frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 30 May 2025
DOI 10.3389/frai.2024.1308206

OPEN ACCESS

EDITED BY
Michal Ptaszynski,
Kitami Institute of Technology, Japan

REVIEWED BY
Savaş Yıldırım,
Toronto Metropolitan University, Canada
Eric Nichols,
Honda Research Institute Japan Co., Ltd., Japan

*CORRESPONDENCE
Xin Huang
✉ xinhuang@jxnu.edu.cn

RECEIVED 09 October 2023
ACCEPTED 29 October 2024
PUBLISHED 30 May 2025

CITATION
Huang X, Song H and Lu M (2025) Small pre-trained model for background understanding in multi-round question answering. *Front. Artif. Intell.* 7:1308206. doi: 10.3389/frai.2024.1308206

Small pre-trained model for background understanding in multi-round question answering

Xin Huang[1]*, Hulin Song[2] and Mingming Lu[3]

[1]Software College, Jiangxi Normal University, Nanchang, China, [2]School of International Economics and Trade, Jiangxi University of Finance and Economics, Nanchang, China, [3]Department of Computer Science and Technology, Tongji University, Shanghai, China

Multi-round Q&A based on background text needs to infer the answer to the question through the current question, historical Q&A pairs, and background text. The pre-trained model has proved its effectiveness in this task; however, the existing model has many problems such as too many parameters and high resource consumption. We propose a knowledge transfer method that combines knowledge distillation, co-learning of similar datasets, and fine-tuning of similar tasks. Through multi-knowledge cooperative training from large model to small model, between different data sets, and between different tasks, the performance of the small model with low resource consumption can match or surpass that of the large model.

KEYWORDS

multi-round Q&A, knowledge transfer, background understanding, knowledge distillation, model compression

1 Introduction

Background-oriented text question answering (Q&A) studies (Minaee et al., 2021; Li et al., 2022; Cui et al., 2022; Huang et al., 2023b) derived from machine-reading comprehension tasks represented by SQuAD (Rajpurkar et al., 2016, 2018) are gaining attention. Considering a background text fragment and a question associated with it, we try to determine an answer to a question based on the background text or mark that the answer does not exist. Although multiple issues with the background text exist, none of them are related. However, in a real environment, Q&A is a multi-round and continuous process, and the questions are not independent as students may ask questions to teachers on random topics of interest (Stede and Schlangen, 2004; Huang et al., 2023c). Because numerous coreferences and ellipses are used in multi-round of Q&A to achieve concision and efficiency, question comprehension should consider the current question content and combine historical Q&A pairs to determine the object of pronoun reference and ellipsis content to understand the current question in detail. The challenge of background comprehension in multi-round Q&A is the deduction of an answer to a question based on the current question, historical Q&A pairs, and background text (Martinez-Gil, 2023; Cui et al., 2023; Shao et al., 2023).

To promote studies on multi-round Q&A considering background and evaluate the validity of the Q&A model, a few datasets have been published, such as CoQA (Reddy et al., 2019) and QuAC (Choi et al., 2018). Each of these datasets involves multiple rounds of Q&A around a single piece of background text; both datasets include two speakers (questioner and responder) and allow for unanswerable questions. Table 1 lists a complete multi-round Q&A based on the CoQA dataset. The background relates to a CNN news report divided into two sections. According to the conversion method proposed by

TABLE 1 Q&A example of CoQA.

(CNN)–American journalist Michael Scott Moore, held for more than 2 years by Somali pirates, has been freed, Moore's family and a Somali official told CNN on Tuesday.	
"We are just elated," Marlis Saunders, Moore's mother, said in a brief conversation. "It took a lot of work for us to get this point. And to hear he is free—just joyful, I can't describe it"	
Q_1: Who was held for two years?	Q_6: Who?
A_1: Michael Scott Moore	A_6: Marlis Saunders
Q_2: What happened to him?	Q_7: Who was that?
A_2: freed	A_7: Moore's mother
Q_3: From who?	Q_8: How Did she feel?
A_3: Somali pirates	A_8: elated
Q_4: After how long?	Q_9: Why?
A_4: More than 2 years	A_9: To hear he is free
Q_5: Did anyone feel about this?	Q_{10}: How does she describe it?
A_5: Yes	A_{10}: She can't.

Yatskar (2019), CoQA answers can be converted into four types: (1) SPAN representing the background text fragment, (2) affirmative YES, (3) negative NO, and (4) UNANS that cannot be answered.

With an increase in the number of pre-training models (Qiu et al., 2020; Yu et al., 2022; Gou et al., 2021), corresponding studies have introduced pre-training models for new studies (Singh et al., 2021; Gu et al., 2021; Kandpal et al., 2023; Lauriola et al., 2022). From the beginning, it was a supplement to the traditional word vector (Yatskar, 2019), which is regarded as a downstream task of the pre-training model, and the model performance was improved by adding word-embedding information (Qu et al., 2019), adjusting the output layer structure (Yeh and Chen, 2019), and improving the training objectives (Ju et al., 2019; Garg and Moschitti, 2021; McCarley et al., 2020; Chen et al., 2021; Yang et al., 2020). We explored a small pre-training model suitable for Q&A background awareness to ensure that the performance of the small model matches or surpasses that of the large model and achieves an effective environment of resource consumption and performance. We proposed a model based on knowledge transfer—KTM. Knowledge transfer is divided into two stages: "preparation" and "learning." First, three fine-tuned large and small models were obtained during the preparation phase. The fine-tuning of the small model was performed on the machine-reading comprehension dataset, that is, SQuAD, which allowed the small model to learn knowledge on similar tasks. Next, in the learning stage, knowledge distillation was applied, and CoQA and QuAC datasets were combined to learn together. The small model learns the knowledge owned by the large model; in contrast, the two background understanding datasets learn from each other and complement each other. Comparative experiments show that compared with the large model, KTM has fewer parameters, lower memory usage, significantly shorter training time, and faster prediction speed, while maintaining the excellent background understanding ability of the large model. The contributions of this study are as follows.

1. The computational time and performance differences in the background understanding task of the automatic Q&A of mainstream pre-training models were compared.

2. A small-scale pre-training model for the background understanding of Q&A was proposed. Knowledge distillation, co-learning of similar datasets, fine-tuning of similar tasks, and other strategies were comprehensively used to achieve various types of knowledge transfers to ensure that the performance of small models with low resource consumption matched or exceeded that of large models.

3. Abundant validation experiments were performed to demonstrate the effectiveness of knowledge transfer. Experimental results on the QuAC validation set indicate that the performance of the knowledge transfer model exceeds that of the large model.

2 Related work

While the foundational work by Hinton et al. (2015) on knowledge distillation provides a basic framework, recent studies have applied knowledge distillation techniques specifically within QA systems (Gou et al., 2021). For instance, Izacard and Grave (2020) demonstrated how distilling knowledge from a reader to a retriever enhances the efficiency of open-domain QA systems. Moreover, Yang et al. (2019b) focused on model compression within large-scale QA systems through multi-task knowledge distillation. Unlike these studies, our approach uniquely optimizes small model performance in a multi-round QA setting by integrating knowledge distillation with co-learning, a combination less explored in prior research.

Considering model performance and computing resources, the traditional background understanding model represented by BiDAF++ w/ *n*-ctx demonstrates lower memory consumption, short training and prediction time, and poor performance. The pre-training model can considerably improve the performance of background understanding; however, its large number of parameters increases resource consumption and renders it difficult to deploy. In addition, the parameters of large and small models are not proportional to performance. For example, XLNet (Yang et al., 2019a) has numerous parameters but does not perform as well as ALBERT (Lan et al., 2020). The current trend of pursuing large models is worrisome for the environment (Schwartz et al., 2019), and performance can be improved only by consuming a large number of computing resources (Wu et al., 2022; Chang et al., 2022; Huang et al., 2023a).

Table 2 compares the application of different models to the CoQA dataset from three aspects: parameter quantity, computation time, and performance. Among the five pre-training models, BERT (Devlin et al., 2019) was first proposed, and the following four models, XLNet, RoBERTa (Liu et al., 2019), ALBERT, and DistilBERT (Sanh et al., 2019), are all improvements to BERT.

1. The parameter quantity of BiDAF++ w/ *n*-ctx is much smaller than that of the pre-trained model; the former is ∼1/10 of the latter. As lightweight pre-training models, ALBERT-base and

TABLE 2 Trainable parameters of traditional and pre-trained models in CoQA.

Model	Trainable Param	Round (h)	Predict (min)	Validation F1
BiDAF++ w/ *n*-ctx	2M	1.12	5.75	69.2
BERT-base	110M	11.17	16.33	80.5
XLNet-base	117M	50.08	23.67	78.8
RoBERTa-base	125M	11.42	16.17	80.2
ALBERT-base	12M	10.67	10.18	80.6
DistilBERT-base	66M	5.50	8.02	75.0

The training and prediction time is the time required to complete the corresponding operation of the model considering a single Tesla K40m graphics card. Among them, Round (h) represents the time required for one round of training, in hours, and Predict (min) represents the time required for prediction, in minutes.

DistilBERT-base have 90 and 40% fewer parameters than BERT-base, respectively.

