasets

We conduct experiments on a recently proposed MRC dataset, ReCO [35] to evaluate the validity of our model. To the best of our knowledge, this is the only large-scale multiple-choice MRC dataset with manually labeled evidence. ReCO contains 300k document-query pairs, each of them is manually labeled with evidence. It is worth noting that, during the annotation process, for 46% samples, the annotators paraphrase or highly summarize the key sentences according to their understanding, resulting in a situation that not all evidence sentences can be found in its corresponding document.

In ReCO, three candidate answers are available for each query. In order to obtain the correct answer, strong inference capability of the model is required. The dataset contains 280k training samples and 20k test samples, which are further divided into testing set A (Test_A) and testing set B (Test_B). Test_B is the complement to Test_A in terms of quantity, and can certify the validity of the model more adequately.

4.2 Baseline

To evaluate the capability of MRC models and select well-performing teacher and student models in our framework, we adopt several strong baselines that perform well on many MRC tasks:

BiDAF [31]: BiDAF uses LSTM as its encoder, and models the relationship between the question and the answer by a bidirectional attention mechanism.

BiDAF* [26,31]: BiDAF* replaces the traditional word embedding in BiDAF with ELMO (a language model trained on unsupervised data), which yields better results.

BERT [4]: A multi-layer bidirectional Transformer, which is pretrained on large unlabeled data, has outperformed state-of-the-art models in many NLP tasks.

ALBERT [18]: ALBERT is an improved version of BERT, which reduces the overall number of parameters, speeds up the training process, and is better than BERT in many aspects.

Since the evidence sentences in the 46% samples in the ReCO dataset could not be explicitly found in the corresponding documents, we use generation models as evidence generators in the pipeline baselines instead of extraction models.

Enc2Dec [33]: We designed a coarse-to-fine framework based on the encoder-decoder framework. This model encodes documents with an LSTM encoder and then generates evidence by an LSTM decoder.

Enc2Dec* [33]: In addition to the Enc2Dec model, we adopt the BERT encoder in the encoder-decoder framework.

4.3 Experimental Setup and Evaluation Metrics

We use ALBERT-base from HuggingFace's Transformer library[1] as the encoder for our MRC model. For both teacher model and student model as well as the discriminator D, the learning rate is set to 2e−5, batch size set to 4, hyper-parameters α and β are chosen from [0–100], specified as $\alpha = 0.5$ and $\beta = 20$, with temperature coefficient $\Omega = 2$. Since D can easily learn and distinguish the features obtained from different encoders, we randomly sample 10,000 training samples each time to train D. We retrain the discriminator every $k = 3000$ batches. All hyper-parameters are obtained by grid search in the validation process.

Following the previous work [35], we use accuracy as our metric to evaluate whether each sample is correctly classified.

4.4 Results

We list the following Research Questions (RQ) as guidelines for experimentation in our work:

[1] https://huggingface.co/.

- RQ1: How well did the MRC models perform before incorporating the knowledge distillation strategies, and which model we select to be the teacher or student model?
- RQ2: Is there a significant improvement in performance after applying our proposed MSKDTS framework, and does the MSKDTS framework outperform the traditional pipeline paradigm?
- RQ3: Whether the feature knowledge distillation we designed can effectively improve the performance by enabling E^{stu} to imitate the output features of E^{tea}?
- RQ4: Will the prediction knowledge distillation strategy we employ be effective in improving the performance of the student model?

Teacher and Student Models. To answer RQ1, this section shows the performance of several baseline models when inputting documents or evidence as reference information, and compares the performance of different baselines to select the teacher and student models in the MSKDTS framework.

Table 1. Experimental results on the development set (Dev), testing set A (Test_A) and testing set B (Test_B) of the ReCO dataset. The second/third column shows the result of these models when taking evidence/documents as inference information input in both training and testing phases. Bold indicates the best model. BERT_b and BERT_l denotes BERT base and BERT large, respectively. $\text{ALBERT}_\text{tiny}$ and ALBERT_b denotes ALBERT tiny and ALBERT base, respectively.

