Keep Skills in Mind: Understanding and Implementing Skills in Commonsense Question Answering

Meikai Bao[1,2], **Qi Liu**[1,2,*], **Kai Zhang**[1,2], **Ye Liu**[1,2], **Linan Yue**[1,2], **Longfei Li**[3], **Jun Zhou**[3]

[1]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China

[2]State Key Laboratory of Cognitive Intelligence

[3]Ant Financial Services Group

{baomeikai, sa517494, liuyer, lnyue}@mail.ustc.edu.cn, qiliuql@ustc.edu.cn, longyao.llf@antgroup.com, jun.zhoujun@antfin.com

Abstract

Commonsense Question Answering (CQA) aims to answer questions that require human commonsense. Closed-book CQA, as one of the subtasks, requires the model to answer questions without retrieving external knowledge, which emphasizes the importance of the model's problem-solving ability. Most previous methods relied on large-scale pre-trained models to generate question-related knowledge while ignoring the crucial role of skills in the process of answering commonsense questions. Generally, skills refer to the learned ability in performing a specific task or activity, which are derived from knowledge and experience. In this paper, we introduce a new approach named **D**ynamic **S**kill-aware **C**ommonsense **Q**uestion **A**nswering (**DSCQA**), which transcends the limitations of traditional methods by informing the model about the need for each skill in questions and utilizes skills as a critical driver in CQA process. To be specific, DSCQA first employs commonsense skill extraction module to generate various skill representations. Then, DSCQA utilizes dynamic skill module to generate dynamic skill representations. Finally, in perception and emphasis module, various skills and dynamic skill representations are used to help question-answering process. Experimental results on two publicly available CQA datasets show the effectiveness of our proposed model and the considerable impact of introducing skills.

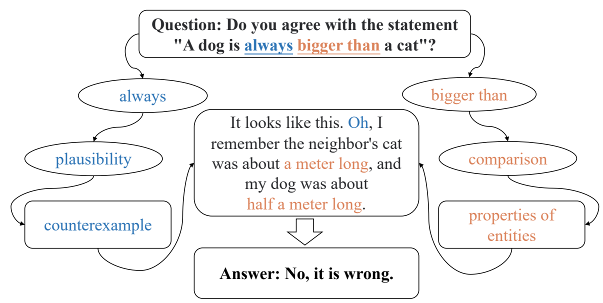

Figure 1: An example of skills help answer questions.

1 Introduction

Commonsense Question Answering (CQA) aims to answer questions that require varieties of commonsense knowledge and skills [Talmor *et al.*, 2019; Talmor *et al.*, 2021]. Closed-book CQA, one of CQA subtasks that measures a model's understanding and question-solving ability without external retrieved information, has been gaining increasing attention from both academia and industry in recent years [Roberts *et al.*, 2020; Petroni *et al.*, 2019].

Currently, most Closed-book CQA approaches improved performance by using knowledge generated by pre-trained models. Some methods [Liu *et al.*, 2022a; Wang *et al.*, 2022] used a pre-trained model to generate background knowledge relevant to the question, and utilized them as additional input for subsequent question answering. The other common methods [Jung *et al.*, 2022; Wei *et al.*, 2022] designed a series of explanatory prompts to mimic the question solving-process. However, relying too heavily on the knowledge generated by pre-trained models may ignore the crucial skill information that is vital in solving CQA questions. Skills are learned response patterns, which are also critical when solving CQA questions by humans [Moore, 2006]. For example, the process of thinking about *"counterexample"* in response to the keyword *"always"* in Figure 1 is a skill. Many previous studies [Talmor *et al.*, 2019; Talmor *et al.*, 2021] have pointed out that certain types of commonsense skills are necessary to arrive at the right answer. Meanwhile, other studies [Yoran *et al.*, 2022; Puerto *et al.*, 2023; Trivedi *et al.*, 2022] have also shown that endowing reasoning skills to pre-trained language models (PLMs) can improve abilities to answer questions in other tasks. Therefore, skills, as crucial information, should be incorporated into the question-solving process.

*Corresponding Author.

As shown in Figure 1, when answering the question *"Do you agree with the statement, a dog is always bigger than a cat?"*, we can identify the two key statements *"always"* and *"bigger than"* using our previously acquired skills. Because *"always"* implies without exception and *"bigger than"* is a comparison of size, they can recall the corresponding skills *"plausibility"* and *"comparison"*. After that, with these two recognized skills, we can realize the need to focus on the counterexamples (*"always"*) and the properties of the entities (*"comparison"*). As a result, we can correctly answer the question by utilizing skills and paying attention to considerations. Specifically, we pay more attention to counterexamples and consider the size properties of cats and dogs. This highlights the importance of identifying the skills required for answering commonsense questions in CQA tasks. However, to the best of our knowledge, there are few works in this area.