2. BiDAF++ w/ *n*-ctx consumes less time than the pre-training model owing to its smaller number of parameters, and the maximum difference is ∼50 times. Remarkably, the training time of ALBERT-base was close to that of BERT-base. This implies that the smaller the number of parameters, the smaller the time consumption; however, ALBERT-base does not conform to this rule. ALBERT reduces the number of parameters but not the number of calculations. DistilBERT-base complies with the aforementioned rules, and its training and prediction times are reduced by 50%, which is aligned with the reduction in the number of transformer coding layers by 50%.

3. In terms of performance, the advantages of pre trained models have been fully demonstrated. The F1 values of BERT-base, RoBERTa-base, and ALBERT-base differ by more than 10 compared to BiDAF++ w/ *n*-ctx. Even DistiBERT base, which has the worst performance among the five pre-trained models, has an F1 value difference of 5.8.

In summary, the larger the model, the better the balance between deep model performance and resource consumption, which is the aim of this study.

3 Approach

3.1 Task definition

Given a background text of length m $B = \{b_1, b_2, \ldots, b_m\}$, current question Q_i $(i \geqslant 1)$, and historical Q&A pair $\{Q_1; A_1; Q_2; A_2; \ldots; Q_{i-1}; A_{i-1}\}$, the goal of the background understanding of the Q&A system is to generate the answer A_i of the question Q_i, which requires the type of A_i t to be SPAN, YES, NO, or UNANS. In particular, the SPAN type requires that the answer must be a span of the background text, that is, $A_i = \{b_j\}_{j=k}^{l} (1 \leqslant k \leqslant l \leqslant m)$, and must satisfy $l - k \leqslant n$, n is the maximum allowed length of the answer, and different datasets have different values. If the answer is to the other three types, A_i is determined by the dataset. For example, UNANS answers are represented by "unknown" and "CANNOTANSWER" in CoQA and QuAC, respectively.

3.2 Pre-trained models in background understanding

Figure 1 shows the background understanding model based on BERT or DistilBERT. As the main difference between BERT and DistilBERT is the number of transformer coding layers, and there is no difference between the input and output, a uniform purple box is used to represent the two models in the figure, which ignores differences in the internal structure of the models.

The input to the model comprises two concatenated sequences: s_1 and s_2 of length N and M, respectively. Sequence s_1 includes two parts: the historical Q&A pair and current question.

$$s_1 = \{[Q]; Q_1; [A]; A_1; \ldots; [Q]; Q_{i-1}; [A]; A_{i-1}; [Q]; Q_i\}, \quad (1)$$

where [Q] and [A] are special words that mark question and answer sentences, respectively, and are located at the beginning of the sentence. As question Q_1 has no historical Q&A pairs, $s_1 = \{[Q]; Q_1\}$. Sequence s_2 includes background B.

$$s_2 = B. \quad (2)$$

The two sequences are spliced together before inputting BERT or DistilBERT: $\{[CLS]; s_1; [SEP]; s_2; [SEP]\}$. Here, [CLS] is used to calculate the probability of the answer type, and [SEP] is used to segment the sequence. After entering the model, first, each word was converted into a vector, which was obtained by adding three parts: word embedding, segment embedding, and position embedding (DisitlBERT has no segment embedding). Segment embedding indicates whether the word belongs to s_1 or s_2, and positional embedding indicates the position of the word in the input sequence. Subsequently, after multi-layer transformer encoding, the latent vector of the last layer of each word was used as the output of the pre-training model, which was recorded as $\boldsymbol{T} \in \mathbb{R}^{d \times L}$. Based on $\boldsymbol{T}$, we calculated the probability of the start position k and the end position l of the segment when the answer is of type SPAN, and the probability of the answer is of types YES, NO, and UNANS. In particular, the calculation methods for the probability distributions p^k and $p^l \in \mathbb{R}^L$ of k and l are as follows.

$$p^k = softmax(\boldsymbol{w}_1^T \boldsymbol{T} + b_1), and \quad (3)$$

$$p^l = softmax(\boldsymbol{w}_2^T \boldsymbol{T} + b_2), \quad (4)$$

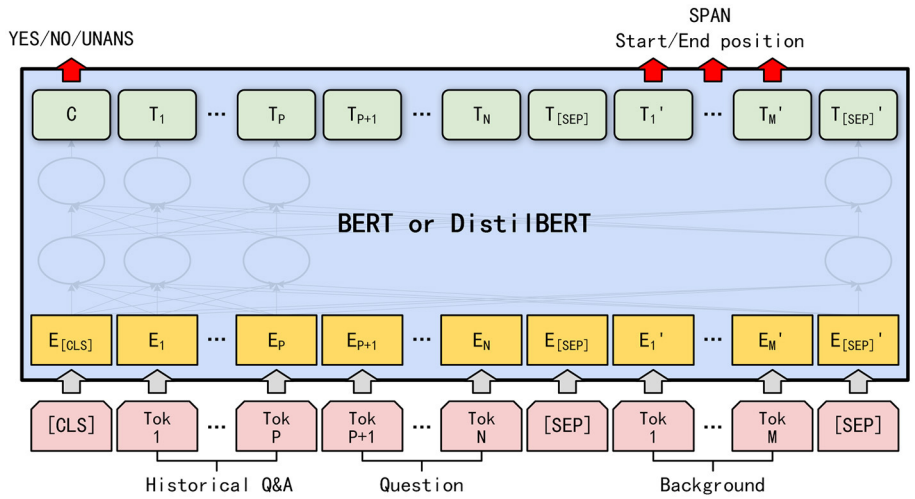

FIGURE 1
Background understanding based on pre-trained models.

where $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^d$, $b_1, b_2 \in \mathbb{R}$ are to-be-trained parameters. The probability distribution $p^t \in \mathbb{R}^3$ of the three types of answers, YES, NO, and UNANS, is calculated as follows.

$$p^t = softmax(\boldsymbol{w}_4^T \tanh(\boldsymbol{w}_3^T \boldsymbol{C} + \boldsymbol{b}_3) + \boldsymbol{b}_4), \tag{5}$$

where $\boldsymbol{C} \in \mathbb{R}^d$ is the hidden vector of [CLS], and $\boldsymbol{w}_3 \in \mathbb{R}^{d\times d}$, $\boldsymbol{b}_3 \in \mathbb{R}^d$, $\boldsymbol{w}_4 \in \mathbb{R}^{d\times 3}$, $\boldsymbol{b}_4 \in \mathbb{R}^3$ are parameters to be trained. Finally, we determined the answer-type estimate $\widehat{t}$ as follows.

$$p_{max} = \max(p^k + p^l), \quad 0 \leqslant l - k \leqslant n, \tag{6}$$

$$a = \arg\max p^t, \tag{7}$$

$$\widehat{t} = \begin{cases} ANS_a, & \text{if } p_1^k + p_1^l > p_{max}, \\ \text{SPAN}, & \text{else.} \end{cases} \tag{8}$$

In Equation 8, $ANS = \{\text{YES}, \text{NO}, \text{UNANS}\}$, p_1^k and p_1^l are the starting and ending position probabilities corresponding to [CLS], respectively. If the answer type is SPAN, its start and end position estimates $(\widehat{k}, \widehat{l})$ are Equation 6 when k and the value of l:

$$(\widehat{k}, \widehat{l}) = \arg\max_{k,l}(p^k + p^l), \quad 0 \leqslant l - k \leqslant n. \tag{9}$$

The three probability distributions of Equations 3–5 are all output by Softmax; therefore, we used the sum of the negative logarithmic likelihood to construct the loss function $\mathcal{L}(\theta)$ for background understanding:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{D}|}\sum_i \Big[\mathbb{1}(y_i^t = \text{SPAN})(\log p_{y_i^k}^k + \log p_{y_i^l}^l) + \mathbb{1}(y_i^t \neq \text{SPAN}) \log p_{y_i^t}^t\Big]. \tag{10}$$

where θ is the model parameter, $|\mathcal{D}|$ is the number of training samples, $\mathbb{1}(\cdot)$ is the indicator function, y_i^t is the real type of answer A_i, and y_i^k and y_i^l are the start and end positions of the span, respectively.

3.3 Knowledge transfer method

The size model is a relative concept determined by the number of parameters. A model with more parameters is called a large model, while a model with fewer parameters is called a small model. Specifically, in this article, the large model refers to BERT-base, while the small model is DistillBERT-base. Figure 2 shows the main steps and processes of the knowledge transfer method. First, we fine-tuned large and small models. The size of the model was determined based on the number of parameters. A model with a large number of parameters is known as a large model, whereas a model with a small number of parameters is known as a small model. The large and small models can be two with the same structure (e.g., BERT and DistilBERT) or two with different structures (e.g., BERT and BiDAF++ w/ n-ctx). In particular, fine-tuning was performed on two datasets, CoQA and QuAC, to obtain CoQA and QuAC fine-tuned large models, respectively. Small model fine-tuning was performed on the machine-reading comprehension SQuAD dataset to obtain SQuAD fine-tuned small models. Fine-tuning is used to prepare for subsequent work; therefore, this process is known as the preparation phase.