	Teacher (Evidence)			Student (Document)		
	Dev	Test_A	Test_B	Dev	Test_A	Test_B
Random [35]	33.3	33.3	33.3	33.3	33.3	33.3
BiDAF [31]	68.9	68.3	67.9	55.7	55.8	56.1
BiDAF* [26,31]	70.3	70.9	71.1	58.4	58.9	58.6
BERT_b [4]	73.8	73.4	72.8	61.4	61.1	62.0
BERT_l [4]	76.3	77.0	76.4	65.5	65.3	65.8
$\text{ALBERT}_\text{tiny}$ [18]	70.9	70.4	71.3	63.1	62.7	62.4
ALBERT_b [18]	**77.2**	**77.6**	**77.0**	**68.2**	**68.4**	**69.1**
Human	-	91.5	-	-	88.0	-

Table 1 shows the performance of several baseline models when evidence and documents are used as reference information input, respectively. Comparing the results in Table 1, we can see that irrelevant information in the documents does have a negative effect on answer prediction (for every model except random, the performance with evidence as reference information input is superior to the performance with documents as reference information input), so evidence has a facilitating effect on answer prediction. Also, the results in Table 1 show that there is a gap between predicting answers by documents and by evidence even

for human, which proves the importance of evidence in machine reading comprehension.

From the results, we can see that ALBERT_b achieves the best performance and can outperform other BERT-based models when taking evidence and document as reference information input, therefore, we choose ALBERT_b as both our teacher and student models.

The teacher model outperforms the student model by 9.2% and 7.9% on Test_A and Test_B, respectively. Since the student model is designed to imitate the behavior of the teacher model in our approach, the performance of the student model cannot exceed that of the teacher model, i.e., the performance of the teacher model is the upper bound of our framework, and the lower bound should be the student model without fine-tuning.

Results on Real Test Scenarios. To answer RQ2, we compared our approach with the pipeline paradigm and the teacher model and student model (which is not fine-tuned with our knowledge distillation strategies) as upper and lower bounds.

Table 2. Experimental results on the development set (Dev), testing set A (Test_A), testing set B (Test_B). Enc2Dec(*) + ALBERT_b is MRC models with evidence generator (pipeline paradigm).

	Dev	Test_A	Test_B
Lower bound	68.2	68.4	69.1
Enc2Dec + ALBERT_b	68.6	68.9	69.3
Enc2Dec* + ALBERT_b	68.9	69.0	69.6
MSKDTS (Ours)	**71.3**	**71.0**	**70.8**
Upper bound	77.2	77.6	77.0

To validate the effectiveness of MSKDTS, we tested the performance of the enhanced student model in real scenarios. In real scenarios, the evidence in the documents is not annotated, and the enhanced student model needs to predict the results directly based on the documents as reference information.

From the experimental results in Table 2, we can see that:

First, our student model achieves the best performance, outperforming all the baseline models that do not use evidence. This demonstrates that the multi-strategy knowledge distillation approach we proposed enables the student encoder to effectively imitate the output features of the teacher encoder, can focus on the evidence sentences in the documents.

Second, to compare with the pipeline model, we train an encoder-decoder model that generates the evidence sentences for each testing sample. The performance of Enc2Dec and Enc2Dec* on the two testing sets is much weaker than

our fine-tuned student model. Our model outperforms Enc2Dec* by 2.0% and 1.2% on Test_A and Test_B, respectively.

Third, there is still a gap between our approach and the teacher model, which shows that it is still a significant challenge of how to engage the model to focus on the evidence sentences from the documents.

The Effect of Feature Knowledge Distillation. To answer RQ3, we study the effect of feature knowledge distillation in this section.

Table 3. Experimental results on the development set (Dev), testing set A (Test_A) and testing set B (Test_B) to verify the effect of feature knowledge distillation.

	Dev	Test_A	Test_B
MSKDTS	**71.3**	**71.0**	**70.8**
MSKDTS (k = 10000)	71.0	70.5	70.6
MSKDTS (k = 50000)	70.6	70.3	70.2
MSKDTS-AFL+COSINE	70.7	70.6	70.4
MSKDTS-AFL	70.2	70.1	69.8

We conducted three experiments to demonstrate the effectiveness of adversarial feature learning as a feature knowledge distillation strategy: 1) Testing the performance of the MSKDTS framework with different hyper-parameters (when the update frequency $k = 10000$ and $k = 50000$ of the discriminator D). 2) Testing the performance of the MSKDTS framework replacing adversarial feature learning with the cosine similarity loss between the features of the teacher model and those of the student model (denoted as MSKDTS-AFL+COSINE), which enables the two features to have the same angle in the high dimensional space. 3) Testing the performance of the MSKDTS framework without any feature knowledge distillation strategies (denoted as MSKDTS-AFL). Table 3 shows the results of these models.