There are three main challenges inherent in utilizing commonsense skills in CQA tasks. First, most commonsense questions often do not have labels indicating the skills related to it, which creates a huge obstacle on how to proceed with further use of skills. Second, commonsense questions often require multiple skills [Talmor *et al.*, 2019; Talmor *et al.*, 2021]. For example, the question in Figure 1 involves both *"plausibility"* and *"comparison"* skills. Therefore, how to combine multiple required skills is a problem that needs to be solved. Third, after acquiring the skills required for the questions, how to use a variety of skills to guide answering the questions is another challenge.

To tackle the above challenges, we propose the **D**ynamic **S**kill-aware **C**ommonsense **Q**uestion **A**nswering (**DSCQA**). First, DSCQA adopts a commonsense skill extraction module to extract features from the training set as skill representations. Then, these representations are used to determine the demand for various types of skills for the current question by calculating the similarity between skill representations and the question. Second, to address the challenge of combining multiple skills, DSCQA designs a dynamic skill module to generate dynamic skill representations. In this module, various types of skill representations are fused to form dynamic skill representations according to the current question. Third, to deal with the challenge of how to use skills to help the question-answering process, DSCQA utilizes perception and emphasis module which uses all skill representations (i.e., dynamic skill representations and various skill representations) by incorporating them into the model's encoding and decoding process. To sum up, the main contributions can be summarized as follows.

- We propose a novel approach of incorporating skills into commonsense question answering in order to increase the logic of the model's solution. To the best of our knowledge, this is the first work to explore the effect of skills in Closed-book CQA tasks.
- We present Dynamic Skill-aware Commonsense Question Answering framework in which the model can understand and implement skills in Closed-book CQA.
- Extensive experiments demonstrate that the performance of DSCQA surpasses other baselines on the CSQA2 and CSQA datasets under comparable conditions.

2 Related Works

Closed-book Commonsense Question Answering. Many studies explore commonsense question answering in Closed-book condition where no additional knowledge is retrieved to help answer questions. Some studies [Petroni *et al.*, 2019; Onoe *et al.*, 2021] pointed out that a large amount of commonsense knowledge is stored in the parameters of large-scale pre-trained language models. Therefore, many researches explored using pre-trained language models to generate knowledge relevant to the current question. For instance, GKP [Liu *et al.*, 2022a] used the prompt method to conduct large-scale pre-trained models with a question to generate additional background knowledge. Chain-of-Thought Prompting [Wei *et al.*, 2022] generated a series of intermediate reasoning processes. ALEAP [Wang *et al.*, 2022] used iterative selection to make better use of the knowledge generated by the model. Selftalk [Shwartz *et al.*, 2020] presented an unsupervised method to produce question clarifications and appends them as external input. Different from previous methods, our model extracts features from the training set to get skills to help answer questions.

Skill Enhancement. As a psychological concept, skill is learned response pattern. It has been widely used in many subjects and fields. In Natural Language Processing (NLP) field, there are two main flavors to use skill to help downstream tasks. One type of research focused on having the model learn examples containing various skills to improve the model's reasoning skills. TeaBReaC [Trivedi *et al.*, 2022] proposed to help improve the performance of the model by letting the model learn elaborately designed question-answer pairs containing multiple reasoning patterns. PReasM [Yoran *et al.*, 2022] employed some predefined templates corresponding skills to extract information from the table and generated question-answer pairs for the corresponding skills. The other common research focused on the characteristics of specific skills, thus adding specialized modules. MetaQA [Puerto *et al.*, 2023] utilized expert agents obtained by using multiple models trained on the respective category corpus (e.g., QA datasets, Google queries) to represent various skills. Then, answer selector chose from the answers provided by each expert agent and gave the final answer. NumNet [Ran *et al.*, 2019] introduced a heterogeneous directed graph to improve the numerical skills of the model. Since commonsense questions are difficult to identify the required skills, our framework places greater emphasis on perceiving the corresponding commonsense skills.

3 Method

3.1 Problem Formulation

We focus on commonsense question answering tasks in the form of multiple-choice and yes or no questions. For multiple-choice questions, given a commonsense question q and a list of candidate answers $C = \{c_1, c_2, \ldots, c_i\}$, our goal is to identify the correct answer among them. For yes or no questions, given a commonsense question q, our task is to judge yes or no. Since we study the problem in the Closed-book condition, we do not retrieve any additional knowledge or other datasets to help answer the questions.