The next phase is learning, which is the core of the knowledge transfer method and is roughly divided into the following three steps. The first step is to initialize the KTM with a small model fine-tuned using SQuAD. The second step is to combine the training samples of the two datasets, CoQA and QuAC, and input them into the KTM and fine-tuned large models to generate the predicted values and soft labels of the samples, respectively. The third step is to calculate the loss value based on the predicted value and soft label, as well as the true label (hard label) of the sample; subsequently, let the gradient propagate

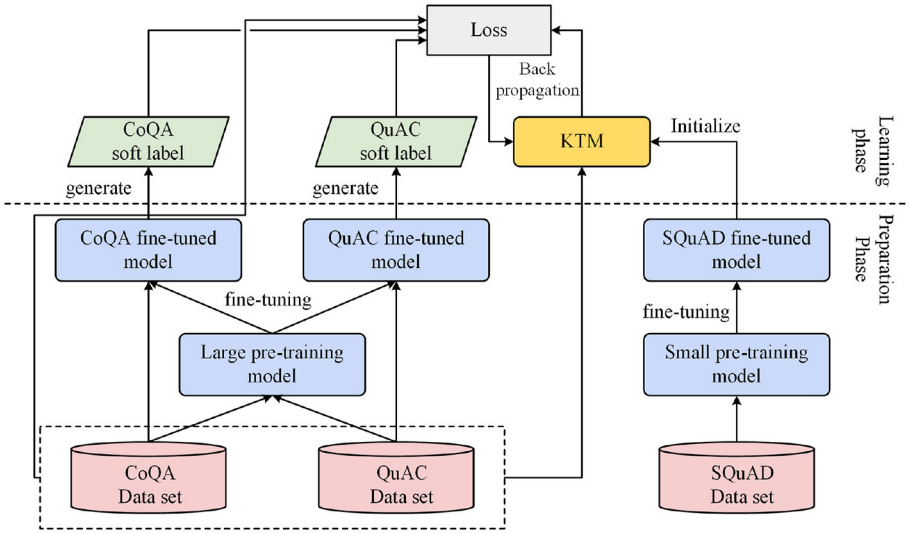

FIGURE 2
Knowledge transfer process for background understanding.

back to the KTM. In the aforementioned steps, the method of combining datasets to learn is known as co-learning, and the method of using soft labels to calculate the loss value is known as knowledge distillation.

In this study, the purpose of SQuAD fine-tuning was to let KTM learn in advance how to extract answers from the background text and jointly learn the data characteristics shared between CoQA and QuAC by expanding the training samples. Knowledge distillation allows KTM to master answers learned from the big model.

Therefore, the proposed knowledge transfer method used SQuAD fine-tuning, co-learning, knowledge distillation, and knowledge transfer between different tasks, between different data sets, and from large to small models. The SQuAD fine-tuning method has been described in Devlin et al. (2019).

3.4 Knowledge distillation

The knowledge distillation framework shown in Figure 3 contains two models: teacher and student. The teacher model is a large trained model or an ensemble of multiple models, whereas the student model is a small model that learns from the teacher model. The idea of knowledge distillation is to let the student model learn from ground-truth labels and the probability distribution output of the teacher model.

Given an m classification dataset of the form (X, Y), the classifier can be trained by minimizing the cross-entropy loss function $\mathcal{L}$:

$$p = softmax(\boldsymbol{z}), \tag{11}$$

$$\mathcal{L}_{CE} = -\sum_{k=1}^{m} Y_k \log p_k, \tag{12}$$

where p is the class probability distribution, and $\boldsymbol{z} \in \mathbb{R}^m$ is the logits of the model. If the aforementioned training process is the process of classifier learning according to the real label Y, then knowledge distillation includes the process of student model learning according to the class probability distribution q output based on the teacher model. Similar to Y, in several cases, the probability value of q for the correct class will be high, approaching 1, and the probability of the other classes will be 0. Thus, q does not provide more information than Y and does not make much sense for model training. To solve this problem, Hinton et al. (2015) introduced the concept of temperature into the softmax function. The modified softmax function is

$$softmax(\boldsymbol{z}; T)_k \equiv \frac{\exp(z_k/T)}{\sum_j \exp(z_j/T)}, \tag{13}$$

where T denotes the temperature. When $T = 1$, Equation 13 reduces to the standard softmax function.

Applying the modified softmax function to the teacher and student models, we obtain

$$q = softmax(\boldsymbol{z}_t; T = \tau), and \tag{14}$$

$$u = softmax(\boldsymbol{z}_s; T = \tau), \tag{15}$$

where $\boldsymbol{z}_t$ and $\boldsymbol{z}_s$ are the logits of the teacher and student models, respectively, and τ is a hyperparameter. The larger the value of T, the more "soft" q and the more information the teacher model provides. Therefore, q is often known as a soft label, the corresponding real label Y is the hard label, u is the soft predicted value, and p is the hard predicted value. According to Equations 14, 15, the loss function $\mathcal{L}_{KD}$ when the student model learns the output of the teacher model is defined as

$$\mathcal{L}_{KD} = -\sum_{k=1}^{m} q_k \log u_k. \tag{16}$$

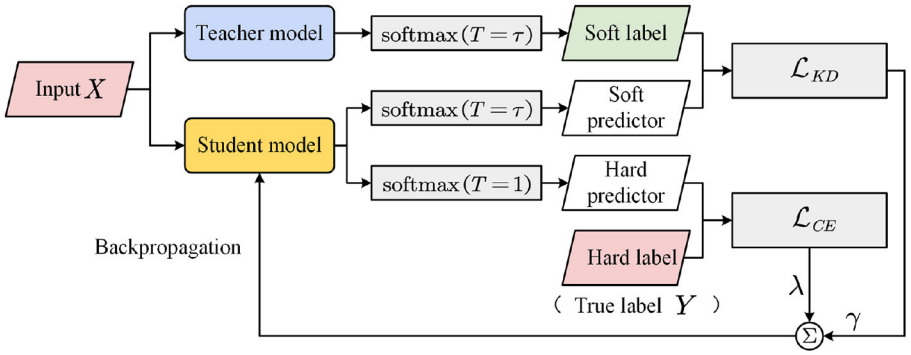

FIGURE 3
Knowledge distillation framework.

The overall optimization objective of the student model is the weighted sum of the two loss functions $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$.

$$\lambda \mathcal{L}_{CE} + \gamma \mathcal{L}_{KD}. \tag{17}$$

where λ and γ are the weights.

3.5 Training method

Based on the knowledge distillation theory (Hinton et al., 2015), the proposed KTM was considered as the student model, and the two fine-tuned large models were considered as the teacher models. The CoQA soft label, QuAC soft label, and KTM soft prediction value are denoted by q^*_{CoQA}, q^*_{QuAC}, and u^* ($* = k, l, t$), respectively. Considering u^k, the calculation method is defined as

$$u^k = softmax(\boldsymbol{w}_1^T \boldsymbol{T} + b_1; T = \tau). \tag{18}$$

According to the aforementioned soft labels and soft prediction values, the loss function of the KTM ($\mathcal{L}_{KD}(\Theta)$) can be obtained as

$$\mathcal{L}_{KD}(\Theta) = -\frac{1}{|\mathcal{D}|}\sum_i \Big[\sum_{j=1}^{L}(q_j^k \log u_j^k + q_j^l \log u_j^l) + \sum_{a=1}^{|ANS|} q_a^t \log u_a^t\Big], \tag{19}$$

where Θ is the sum of the parameters of KTM, and $|\cdot|$ represents the number of set elements. In summary, the optimization goals of the KTM are

$$\mathcal{L}(\Theta) = \lambda \mathcal{L}_{CE}(\Theta) + \gamma \mathcal{L}_{KD}(\Theta). \tag{20}$$

In particular, QuAC only has two answer types, SPAN and UNANS, and the probability distribution p^t can be discarded. The answer type is determined to be UNANS according to $p_0^k + p_0^l > p_{max}$. Therefore, for the QuAC dataset, the estimation method for the answer type $\widehat{t}_{\text{QuAC}}$ is as follows.

$$\widehat{t}_{\text{QuAC}} = \begin{cases} \text{UNANS}, & \text{if } p_0^k + p_0^l > p_{max}, \\ \text{SPAN}, & \text{else.} \end{cases} \tag{21}$$

Accordingly, the optimization objective $\mathcal{L}(\Theta)$ simplifies to

$$\mathcal{L}_{\text{QuAC}}(\Theta) = -\frac{1}{|\mathcal{D}|}\sum_i \Big[\lambda(\log p^k_{y_i^k} + \log p^l_{y_i^l}) + \gamma \sum_{j=1}^{L}(q_j^k \log u_j^k + q_j^l \log u_j^l)\Big]. \tag{22}$$

Based on the aforementioned optimization objectives, we developed the KTM training mechanism of the KTM, as explained in Algorithm 1. As the KTM training process involves two datasets, first, CoQA and QuAC, the mini-batches of the two datasets were merged, and the combined KTM training set is denoted as $\mathcal{D}$. The entire training process iterated $epoch_{max}$ rounds. Before each round, the training set $\mathcal{D}$ was disturbed to ensure the randomness of the training samples. Subsequently, a mini-batch b_α was selected from the out-of-order $\mathcal{D}$ with samples from the same dataset α. B_α was input into the large model fine-tuned by dataset α and KTM, and the soft labels and soft and hard predicted values needed to calculate the loss value were output. If α = CoQA, use Equation 20 to calculate the loss value; if α = QuAC, use Equation 22. Finally, the gradient of each parameter was calculated from the loss value, and the KTM was updated.