From these results, we can see that: 1) Our approach outperforms all variants, which proves the effectiveness of our feature knowledge distillation strategy. 2) For $k = 10000$ and $k = 50000$, the performance degradation is caused by poor discriminator performance due to updating the discriminator after a larger number of batches. 3) The cosine similarity loss causes a performance degradation with -0.4% and -0.4% on Test_A and Test_B due to features learned based on specific distances is prone to be approximated from a certain aspect (angle) in the high dimensional space, which may result in a loss of semantic information. 4) In the absence of feature knowledge distillation, the performance degrades significantly due to the lack of proximity to the features of the teacher model.

The Effect of Prediction Knowledge Distillation. To answer RQ4, we study the effect of the prediction knowledge distillation in this section.

Table 4. Experimental results on the development set (Dev), testing set A (Test_A), testing set B (Test_B) to verify the effect of prediction knowledge distillation.

	Dev	Test_A	Test_B
MSKDTS	**71.3**	**71.0**	**70.8**
MSKDTS-PKD+KL	70.9	70.5	70.3
MSKDTS-PKD	69.7	69.9	70.1

As shown in Table 4, we use KL-divergence (MSKDTS-PKD+KL) to replace the knowledge distillation loss. This variant causes performance degradation due to the absence of the temperature coefficient Ω in the knowledge distillation loss. Therefore, it is difficult for the model to learn small logit values and affects the knowledge distillation ability. Compared to the model without Prediction Knowledge Distillation, our model achieves significant improvement. It demonstrates the effectiveness of prediction knowledge distillation. Compared to the student model trained with document only (without fine-tuning), it verifies that the simultaneous use of feature knowledge distillation and prediction knowledge distillation can effectively improve the performance.

5 Conclusion

We propose a **M**ulti-**S**trategy **K**nowledge **D**istillation based **T**eacher-**S**tudent framework (**MSKDTS**) for MRC to address the challenges posed by irrelevant information in documents for answer prediction. The teacher-student framework naturally circumvents the error accumulation problem in the traditional pipeline paradigm and the knowledge distillation strategies enhance the model capability at the feature and prediction levels. Experiments on the ReCO dataset demonstrate the effectiveness of our approach.

Acknowledgements. This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106400), the National Natural Science Foundation of China (No. 61922085, 61976211, 61632020, U1936209 and 62002353) and Beijing Natural Science Foundation (No.4192067). This work is also supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0301), the Key Research Program of the Chinese Academy of Science (Grant No. ZDBS-SSW-JSC006), the independent research project of National Laboratory of Pattern Recognition, the Youth Innovation Promotion Association CAS and Meituan-Dianping Group.

References

1. Cao, P., Chen, Y., Zhao, J., Wang, T.: Incremental event detection via knowledge consolidation networks. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November, 2020, pp. 707–717 (2020)

2. Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A., Berant, J.: Coarse-to-fine question answering for long documents. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July – 4 August, Volume 1: Long Papers, pp. 209–220 (2017)
3. Chuang, Y., Su, S., Chen, Y.: Lifelong language knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November, 2020, pp. 2914–2924 (2020)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
5. Dhingra, B., Mazaitis, K., Cohen, W.W.: Quasar: datasets for question answering by search and reading. CoRR abs/1707.03904 (2017)
6. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April, 2017, Conference Track Proceedings (2017)
7. Goodfellow, I.J., et al.: Generative adversarial networks. CoRR abs/1406.2661 (2014)
8. Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Schiller, B., Schulz, C., Gurevych, I.: Ukp-athene: multi-sentence textual entailment for claim verification. CoRR abs/1809.01479 (2018)
9. He, W., et al.: DuReader: a Chinese machine reading comprehension dataset from real-world applications. In: Proceedings of the Workshop on Machine Reading for Question Answering@ACL 2018, Melbourne, Australia, 19 July, 2018, pp. 37–46 (2018)
10. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, 7–12 December, 2015, Montreal, Quebec, Canada, pp. 1693–1701 (2015)
11. Hill, F., Bordes, A., Chopra, S., Weston, J.: The goldilocks principle: Reading children's books with explicit memory representations. In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May, 2016, Conference Track Proceedings (2016)
12. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR abs/1503.02531 (2015)
13. Hu, M., Wei, F., Peng, Y., Huang, Z., Yang, N., Li, D.: Read + verify: machine reading comprehension with unanswerable questions. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January – 1 February, 2019, pp. 6529–6537 (2019)
14. Huang, H., Choi, E., Yih, W.: Flowqa: Grasping flow in history for conversational machine comprehension. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May, 2019 (2019)
15. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: Triviaqa: a large scale distantly supervised challenge dataset for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July – 4 August, Volume 1: Long Papers, pp. 1601–1611 (2017)

16. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.H.: RACE: large-scale reading comprehension dataset from examinations. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September, 2017, pp. 785–794 (2017)
17. Lample, G., Conneau, A., Denoyer, L., Ranzato, M.: Unsupervised machine translation using monolingual corpora only. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April – 3 May, 2018, Conference Track Proceedings (2018)
18. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April, 2020 (2020)
19. Li, W., Li, W., Wu, Y.: A unified model for document-based question answering based on human-like reading strategy. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, 2–7 February, 2018, pp. 604–611 (2018)
20. Lin, Y., Ji, H., Liu, Z., Sun, M.: Denoising distantly supervised open-domain question answering. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July, 2018, Volume 1: Long Papers, pp. 1736–1745 (2018)
21. Liu, J., Chen, Y., Liu, K.: Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January – 1 February, 2019, pp. 6754–6761 (2019)
22. Min, S., Zhong, V., Socher, R., Xiong, C.: Efficient and robust question answering from minimal context over documents. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July, 2018, Volume 1: Long Papers, pp. 1725–1735 (2018)
23. Nguyen, T., et al.: MS MARCO: a human generated machine reading comprehension dataset. In: Proceedings of 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, 9 December, 2016, vol. 1773 (2016)
24. Niu, Y., Jiao, F., Zhou, M., Yao, T., Xu, J., Huang, M.: A self-training method for machine reading comprehension with soft evidence extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July, 2020, pp. 3916–3927 (2020)
25. Ostermann, S., Modi, A., Roth, M., Thater, S., Pinkal, M.: MCScript: a novel dataset for assessing machine comprehension using script knowledge. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, 7–12 May, 2018 (2018)
26. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June, 2018, Volume 1 (Long Papers), pp. 2227–2237 (2018)

27. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: unanswerable questions for squad. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July, 2018, Volume 2: Short Papers, pp. 784–789 (2018)
28. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–4 November, 2016, pp. 2383–2392 (2016)
29. Reddy, S., Chen, D., Manning, C.D.: CoQA: a conversational question answering challenge. Trans. Assoc. Comput. Linguist. **7**, 249–266 (2019)
30. Richardson, M., Burges, C.J.C., Renshaw, E.: MCTest: a challenge dataset for the open-domain machine comprehension of text. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18–21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 193–203 (2013)
31. Seo, M.J., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April, 2017, Conference Track Proceedings (2017)
32. Sun, F., Li, L., Qiu, X., Liu, Y.: U-net: machine reading comprehension with unanswerable questions. CoRR abs/1810.06638 (2018)
33. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, 8–13 December, 2014, Montreal, Quebec, Canada, pp. 3104–3112 (2014)
34. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017)
35. Wang, B., Yao, T., Zhang, Q., Xu, J., Wang, X.: Reco: a large scale Chinese reading comprehension dataset on opinion. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February, 2020, pp. 9146–9153 (2020)
36. Wang, H., et al.: Evidence sentence extraction for machine reading comprehension. In: Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, 3–4 November, 2019, pp. 696–707 (2019)
37. Wang, Y., et al.: Multi-passage machine reading comprehension with cross-passage answer verification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July, 2018, Volume 1: Long Papers, pp. 1918–1927 (2018)
38. Wu, Y., Passban, P., Rezagholizadeh, M., Liu, Q.: Why skip if you can combine: a simple knowledge distillation technique for intermediate layers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November, 2020, pp. 1016–1021 (2020)