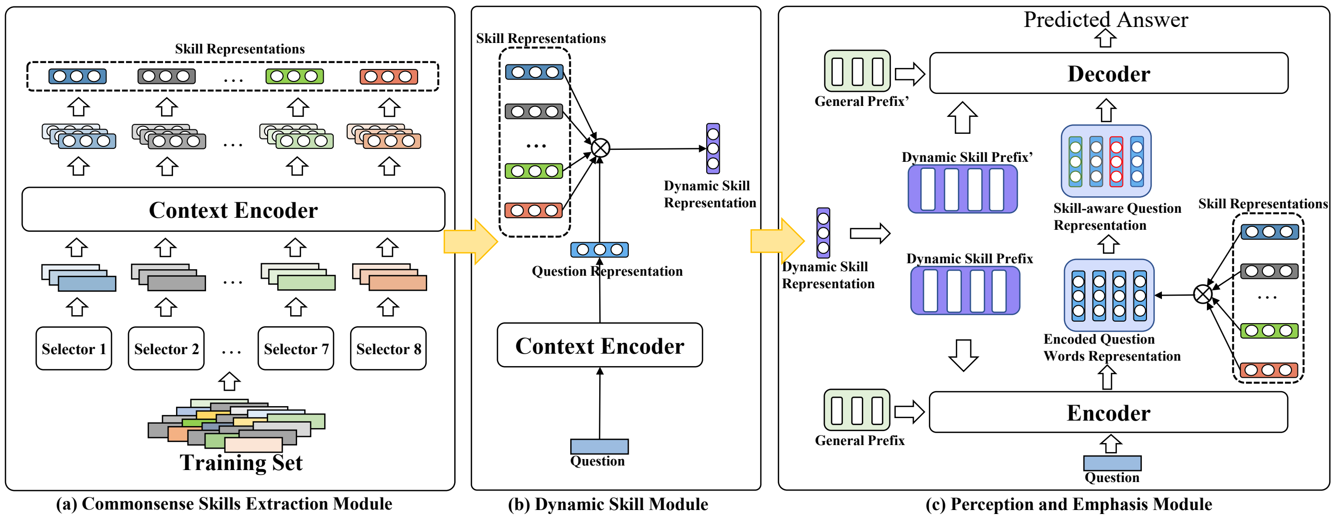

Figure 2: Overview of our proposed DSCQA framework for commonsense question answering. DSCQA consists of three modules: (a) Commonsense Skill Extraction Module; (b) Dynamic Skill Module; (c) Perception and Emphasis Module.

3.2 Overview

The pipeline of our framework is shown in Figure 2. It can be viewed as three parts: (1) Commonsense Skill Extraction Module uses features extracted from the training set to generate various skill representations; (2) Dynamic Skill Module utilizes various skill representations to generate dynamic skill representations specific to the current question; (3) Perception and Emphasis Module takes advantage of dynamic skill embeddings and various skill representations to help solve questions. Next, we will describe each part in detail.

3.3 Commonsense Skill Extraction Module

In order to identify the skills required for a question, it is necessary to understand the characteristics of each skill. However, the semantic features of questions with the same skill in various datasets vary greatly. That's because the datasets vary in question structure and focus. Therefore, the skill characteristics in different datasets need to be obtained separately to represent each skill's characteristics better. There are many studies [Chen *et al.*, 2022a; Liu *et al.*, 2020; Liu *et al.*, 2023; Yue *et al.*, 2022] that enrich the representation by extracting features from the datasets or documents. Here, in order to obtain the skill characteristics in the corresponding dataset, we first construct a skill seed dictionary based on the prior knowledge and data analysis, and then these skill seeds are used to further obtain the skill represetations.

Skill Seeds. Inspired by CSQA2 [Talmor *et al.*, 2021], we extracted a series of common words or phrases related to skill statements from dictionaries and language learning websites. After that, by combining these words or phrases with the features of commonsense questions, we derived a set of skill seeds through filtering. As shown in Table 1, skill seeds are a set of representative words or phrases of the corresponding skill type questions. After getting skill seeds, we can use regularization-based selectors to assign some pseudo skill labels to those questions in the training set successfully identified by the regularization. That is, if a skill seed appears in the question, the label to which the skill seed belongs is temporarily assigned to the question. For questions that do not contain any of the above skill seeds, we assign *unknown* labels to them.

Skills	Seeds
comparison	than, same..as, as..as
negation	never, no, cannot, not, without, neither, nowhere, nothing, nobody, none
causality	cause, because
capable	capable, able, can, cannot
plausibility	always, never
temporal	before, after
meronymy	have, part of, is a

Table 1: Description of skill seeds.