4 Experiment

4.1 Details

For datasets, CoQA and QuAC use release versions, whereas SQuAD uses version 2.0 (Rajpurkar et al., 2018). In terms of data preprocessing, all characters were lowercased, the maximum length of s_1 was maximum (64), and the sliding step size was stride = 128. The dropout probability of each layer was 0.1, the softmax temperature was $\tau = 2$, and the loss function weights were $\lambda = 0.5$ and $\gamma = 0.5$. For training, the mini-batch size was 12, and the maximum number of iterations was $epoch_{max} = 2$. The optimizer was Adam (Kingma and Ba, 2014), and the learning rate was ϵ = 3e-5. The first 10% of the training samples were used to warm up the learning rate and then decay linearly. In terms of prediction, the maximum answer length n of the two datasets was different: 17 for CoQA and 30 for QuAC.

```
Initialize the KTM with a small model fine-tuned using SQuAD with the initial state parameter Θ_0;
i ← 0;
foreach α in {CoQA, QuAC} do
    Divide the training samples of dataset α into mini-batches: D_a;
end
Merge a mini-batch of two datasets: D ← D_CoQA ∪ D_QuAC;
for epoch = 1 to epoch_max do    // epoch_max is the maximum number of training rounds
    shuffle D;
    foreach b_a in D do    // b_a is a mini-batch of dataset α
        Input b_a into the large model fine-tuned by dataset α and output the soft label;
        Input b_a into the KTM and output the soft and hard predicted values.
        Calculate the loss value using the soft and hard labels and the predicted values:
        L_α(Θ_i);
        Calculate the gradient: ∇(Θ_i);
        Update the KTM: Θ_{i+1} ← Θ_i − ε∇(Θ_i)
        tcp * ε is the learning rate.
        i ← i + 1;
    end
end
```

Algorithm 1. KTM training.

4.2 Metrics

CoQA and QuAC use word level F1 as the main metric, which is calculated as follows.

$$overlap = |S_{pred} \cap S_{gold}|, \tag{23}$$

$$P = \frac{overlap}{|S_{pred}|}, \tag{24}$$

$$R = \frac{overlap}{|S_{gold}|}, \tag{25}$$

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{26}$$

where S_{pred} and S_{gold} are the sequences of the predicted and standard answers, respectively, and $|\cdot|$ is the length of the sequence. We combined the two-word sequences, considered the overlapping part, calculated its length as *overlap*, and divided it with the predicted answer length and standard answer length, respectively. The accuracy P and recall rate R were obtained, and Equation 26 was used to calculate F1.

In addition to F1, CoQA uses metric exact match (EM) to measure the exact match between the predicted and standard answers. If the two answers are exactly the same, EM = 1; otherwise, EM = 0. QuAC introduced the human equivalence score (HEQ) to judge whether the model prediction reached the human average level, that is, whether modeled F1 exceeded or was equal to Human F1, which was measured in percentage. QuAC designed two evaluation metrics, HEQ-Q and HEQ-D, based on questions and dialogs. HEQ-Q and HEQ-D count the proportion of questions and dialogs, respectively, that meet the aforementioned conditions in each round.

4.3 Main results

Table 3 lists the experimental results of the background understanding of the Q&A system for the Bert-Base, Distilbert-Base, and KTM models. The first two models are fine-tuned, whereas KTM uses Bert-Base as a large model and Distilbert-Base as a small model. It was trained using the proposed knowledge transfer method. Because neither CoQA nor QuAC disclosed test sets, Table 1 lists only the experimental results of the validation sets.

As BERT-Base possesses twice as many transformer coding layers as Distilbert-Base, the performances of the two models differ substantially. The proposed knowledge transfer method attempts to bridge this gap by enabling small models to perform better than large models, even with fewer parameters. As listed in Table 3, KTM narrowed the gap between the CoQA validation sets from 5.5 F1 to 1.7 F1. In the QuAC verification set, KTM exceeded BERT-Base, thereby increasing the F1 value by 0.8 and the HEQ-Q value by 1.0; HEQ-D is equal to BERT-Base. By comparing KTM and Distilbert-Base models, which have the same structure but different training methods, we found that the indices of KTM are better than DistilBERT-Base, and the gap is evident. The knowledge transfer method is much better than the direct fine-tuning method.

In Table 3, the sum of the time for one round of training on both datasets is 27 h for BERT-base and 13.25 h for DistilBERT-base. KTM is trained for both datasets together, and the time for one iteration is 13.5 h, which is close to DistilBERT-base but only half of the BERT-base. It can be observed that in terms of training time, knowledge transfer is the same as direct fine-tuning, without increasing the complexity of training. In addition, KTM is consistent with Distilbert-Base but lower than BERT-Base in

TABLE 3 Experimental results with regard to background understanding.

Model	Params	Training	CoQA		QuAC		
	(M)	(Hour)	F1	EM	F1	HEQ-Q	HEQ-D
BERT-base	110	27.00*	80.5	71.5	64.5	60.4	6.7
DistilBERT-base	66	13.25*	75.0	65.8	59.7	55.5	4.7
KTM	66	13.50	78.8	69.5	**65.3**	**61.4**	6.7

* is the sum of training duration of CoQA and QuAC datasets.
The training and prediction time is the time required to complete the corresponding operation of the model on a single Tesla K40m graphics card.
Bold indicates best results.

TABLE 4 Background understanding ablation experiment results.

Model	CoQA		QuAC	
	F1	ΔF1	F1	ΔF1
KTM	78.8	—	65.3	—
− Knowledge Distillation (KTM w/o KD)	77.4	−1.4	63.3	−2.0
− Co-learning with Homogeneous Datasets	78.8	−0.0	64.9	−0.4
− Similar Task Fine-tuning (KTM w/o SQuAD)	75.9	−2.9	63.5	−1.8

Underline indicates best results.

terms of memory footprint and predicted speed owing to the same model structure.

In summary, the performance of the knowledge transfer model is close to or better than that of the large model, whose scale is approximately twice as large under the condition of low resource consumption, thereby achieving an effective situation of resource consumption and performance.

5 Analysis and discussion

To deeply analyze the utility of knowledge transfer, first, the ablation experiment analyzed the contribution of various strategies to the performance of the knowledge transfer model. Subsequently, the impact of knowledge distillation and fine-tuning of similar tasks were analyzed. Finally, the advantages and disadvantages of the knowledge transfer methods were summarized by comparing the four aspects: question type, answer type, answer span length, and dialog rounds.

5.1 Ablation study

The main idea of the ablation experiment is to remove one of the above strategies for model training, thereby obtaining KTM w/o KD, KTM w/o QuAC (training CoQA separately), KTM w/o CoQA (training QuAC separately), KTM w/o Four models, such as SQuAD, and then compare the F1 value with KTM. The larger the gap, the greater the impact and contribution are.

The ablation results are listed in Table 4. We found that knowledge distillation considerably affects QuAC, CoQA is influenced primarily by SQuAD fine-tuning, and the effect of co-learning between the two datasets is negligible. After analysis, the reason for the small effect of co-learning may be that the two data sets differently deal with general questions: CoQA uses "yes" and "no" for answers, whereas QuAC uses background text for answers. The different processing modes of the two models lead to the failure of unified cognition in the learning process but increase the noise during training. Thus, by skipping co-learning, the model can still achieve better performance.

TABLE 5 Statistics of the number of questions whose F1 is improved owing to knowledge distillation.

	CoQA	QuAC
Number of questions with increased F1	205	244
And answer exactly (F1 = 1)	125	112

5.2 Influence analysis of knowledge distillation

To analyze the impact of knowledge distillation on the model, we compared the F1 of the three models of BERT-base, KTM, and KTM without KD for each question in the validation set. The former ones, that is, BERT-base and KTM, are larger than the latter one. The results are listed in Table 5, and the last row of the table lists the number of questions for which F1 was raised from l<1 to 1. However, compared to the problem of F1 = 1, both CoQA and QuAC have a large gap. Knowledge distillation can transfer answer-related knowledge from a large model to a small model; however, this transfer can only improve the part of the performance.

Figures 4, 5 show the attention matrices of Q_1 and some background text on the three models (BERT-base, KTM w/o KD,

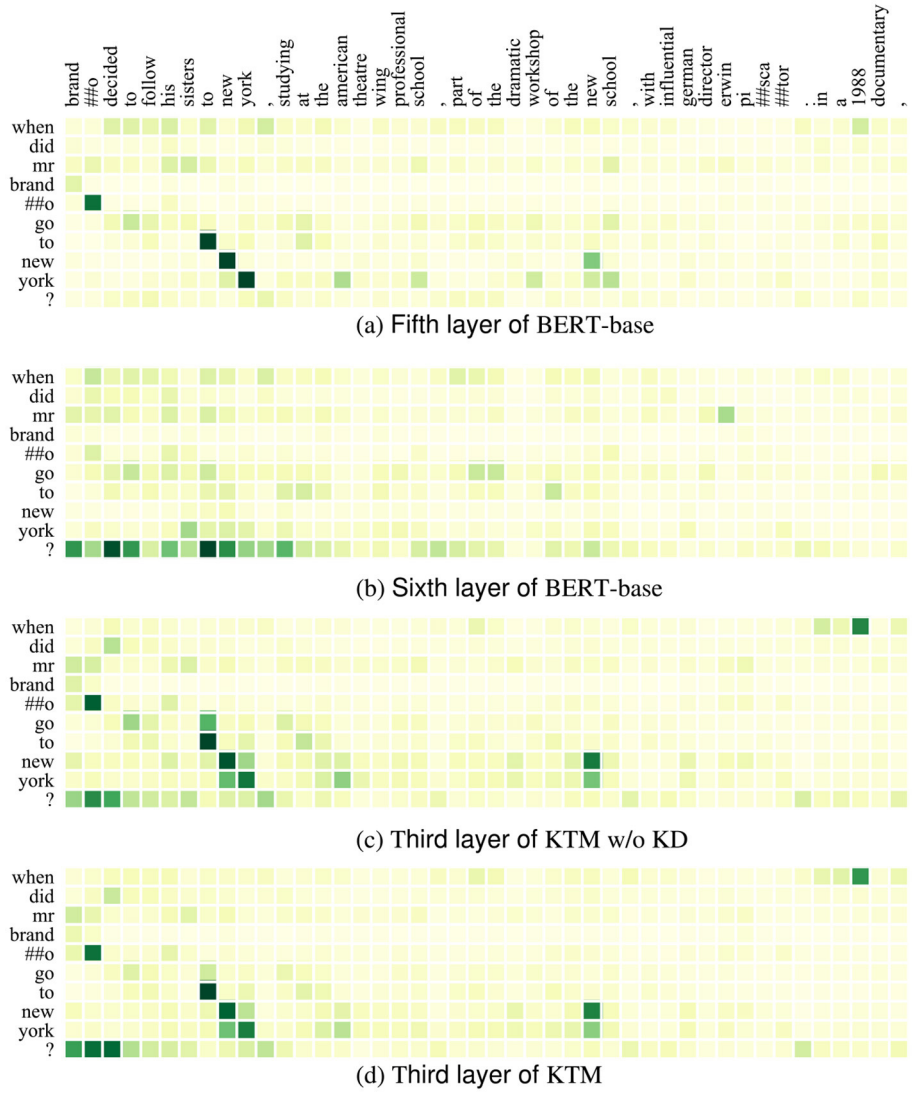

FIGURE 4
Attention matrix of QuAC example Q_1 and part of background text before and after knowledge distillation. **(a)** Fifth layer of BERT-base. **(b)** Sixth layer of BERT-base. **(c)** Third layer of KTM w/o KD. **(d)** Third layer of KTM.

and KTM) and the predicted probabilities of the start and end positions of the answer. The "when" in Q_1 is a time-related problem; in Figure 4, KTM w/o KD and KTM focus on "1988", whereas BERT-base focuses more on "brando" and "to new york." KTM learned this feature through knowledge distillation; therefore, it increased the attention weight of "brando." Finally reflected in the span probability $\widehat{k} + \widehat{l}$ predicted in Figure 1, the KTM w/o KD model lacks attention to "brando" to ensure that the correct answer is in the span "brando ... school," has a probability value of only 0.391, which is slightly smaller than the probability value of the fragment "1988," which is 0.392, and finally outputs the wrong answer "1988." The probabilities of BERT-base with regard to these two spans are 0.910 and 0.153, respectively, which express sufficient affirmation for the correct answer and avoiding the wrong answer. KTM learns this from BERT-base; therefore, it increases the probability value of the span where the correct answer is located at 0.502, whereas the probability value of the span "1988" does not change much to 0.415. As 0.502 > 0.415, KTM, like BERT-base, outputs the correct answer fragment "brando ... school." Knowledge distillation uses the large model to correct the misunderstanding of the small model.

5.3 Analysis of the impact of fine-tuning on similar tasks

Similar to the impact analysis of knowledge distillation, we compared the F1 of each question in the verification set of KTM and KTM W/O SQuAD models, counted the number of the former ones that are greater than the latter ones, and divided them according to the types of answers (Table 6). The purpose of introducing SQuAD fine-tuning in this study was to allow the model to learn in advance how

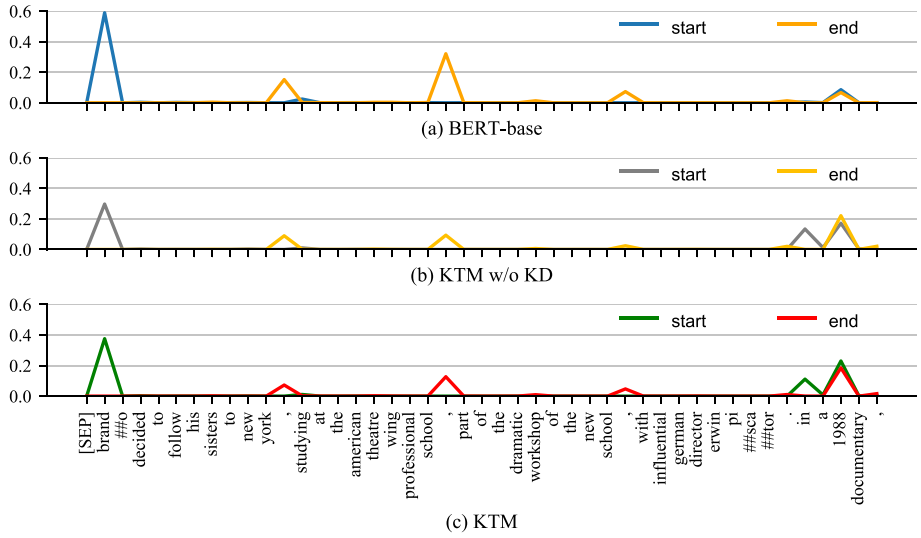

FIGURE 5
Probability distribution of answer spans for Q_1 in QuAC examples before and after knowledge distillation. **(a)** BERT-base. **(b)** KTM w/o KD. **(c)** KTM.

TABLE 6 Statistics of the number of questions whose F1 has been improved owing to fine-tuning of similar tasks.

	SPAN answer (percentage)	Others	Total
CoQA	435 (86.1%)	70	505
QuAC	410 (88.7%)	52	462

to extract answer spans from the background text; Table 6 lists the SPAN types separately. Thus, we found that SQuAD fine-tuning contributes the most to the SPAN class answer in terms of performance improvement. This demonstrates that SQuAD fine-tuning can allow small models to learn in advance how to extract background snippets from similar machine-reading comprehension tasks, which renders it an indispensable step in knowledge transfer methods.

Herein, we demonstrate the impact of SQuAD fine-tuning on model decisions using Q_1 in CoQA (Table 1). Figures 6, 7 show the attention matrices of Q_1 and part of the background text on the KTM without SQuAD and KTM models, and the predicted probabilities of the start and end positions of the answer. The "who" in Q_1 is a person-related question; thus, as shown in Figure 6, KTM focuses the "who" attention on the person named entity "michael scott moore," whereas the KTM w/o SQuAD model on the segment "michael scott moore" and "somali pirates" has concerns. Finally, while making a decision based on the span probability $\widehat{k} + \widehat{l}$, as shown in Figure 1, the KTM w/o SQuAD model makes a judgment that both spans may be the answers, and as the former probability value (0.864) is less than the latter probability value (1.063); thus, the wrong answer "somali pirates" is the output. As KTM learned the pattern of "who" and person's name from SQuAD, knowing "michael scott moore" was the only correct answer, thereby giving this clip a very high degree of confidence. This demonstrates the positive significance of SQuAD fine-tuning, which makes the extraction of answer spans more accurate.

5.4 Compare by type of problem

To analyze the impact of knowledge transfer on different types of questions, we divide the questions into "what," "who," "when," "which," "where," "how," "why," "general" according to the question words and "others" (nine categories). Herein, "general" refers to a general interrogative sentence, and when the question does not meet the first eight categories, it is assigned to the last "others" category. The comparison results of the three models of DistilBERT-base, KTM, and BERT-base on nine types of problems are shown in Figures 8, 9. From the perspective of CoQA, KTM surpasses DistilBERT-base of the same size in all the nine categories of questions, and on "how" category questions, KTM is closest to the large model BERT-base. However, in the "general" category, the performance of KTM has not significantly improved, and Section 5.5 discusses more on this. In terms of QuAC, KTM outperformed Distilbert-Base for all the problem types and BERT-Base for all the types except for "which." In conclusion, knowledge transfer positively affects the problem types.

5.5 Comparison by type of answer

Based on whether the answer exists and general question is answered in the form of yes/no, this study divides CoQA answers into SPAN, YES, NO, and UNANS types and divides QuAC answers into SPAN and UNANS types. When making predictions, the model first determines the answer type and then generates a specific answer text. Therefore, it is necessary to analyze the impact of knowledge transfer on performance by classifying answer types.