Skills Representation. We use the same type of questions to enrich the representation of the corresponding skills. First, we utilize the pseudo labels (including unknown) given by the skill seed to group the questions. Then, considering that the questions containing the same type of skills have certain similarities in the semantic space after encoding, similar to previous researches [Yu *et al.*, 2022; Ye *et al.*, 2022; Yang *et al.*, 2018], we aggregate the questions in the same group so that the aggregated representation is taken as the representation of the corresponding skill. Specifically, we encode questions by existing context encoder (e.g., sentence-T5 model [Ni *et al.*, 2022]), which takes average pooling of the output question word vectors $\{w_1, w_2, ..., w_n\}$ as the representation of the input questions. The question representation q_i and skill representation s_{v_j} is:

$$q_i = MEAN(w_1, w_2, ..., w_n), \tag{1}$$

$$s_{v_j} = MEAN(q_{j_1}, q_{j_2}, ..., q_{j_m}). \quad (2)$$

As a result, we obtain a collection of skill representations that have the characteristics of the corresponding dataset. These skill representations originate from the questions in the training set, so they have homology with the questions in the dataset. Therefore, the similarities between skills and the current question can reflect the demand degree of each skill for the current question.

3.4 Dynamic Skill Module

As we discussed in Section 1, when answering a specific question, people will consider and apply the skills corresponding to it. Previous researches [Zhang *et al.*, 2022; Wang *et al.*, 2023] have also shown that incorporating dynamic semantics can help downstream tasks. Therefore, we consider informing the model of the required skills corresponding to the question. To be specific, we propose dynamic skill module to obtain the dynamic skill representation corresponding to the question. It extracts features from each skill representation according to the similarities between the current question representation and each skill representation, thus obtains the dynamic skill representation which is dynamically generated according to the current question.

Specifically, consistent with the method of obtaining skill representations, we apply the same context encoder to encode questions and take average pooling of the question word vectors as question representation q. Then, the question representation q serves as the query, and each skill representation $\{s_{v_1}, s_{v_2}, \ldots, s_{v_m}\}$ functions as the key and value for attention calculations. Consequently, these features are extracted from each skill representation to obtain the question specific dynamic skill representation ds_{v_i}:

$$ds_{v_i} = MultiHeadAttn(\{s_{v_1}, s_{v_2}, \ldots, s_{v_m}\}, q). \quad (3)$$

3.5 Perception and Emphasis Module

In this subsection, we will introduce perception and emphasis module which uses all skill representations obtained from commonsense skills extraction module and dynamic skill module to help the model solve the commonsense questions. This module exploits commonsense skills from two perspectives, which we will introduce step by step below.

In order to make the model perceive the corresponding skill to the question during the encoding and decoding process, following [Clive *et al.*, 2022], we map skill representations into prefixes so that the question can be inferred under the control of skills. In this way, we guide the question-solving process in the desired direction and provide the model with skill attribute-level information.

General Prefix. We use Prefix-tuning [Li and Liang, 2021] to learn task-level information instead of the usual fine-tuning. By doing so, we only need to fine-tune a small set of prefix parameters (general prefix), while keeping the parameters of the pre-trained language model frozen. This not only reduces computational costs but also facilitates the learning of MLPs that map dynamic skill prefixes. Specifically, we add a pair of trainable continuous prefix tokens $\left\{P_g, P_g^{'}\right\}$ for encoder and decoder. When doing attention calculations in the i-th layer, the current K, V vectors will be updated:

$$K_i^{'} = [P_{g,i,K}; K_i], V_i^{'} = [P_{g,i,V}; V_i], \quad (4)$$

where $K_i^{'}, V_i^{'} \in R^{(L+M)\times d}$, L is the length of the general prefix, d is the dimension of the hidden layer, M is the number of tokens associated with keys and values.

Dynamic Skill Prefix. We introduce dynamic skill prefixes to make the model aware of the skills corresponding to the current question. Specifically, we use MLPs to map dynamic skill representation ds_{v_i} to dynamic skill prefix form $\{P_d, P_d'\}$, which is concatenated to K and V in each attention layer of encoder and decoder, respectively, together with the previous general prefix:

$$P_d = MLP(ds_{v_i}), \quad (5)$$

$$K_i^{''} = [P_{d,i,K}; P_{g,i,K}; K_i], V_i^{''} = [P_{d,i,V}; P_{g,i,V}; V_i], \quad (6)$$

where $K_i^{''}, V_i^{''} \in R^{(L_d+L+M)\times d}$, and L_d is the length of the dynamic skill prefix.

Our method differs from GTEE-DYNPREF [Liu *et al.*, 2022b] , which obtains the current context-specific dynamic prefixes from multiple task prefixes. And these multiple task prefixes need to be trained separately. While our approach just needs to train the mapping function from the dynamic skill representations to the dynamic skill prefixes, which eliminates the need for multiple training sessions. In addition, our dynamic skill prefixes introduce extra dataset features that are extracted from the training set data instead of being randomly initialized.