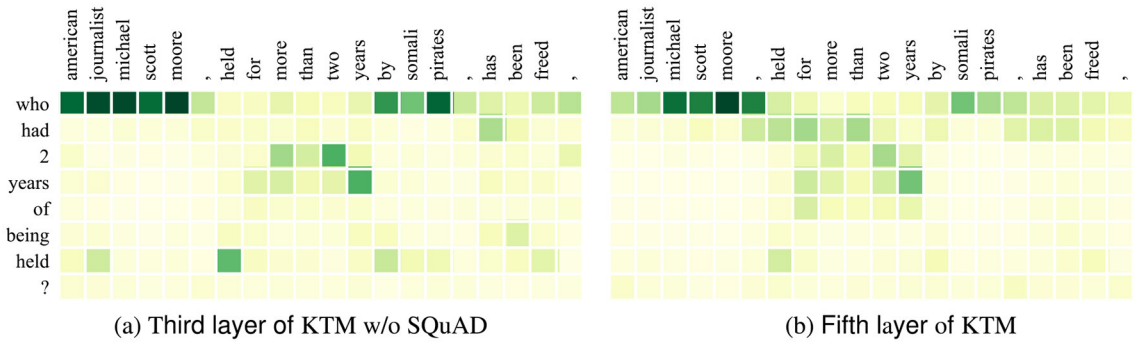

FIGURE 6
Attention matrix of CoQA example Q_1 and some background text before and after fine-tuning on similar tasks. **(a)** Third layer of KTM w/o SQuAD. **(b)** Fifth layer of KTM.

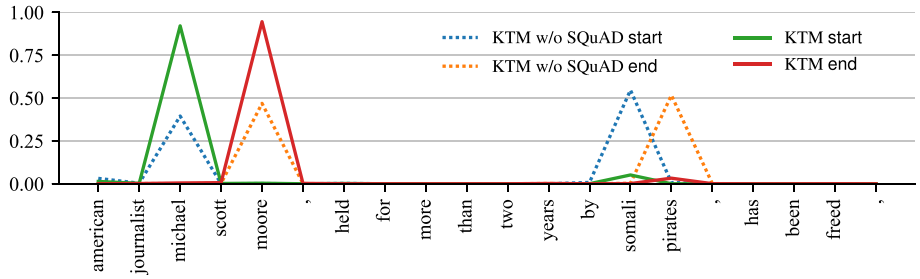

FIGURE 7
Probability distribution of answer spans for CoQA examples before and after fine-tuning on similar tasks.

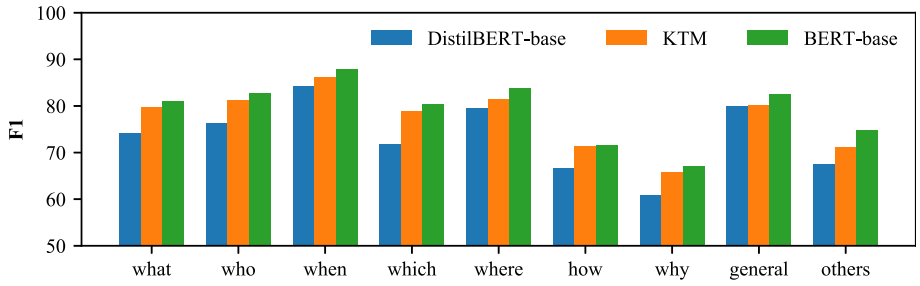

FIGURE 8
F1 of each problem type in CoQA.

First, we compared the results of the three models: DistilBERT-base, KTM, and BERT-base on CoQA (Figure 10). The performance of the KTM significantly improved based on the SPAN and UNANS answers, followed by NO answers. For the YES answers, the performance declined.

Therefore, we analyzed the confusion matrix of answer type discrimination (Figure 11) and found that BERTert-Base misjudged YES as UNANS for a higher number of samples than Distilbert-Base. The KTM retained this feature and misjudged a similar number of YES samples to UNANS. As the YES samples in the training set are fewer than the NO samples, the model is biased, thereby causing another part of the YES samples in the validation set to be misjudged as NO. Therefore, the performance of the KTM with regard to the YES class answer of the CoQA validation set is not at par with DistilBERT-base.

Figure 12 shows a comparison of the results of the three models with regard to QuAC. The overall performance of the KTM surpassing the BERT-base is owing to the improvement in the performance of the SPAN-type answers, whereas for the UNANS-type answers, the situation is the same as that of the YES-type answers with regard to CoQA, thus degrading the performance. Further analysis of the answer-type discriminant confusion matrix, as shown in Figure 13, demonstrates that the performance drop is caused by the misclassification of UNANS samples as SPAN. The underlying reason may be that the sample imbalance causes the model prediction to be biased toward SPAN, which is evident

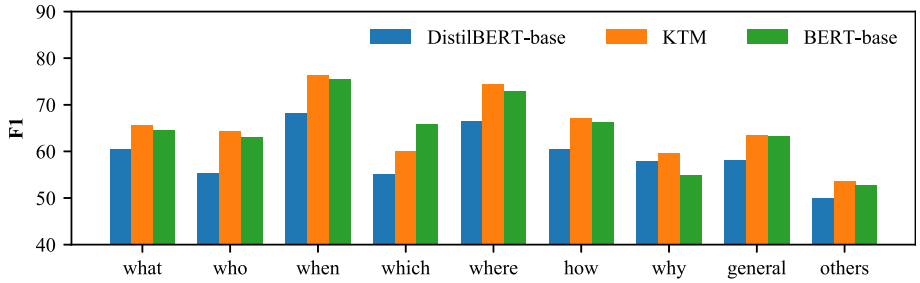

FIGURE 9
F1 of each problem type in QuAC.

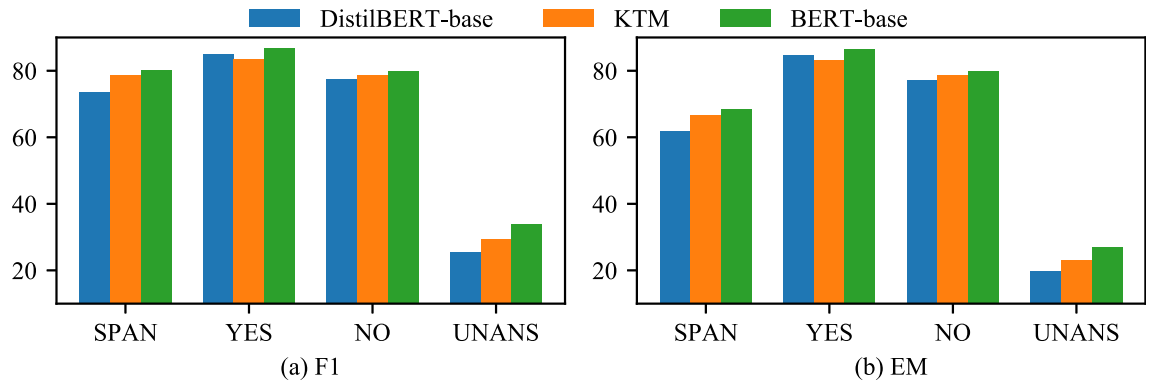

FIGURE 10
Experimental results of each answer type in CoQA. **(a)** F1. **(b)** EM.

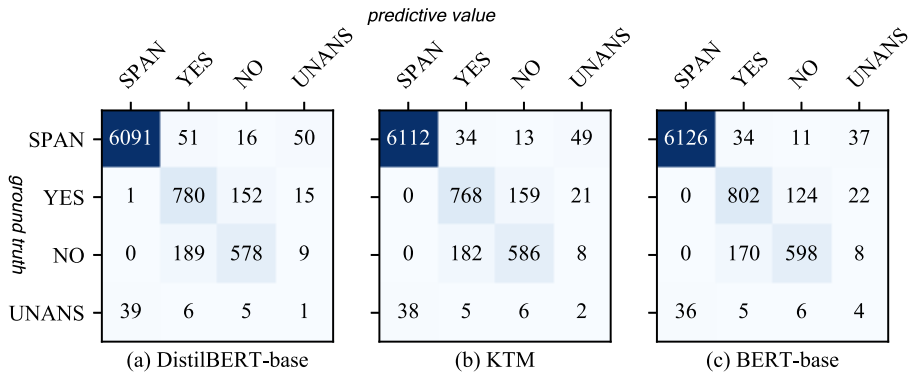

FIGURE 11
Confusion matrix for answer type discrimination in CoQA. **(a)** DistilBERT-base. **(b)** KTM. **(c)** BERT-base.

from the misjudgment rate (the UNANS misjudgment rate is approximately two times that of SPAN).

Compared with direct fine-tuning, knowledge transfer is beneficial to the performance improvement of different answer types. However, owing to the misjudgment of the large model and the problem of unbalanced samples, in few cases, the performance of a certain type of answer will slightly decrease; however, the overall impact will be negligible.