Skill-aware Keyword Focus. Humans can use existing similar memories to quickly focus on the core of the question and some can even affect the answer. Since skill representations mentioned earlier are obtained by extracting features from the training set, they can also be perceived as abstract memory storage for this particular skill. Therefore, by referring to the studies about information interaction [Zhang *et al.*, 2019; Chen *et al.*, 2022b; Zhang *et al.*, 2021], we introduce skills to help the model focus on the key points of the question. Considering that different skills focus on different aspects, we use various skill representations to re-weight the individual words of the question which have been encoded by the encoder. Specifically, by modifying the method of BiDAF [Seo *et al.*, 2017], we use each skill representation to do attention calculations for each word of the question to get the weight assigned by each skill to the current word:

$$S_{ij} = \alpha(w_i, s_{v_j}), \quad (7)$$

where S_{ij} represents the similarity between the i-th question word and the j-th skill representation. α is a function that can be learned to compute the similarity of two input vectors. After each word in the question gets the weight assigned to it by different skills, each word takes the maximum weight assigned to it as the final weight assigned to the word:

$$b = softmax(max_{col}(S)), \quad (8)$$

where b represents the final weight assigned to each question word, S represents the similarity matrix of question words and skill representations, max_{col} denotes the function that takes the largest element in a column.

Dataset	CSQA2			CSQA	
skill	train	dev	test	train	dev
causality	5.71%	5.63%	6.47%	4.39%	3.85%
temporal	8.85%	9.64%	8.77%	5.37%	5.16%
plausibility	11.29%	12.32%	11.93%	1.75%	1.64%
comparison	13.72%	14.44%	14.84%	1.28%	0.98%
negation	17.57%	17.20%	17.43%	13.23%	13.19%
capable	22.28%	25.38%	24.79%	13.20%	14.17%
meronymy	33.37%	33.81%	32.03%	23.76%	24.08%
unknown	22.05%	19.56%	20.66%	52.05%	52.99%

Table 2: Skills and their frequency in datasets (an example may involve more than one skill).

4 Experiments

In this section, we first present the profile of datasets and some Closed-book CQA methods. After that, by conducting extensive experiments on DSCQA and comparing with various baselines, we explore the following questions:

- **Q1:** How does DSCQA perform compared to other Closed-book CQA methods?
- **Q2:** How well do the various modules in DSCQA work?
- **Q3:** How does DSCQA specifically perform on questions across skill categories?
- **Q4:** What is the effect of different dynamic skill prefix lengths on the results?
- **Q5:** How does DSCQA correctly answer confusing questions compared to basic models?

The code is at https://github.com/BAOOOOOM/DSCQA.

4.1 Dataset

We use two widely-used commonsense datasets, i.e., CommonsenseQA (CSQA [Talmor *et al.*, 2019]) and CommonsenseQA 2.0 (CSQA2 [Talmor *et al.*, 2021]), as benchmarks. CommonsenseQA is a widely-used commonsense dataset that consists of 12,247 multiple-choice questions. Its questions and answers are based on related concepts in ConceptNet [Speer *et al.*, 2017], including two other concepts connected to the concepts in question and two artificially created concepts are used as wrong options. CommonsenseQA 2.0 is a more challenging dataset for answering commonsense questions. It includes 14,343 yes or no questions that are made by people in order to make it hard for AI to get the answers right. In particular, since our research focuses on using skills to help with commonsense question answering, we use a regular matching method with certain skills to preliminarily classify questions in CSQA and CSQA2, and the statistics are demonstrated in Table 2.

4.2 Comparison Methods

Our study focuses on commonsense question answering in the Closed-book setting. Therefore, we have selected the following methods that do not depend on retrieving external knowledge for comparison.

- **Direct Inference under Fine-tuning.** We use T5-large [Raffel *et al.*, 2020] to perform inference directly by fine-tuning on the corresponding training set without introducing relevant external knowledge.
- **Prefix-tuning [Li and Liang, 2021].** It is a novel prompt-based approach that keeps the parameters of language model frozen and fine-tunes a small continuous vector of prefixes during the training.
- **Self-talk [Shwartz *et al.*, 2020].** This method generates a sequence of information search questions by combining the current question with a predefined question prefix. After that, these questions are used to inquire zero-shot language models to produce additional relevant background knowledge, which is passed along with the questions to the pre-trained model for fine-tuning.
- **GPT-3 [Brown *et al.*, 2020].** This method uses some demonstrative prompts to derive fixed GPT-3 to generate relevant background knowledge. The knowledge and questions are then used together for model training and inference. Following the implementation of ALEAP [Wang *et al.*, 2022], we use ten sampled knowledge spans as implementation questions.
- **ALEAP [Wang *et al.*, 2022].** It utilizes fixed GPT-3 to generate question-related knowledge and selects suitable knowledge to help solve the question by alternately optimizing the knowledge selector and answer predictor.

4.3 Implementation

In our experiment, we use T5-large [Raffel *et al.*, 2020] as our backbone, which has 1024 dimensions hidden representations. Considering that the datasets have different answer forms, we treat CSQA as a generation problem and CSQA2 as a classification problem. We take the best result from the training process as the final result. We use AdamW [Loshchilov and Hutter, 2019] as the optimizer and set the learning rate to 1e-5. We set the maximum length of the model input to 64. We use fine-tuned Sentence-T5 [Ni *et al.*, 2022] as our context encoder to get the representation of the question words and use the obtained representation to represent the sentence and skill representations later. We modify the OpenPrompt [Ding *et al.*, 2022] and use it as a framework for a series of prefix-tuning in Section 3.5. For general prefixes, the prefix length is set to 100, and its dropout rate is set to 0.5. The number of attention heads is set to 12 for question-skill attention and 8 for skill-question attention.

In addition, Prefix-tuning [Li and Liang, 2021] mentioned that directly optimizing the prefixes would lead to unstable and degraded performance, so we followed their advice to use multilayer perceptron (MLP) to reparameterize general prefixes. To be specific, we initialize a smaller matrix $\widetilde{P}$ and then reparameterize the prefix matrix $P = MLP(\widetilde{P})$. Once the training is complete, we will keep only the final prefix matrix P and discard the intermediate matrix $\widetilde{P}$.

4.4 Main Results (Q1)

The experimental results for CSQA and CSQA2 are presented in Table 3. In general, our model outperforms other baselines

Dataset	CSQA2		CSQA
Inference model	dev	test	dev
T5-large	58.04	56.09	65.68
Prefix-tuning+T5-large	57.54	56.73	66.26
GPT-3	58.56	56.98	67.23
Selftalk	55.88	54.87	65.03
ALEAP	58.72	57.58	67.32
DSCQA	**59.11**	**58.51**	**68.47**

Table 3: Results of models without introducing external knowledge sources. We emphasize the best scores in **bold**.

Dataset	CSQA2		CSQA
Inference model	dev	test	dev
DSCQA	**59.11**	**58.51**	**68.47**
- w/o dynamic prefix	58.68	57.02	67.57
- w/o re-weight	58.84	57.22	67.40
- w/o skill classification	58.32	55.88	67.90
- static skill labels	58.44	56.53	67.08

Table 4: Ablation study on the DSCQA framework.

under the same setting. For CSQA2 dataset, DSCQA outperforms the Prefix-tuning T5-large baseline by 1.78% and the previous strong baseline ALEAP [Wang *et al.*, 2022] by 0.93% on the test set. For CSQA dataset, our method has 2.21%, 1.15% performance improvement compared with Prefix-tuning T5-large and ALEAP, respectively. By making the model perceive the skill corresponding to the question, so as to notice the notes or tricks of the corresponding skill, the performance of DSCQA is improved compared with other methods. In particular, compared with CSQA2, DSCQA has a greater improvement on CSQA dataset. We consider that the questions in CSQA2 are generated by humans against AI, the questions are more difficult and thus the skills in the questions are more difficult to identify. Moreover, compared with previous methods, our model no longer requires the assistance of large-scale PLMs (e.g., GPT-3), but achieves better results, which reflects that only using PLMs to generate knowledge has limited performance improvement, and the perception and use of various skills is necessary.

4.5 Ablation Study (Q2)

To assess the efficacy of the individual modules within our model, we remove the dynamic skill prefix and the question word re-weight part in perception and emphasis module. In addition, the effectiveness of the dynamic skill prefix is further investigated by experimentally modifying the skill prefixes. Specifically, we explored four additional configurations in comparison to our original approach. For the setting of removing dynamic skills, it is to investigate the effectiveness of making the model aware of the current skill. For the setting of removing question words re-weighting, it is to explore the effectiveness of skills in helping the model focus on the question focus. For the configuration that do not use skill classification, its purpose is to demonstrate that the improvement of DSCQA mainly depends on the skills, rather than the non-answer question features extracted from the training set. For the configuration which uses static skill labels instead of dynamic skill prefixes, it is to explore whether skill representations that include dataset features have more question affinity than pure text labels. The experimental results are presented in Table 4.

When we remove the dynamic prefixes, we see a 0.43%, 1.49% and 0.90% decline in CSQA2 development set, CSQA2 test set, CSQA development set, respectively. This suggests that informing the model skills for the current question and applying it to the inference process can help improve performance. After removing the module that re-weight the question words, performance shows declines of 0.27%, 1.29% and 1.07%. The results demonstrate that skills can help questions find key information.