5.6 Compare by length of span

The SPAN-type answers in the CoQA and QuAC validation sets were filtered out, and the F1 of the three models of DistilBERT-base, KTM, and BERT-base was aggregated by span length; the results are shown in Figures 14, 15. As each question in the validation set corresponds to multiple optional answers, when calculating the length of the answer span for each question, the average is rounded off. After processing, the CoQA and QuAC answer span lengths

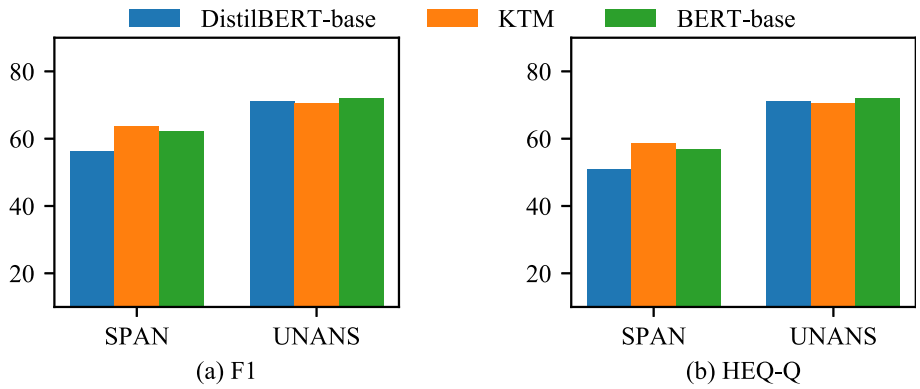

FIGURE 12
Experimental results of each answer type with regard to QuAC. **(a)** F1. **(b)** HEQ-Q.

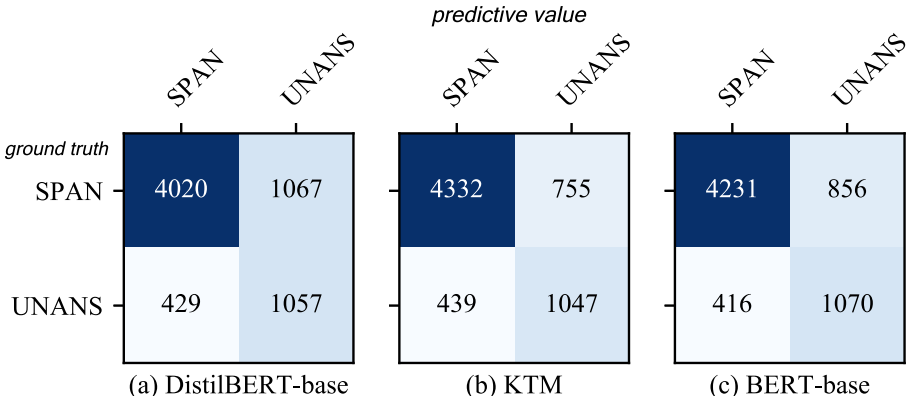

FIGURE 13
Confusion matrix for answer type discrimination in QuAC. **(a)** DistilBERT-base. **(b)** KTM. **(c)** BERT-base.

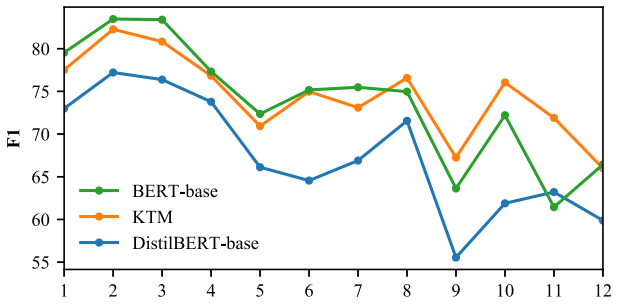

FIGURE 14
F1 for different answer span lengths in CoQA.

ranged from 1 to 19 and from 1 to 28, respectively. Although the maximum length of CoQA answer fragments can reach 19, the median is only 2, and text spans with a length of i <12 account for 99.7%. Figure 14 shows the comparison of the results. Similarly, 99.7% of the answer spans in the QuAC validation set were <25 in length, and only these results were compared (Figure 15).

The KTM curve in Figure 14 is close to the BERT-base, and after the length span of eight, KTM surpasses the BERT-base. For this phenomenon, we explain that QuAC answer spans are longer than CoQA, and KTM masters the answering skills of long spans based on the joint learning of the two datasets. For 8–11 in Figure 15, KTM surpasses BERT-base, which further proves that KTM possesses advantages in answering long spans. In addition, the overall KTM curve in Figure 15 is above BERT-base. On QuAC, the overall performance of KTM surpasses that of the large-model BERT-base and is valid for answer spans of all lengths. As shown in Figures 14, 15, we can conclude that compared with direct fine-tuning, knowledge transfer improves the performance of all answer spans of length, particularly for answer spans of 8–11, which exceeds the large model and significantly improves.

5.7 Compare by dialogue rounds

A complete interactive Q&A comprises multiple rounds of questioning and answering; therefore, this section attempts to analyze the knowledge transfer utility based on the perspective of dialogue rounds and intends to answer the following questions. (1)

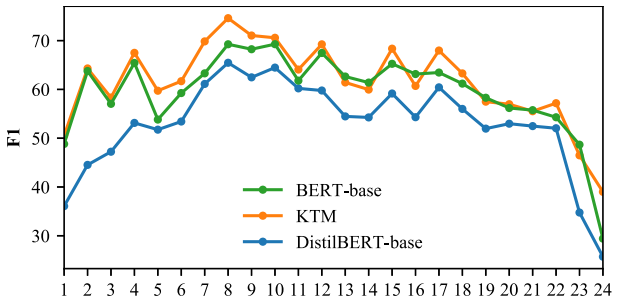

FIGURE 15
F1 for different answer span lengths in the QuAC.

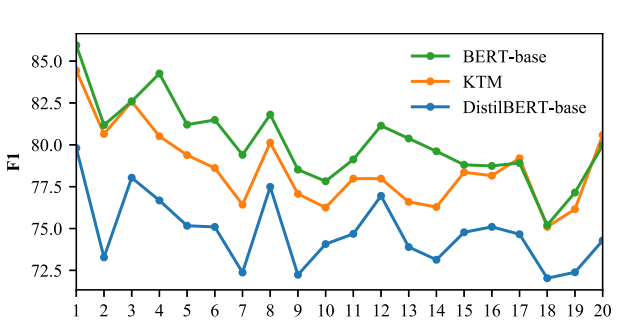

FIGURE 16
F1 for different dialogue rounds in the CoQA.

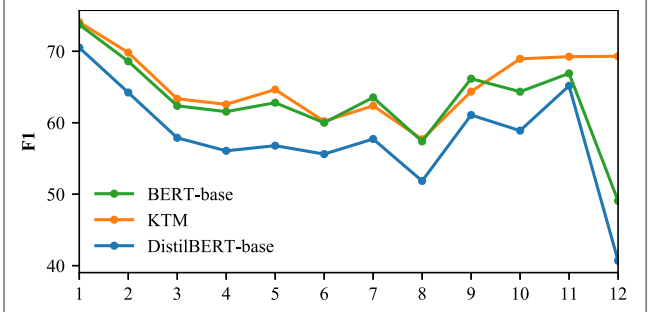

FIGURE 17
F1 for different dialogue rounds in the QuAC.

How does knowledge transfer occur in different rounds of dialogue? Compared with direct fine-tuning, how much has the performance improved? How much is the difference compared to the big model? (2) As the number of dialogue turns increases, thus gradually decreasing the performance, can knowledge transfer improve this problem?

We used the CoQA and QuAC validation sets to compare the F1 of the DistilBERT-base, KTM, and BERT-base models for different dialogue rounds, and the results are shown in Figures 16, 17. The CoQA dialogue rounds were distributed between 1 and 25, with an average value of 15.97. There were 11 rounds with more than 20 rounds, accounting for only 2.2% of the total. Therefore, Figure 1 only compares F1 of rounds 1 to 20. The QuAC validation set has 1,000 complete interactive Q&As, with all dialogue turns distributed between 1 and 12. Figure 17 compares F1 for all the rounds.

The KTM curve in Figure 16 is located between BERT-base and DistilBERT-base. Compared with the direct fine-tuning DistilBERT-base of the same scale, the performance of KTM is considerably improved; however, compared with the large model BERT-base, KTM still has a large gap. With an increase in dialogue rounds, all three curves exhibit a trend of fluctuation and decline. The performance of knowledge transfer is consistent with that of direct fine-tuning and fails to address the problem of performance degradation. Figure 17 shows different performances. The KTM curve is slightly higher than that of BERT-base, and in the later stage of interaction (10–12 rounds), when the performance of DistilBERT-base and BERT-base decreases, the KTM trained by knowledge transfer maintains excellent performance. In the QuAC validation set, knowledge transfer significantly improves the performance of all the rounds, which renders it equivalent with or allows it to surpass the large model. Moreover, knowledge transfer has been successful in improving the performance of deep dialogue rounds.

Based on the aforementioned analysis, we have the following answers to the two questions raised at the beginning of this section. In all the dialogue rounds, the models trained by knowledge transfer are better than those trained by direct fine-tuning, and the improvement is significant. With the gap with large models and whether they can improve the performance degradation of deep dialogue rounds, different datasets have different performances.