For the setting that does not use skill classification, we treat all questions of the training set as a large skill category, and then obtain a overall skill representation through the commonsense skill extraction module. Following this, we utilize the overall skill representation to replace the individual skill representation for the experiment. In this way, the features of the training set are preserved, but no skill classification is performed. For the setting of static skill labels, we make use of each skill seed to classify the questions through regular matching. We then incorporate each pseudo label into the model together with the question. In particular, since this pseudo labels corresponding to the questions have no semantic information, we do not re-weight the question words in this setting. As shown in Table 4, the performance of the CSQA2 development set, CSQA2 test set, and CSQA development set decreases by 0.79%, 2.63% and 0.57% respectively after removing the skill classification. It indicates that only extracting features from the training set without skills is of limited help or even harmful to solving questions. In addition, the performance of using static skill labels decreases by 0.67%, 1.98% and 1.39% compare to DSCQA without using question word re-weighting, which indicates that extracting features from the training set can help skills guide question reasoning. All in all, with the above experimental analyses about DSCQA and its ablation variants, we thoroughly demonstrate the effectiveness of different modules.

4.6 Performance in Various Skill Categories (Q3)

In order to better evaluate the impact of our model on question using skills, we further compare the performance of DSCQA and Prefix-tuning in various skill categories on CSQA2 and CSQA development sets. Table 5 illustrates the accuracy results of DSCQA and Prefix-tuning on various skill class questions. We observe an overall performance increase in accuracy on questions across skill categories, which shows that our approach has a positive effect on helping to answer questions in the majority of skill categories. In particular, we observe that for questions involving the comparison category, the performance of DSCQA is degraded compared to Prefix-tuning in CSQA2. We consider it may be because the com-

Dataset	CSQA2			CSQA		
skill type	Prefix-tuning	DSCQA	△	Prefix-tuning	DSCQA	△
causality	56.03	**58.16**	2.13	58.70	**60.87**	2.17
temporal	55.31	**57.08**	1.77	68.52	**75.93**	7.41
plausibility	62.82	**65.71**	2.89	65.00	**75.00**	10.00
comparison	**56.68**	53.41	-3.27	**66.67**	**66.67**	0.00
negation	56.78	**61.12**	4.34	65.91	**68.18**	2.27
capable	55.20	**59.28**	4.08	67.71	**69.79**	2.08
meronymy	58.88	**59.01**	0.13	61.61	**63.98**	2.37
unknown	56.19	**59.79**	3.60	67.53	**69.46**	1.93

Table 5: Accuracy for various skills on CSQA2 and CSQA. △ denote the increase or decrease of the performance.

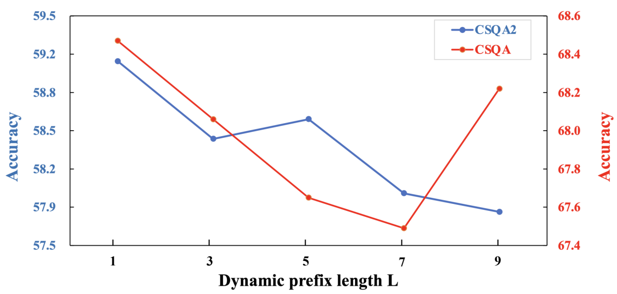

Figure 3: Performance of dynamic skill prefix lengths.

parison type questions involve comparing attributes to each other, which requires the model to have a deeper knowledge of the individual attributes. Therefore, for comparison skill questions, it is not enough to just let the model know that the question involved corresponds to the skill and highlight the keywords related to the skill. At the same time, it is observed that the accuracy of DSCQA for questions with skill category unknown is also improved by 3.60% and 1.93% on CSQA2 and CSQA, respectively. This indicates that extracting information from questions in each skill category is also helpful for solving questions where the skill category is difficult to determine. These results based on skill type are consistent with our assumption that the introduction of skills in commonsense question answering improves performance.

4.7 Hyper-parameter Analysis (Q4)

In this part, we investigate the influences of different dynamic skill prefix lengths on the performance on the CSQA2 of CSQA datasets. Figure 3 shows how the accuracy varies with the length of the dynamic skill prefix. We can see that the accuracy progressively decreases as the skill prefix length grows. We analyze that two factors may cause this. First, the length of the question itself is relatively short, and too long dynamic skill prefixes will cause the model to focus too much on the skill and ignore the question itself. Second, the process of expanding dynamic skill prefixes requires learning, which increases the difficulty of learning other modules. Therefore, we choose 1 as the length of the dynamic skill prefix in the DSCQA framework.

Question	Answer	Prediction
Peter is always a name of a man?	No	Prefix-tuning: Yes **DSCQA: No**
Where is likely to not just have a kosher restaurant?	New York city	Prefix-tuning: Jerusalem **DSCQA: New York city**

Figure 4: Examples about DSCQA can predict correctly, but Prefix-tuning made the wrong predictions.