6 Conclusion

This study investigates small pre-trained models for background understanding in automated Q&A. The main work includes a comprehensive comparison of the computing time and performance differences of mainstream pre-training models in the task of Q&A background understanding. A small-scale pre-training model suitable for interactive Q&A background understanding is proposed, and strategies such as knowledge distillation, co-learning of similar data sets, and fine-tuning of similar tasks are used to realize a variety of knowledge transfer and make the performance of small models with low resource consumption, comparable or surpassed by large models. In general, knowledge distillation, co-learning on similar datasets, and fine-tuning on similar tasks play their respective roles in the process of knowledge transfer and contribute to model performance to varying degrees. In future work, we will continue to explore a unified question processing method to enhance the influence of co-learning.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

XH: Conceptualization, Writing – original draft. HS: Validation, Writing – review & editing. ML: Data curation, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was supported by the National Natural Science Foundation of China (No. 62262029), the Natural Science Foundation of Jiangxi Province (No. 20212BAB202016), and the Science and Technology Research Project of Jiangxi Provincial Department of Education (Nos. GJJ200318 and GJJ210520).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Chang, Y., Narang, M., Suzuki, H., Cao, G., Gao, J., Bisk, Y., et al. (2022). "Webqa: multihop and multimodal QA," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA: IEEE), 16495–16504. doi: 10.1109/CVPR52688.2022.01600

Chen, C., Wang, C., Qiu, M., Gao, D., Jin, L., Li, W., et al. (2021). "Cross-domain knowledge distillation for retrieval-based question answering systems," in *Proceedings of the Web Conference 2021* (New York, NY: ACM), 2613–2623. doi: 10.1145/3442381.3449814

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., et al. (2018). "Quac: question answering in context," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels: ACL), 2174–2184. doi: 10.18653/v1/D18-1241

Cui, H., Peng, T., Xiao, F., Han, J., Han, R., Liu, L., et al. (2023). Incorporating anticipation embedding into reinforcement learning framework for multi-hop knowledge graph question answering. *Inf. Sci.* 619, 745–761. doi: 10.1016/j.ins.2022.11.042

Cui, Y., Liu, T., Che, W., Chen, Z., and Wang, S. (2022). Teaching machines to read, answer and explain. *IEEE/ACM Trans. Audio Speech Lang. Process.* 30, 1483–1492. doi: 10.1109/TASLP.2022.3156789

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "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, Volume 1 (Long Papers)* (Minneapolis, MN: Association for Computational Linguistics), 4171–4186.

Garg, S., and Moschitti, A. (2021). Will this question be answered? Question filtering via answer model distillation for efficient question answering. *arXiv* [preprint]. arXiv:2109.07009. doi: 10.48550/arXiv.2109.07009

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: a survey. *Int. J. Comput. Vis.* 129, 1789–1819. doi: 10.1007/s11263-021-01453-z

Gu, Y., Kase, S., Vanni, M., Sadler, B., Liang, P., Yan, X., et al. (2021). "Beyond IID: three levels of generalization for question answering on knowledge bases," in *Proceedings of the Web Conference 2021* (New York, NY: ACM), 3477–3488. doi: 10.1145/3442381.3449992

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv* [Preprint]. arXiv:1503.02531. doi: 10.48550/arXiv.1503.02531

Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., et al. (2023a). Language is not all you need: aligning perception with language models. *arXiv* [preprint] arXiv:2302.14045. doi: 10.48550/arXiv.2302.14045

Huang, X., Song, H., and Lu, M. (2023b). "Intent understanding for automatic question answering in network technology communities based on multi-task learning," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (Cham: Springer), 117–129. doi: 10.1007/978-3-031-36822-6_10

Huang, X., Song, H., and Lu, M. (2023c). "Role understanding for spoken dialogue system based on relevance ranking," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (Cham: Springer), 105–116. doi: 10.1007/978-3-031-36822-6_9

Izacard, G., and Grave, E. (2020). Distilling knowledge from reader to retriever for question answering. *arXiv* [preprint] arXiv:2012.04584. doi: 10.48550/arXiv.2012.04584

Ju, Y., Zhao, F., Chen, S., Zheng, B., Yang, X., Liu, Y., et al. (2019). Technical report on conversational question answering. *arXiv* [Preprint]. arXiv:1909.10772. doi: 10.4855/arXiv.1909.10772

Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. (2023). "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning* (PMLR), 15696–15707.

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv* [Preprint]. arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., et al. (2020). "Albert: a lite Bert for self-supervised learning of language representations," in *8th International Conference on Learning Representations (ICLR)*.

Lauriola, I., Lavelli, A., and Aiolli, F. (2022). An introduction to deep learning in natural language processing: models, techniques, and tools. *Neurocomputing* 470, 443–456. doi: 10.1016/j.neucom.2021.05.103

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., et al. (2022). A survey on text classification: from traditional to deep learning. *ACM Trans. Intell. Syst. Technol.* 13, 1–41. doi: 10.1145/3495162

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: a robustly optimized bert pretraining approach. *arXiv* [preprint]. arXiv:1907.11692. doi: 10.48550/arXiv.1907.11692

Martinez-Gil, J. (2023). A survey on legal question-answering systems. *Comput. Sci. Rev.* 48:100552. doi: 10.1016/j.cosrev.2023.100552

McCarley, J., Chakravarti, R., and Sil, A. (2020). Structured pruning of a Bert-based question answering model. *arXiv* [Preprint]. arXiv:1910.06360. doi: 10.48550/arXiv.1910.06360

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J., et al. (2021). deep learning-based text classification: a comprehensive review. *ACM Comput. Surv.* 54, 1–40. doi: 10.1145/3439726

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X., et al. (2020). Pre-trained models for natural language processing: a survey. *arXiv* [Preprint]. arXiv:2003.08271. doi: 10.48550/arXiv.2003.08271

Qu, C., Yang, L., Qiu, M., Croft, W. B., Zhang, Y., Iyyer, M., et al. (2019). "Bert with history answer embedding for conversational question answering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY: ACM), 1133–1136. doi: 10.1145/3331184.3331341

Rajpurkar, P., Jia, R., and Liang, P. (2018). "Know what you don't know: unanswerable questions for squad," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Melbourne: ACL), 784–789. doi: 10.18653/v1/P18-2124

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (Austin, TX: ACL), 2383–2392. doi: 10.18653/v1/D16-1264

Reddy, S., Chen, D., and Manning, C. D. (2019). Coqa: a conversational question answering challenge. *Trans. Assoc. Comput. Linguist*. 7, 249–266. doi: 10.1162/tacl_a_00266

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). "Distilbert, a distilled version of Bert: smaller, faster, cheaper and lighter," in *NeurIPS EMC*^{2} *Workshop*.

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2019). Green Ai. *arXiv* [preprint]. arXiv:1907.10597. doi: 10.48550/arXiv.1907.10597

Shao, Z., Yu, Z., Wang, M., and Yu, J. (2023). "Prompting large language models with answer heuristics for knowledge-based visual question answering," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Vancouver, BC: IEEE), 14974–14983. doi: 10.1109/CVPR52729.2023.01438

Singh, D., Reddy, S., Hamilton, W., Dyer, C., and Yogatama, D. (2021). End-to-end training of multi-document reader and retriever for open-domain question answering. *Adv. Neural Inf. Process. Syst*. 34, 25968–25981. doi: 10.5555/3540261.3542249

Stede, M., and Schlangen, D. (2004). "Information-seeking chat: dialogues driven by topic-structure," in *Proceedings of Catalog (The 8th Workshop on the Semantics and Pragmatics of Dialogue; SemDial04)* (Barcelona).

Wu, J., Lu, J., Sabharwal, A., and Mottaghi, R. (2022). Multi-modal answer validation for knowledge-based VQA. *Proc. AAAI Conf. Artif. Intell*. 36, 2712–2721. doi: 10.1609/aaai.v36i3.20174

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., Le, Q. V., et al. (2019a). "Xlnet: Generalized autoregressive pretraining for language understanding," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Red Hook, NY: Curran Associates Inc.), 5754–5764.

Yang, Z., Shou, L., Gong, M., Lin, W., and Jiang, D. (2019b). Model compression with multi-task knowledge distillation for web-scale question answering system. *arXiv* [preprint]. arXiv:1904.09636. doi: 10.48550/arXiv.1904.09636

Yang, Z., Shou, L., Gong, M., Lin, W., and Jiang, D. (2020). "Model compression with two-stage multi-teacher knowledge distillation for web question answering system," in *Proceedings of the 13th International Conference on Web Search and Data Mining* (New York, NY: ACM), 690–698. doi: 10.1145/3336191.3371792

Yatskar, M. (2019). "A qualitative comparison of coqa, squad 2.0 and quac," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, MN: Association for Computational Linguistics), 2318–2323.

Yeh, Y.-T., and Chen, Y.-N. (2019). "Flowdelta: modeling flow information gain in reasoning for conversational machine comprehension," in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering* (Hong Kong: ACL), 86–90. doi: 10.18653/v1/D19-5812

Yu, N., Zhang, M., Fu, G., and Zhang, M. (2022). "RST discourse parsing with second-stage edu-level pre-training," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Dublin: ACL), 4269–4280. doi: 10.18653/v1/2022.acl-long.294