4.8 Case Study (Q5)

We use case studies to further demonstrate the role of skills in question-solving. As shown in Figure 4, the question *"Peter is always a name of a man?"* belongs to the plausibility category question, and the word *"always"* in the question determines the answer. Prefix-tuning method predicted answer is *"Yes"*, while DSCQA predicted answer is *"No"*. Clearly, our method predicted the correct answer. Since the pre-trained model has seen the male name *"Peter"* more often during training, it is more likely to assume that the statement is true. However, if the model does not realize that the question involves the *"plausibility"* skill, then it is likely to ignore the inherent requirements and not make the right choice. For question *"Where is likely to not just have a kosher restaurant?"*, Prefix-tuning may overlook the *"negation"* skill corresponding to the word *"not"* and thus chooses *"Jerusalem"* which is more semantically similar to *"kosher restaurant"*. These two examples demonstrate that after being endowed with skill information, the model is more likely to recognize the underlying requirement and thus is more likely to pick out the correct answer.

5 Conclusion

In this paper, we studied the use of skills in Closed-book CQA and proposed the DSCQA model. Specifically, we extracted features from the training set to form individual skill representations that help questions identify the required commonsense skills. We introduced dynamic skill prefixes to make the question aware of the current corresponding skill during inference. In addition, we used skills to re-weight the question words to help the model find keywords in the question. Experimental results showed that introducing skills can help commonsense reasoning. To the best of our knowledge, we are the first to attempt to use skills in Closed-book CQA tasks.

Acknowledgements

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2021YFF0901003). This work was supported by Ant Group through CCF-Ant Research Fund.

References

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[Chen *et al.*, 2022a] Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Chen *et al.*, 2022b] Liyi Chen, Zhi Li, Tong Xu, Han Wu, Zhefeng Wang, Nicholas Jing Yuan, and Enhong Chen. Multi-modal siamese network for entity alignment. In *Proc. of KDD*, 2022.

[Clive *et al.*, 2022] Jordan Clive, Kris Cao, and Marek Rei. Control prefixes for parameter-efficient text generation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 363–382, 2022.

[Ding *et al.*, 2022] Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. Openprompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, 2022.

[Jung *et al.*, 2022] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279. Association for Computational Linguistics, 2022.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[Liu *et al.*, 2020] Ye Liu, Han Wu, Zhenya Huang, Hao Wang, Jianhui Ma, Qi Liu, Enhong Chen, Hanqing Tao, and Ke Rui. Technical phrase extraction for patent mining: A multi-level approach. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1142–1147. IEEE, 2020.

[Liu *et al.*, 2022a] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, 2022.

[Liu *et al.*, 2022b] Xiao Liu, He-Yan Huang, Ge Shi, and Bo Wang. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, 2022.

[Liu *et al.*, 2023] Ye Liu, Han Wu, Zhenya Huang, Hao Wang, Yuting Ning, Jianhui Ma, Qi Liu, and Enhong Chen. Techpat: Technical phrase extraction for patent mining. *ACM Transactions on Knowledge Discovery from Data*, 2023.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[Moore, 2006] Chris Moore. *The development of commonsense psychology*. Lawrence Erlbaum Associates Publishers, 2006.

[Ni *et al.*, 2022] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, 2022.

[Onoe *et al.*, 2021] Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. Creak: A dataset for commonsense reasoning over entity knowledge. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[Petroni *et al.*, 2019] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, 2019.

[Puerto *et al.*, 2023] Haritz Puerto, Gözde Şahin, and Iryna Gurevych. MetaQA: Combining expert agents for multi-skill question answering. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3566–3580, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[Ran *et al.*, 2019] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. Numnet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019*

Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2474–2484, 2019.

[Roberts *et al.*, 2020] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, 2020.

[Seo *et al.*, 2017] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*, 2017.

[Shwartz *et al.*, 2020] Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, 2020.

[Speer *et al.*, 2017] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[Talmor *et al.*, 2019] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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)*, pages 4149–4158, 2019.

[Talmor *et al.*, 2021] Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. Commonsenseqa 2.0: Exposing the limits of ai through gamification. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[Trivedi *et al.*, 2022] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Teaching broad reasoning skills for multi-step qa by generating hard contexts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6541–6566, 2022.

[Wang *et al.*, 2022] Wenya Wang, Vivek Srikumar, Hannaneh Hajishirzi, and Noah A. Smith. Elaboration-generating commonsense question answering at scale. *ArXiv*, abs/2209.01232, 2022.

[Wang *et al.*, 2023] Kehang Wang, Qi Liu, Kai Zhang, Ye Liu, Hanqing Tao, Zhenya Huang, and Enhong Chen. Class-dynamic and hierarchy-constrained network for entity linking. In *Database Systems for Advanced Applications: 28th International Conference, DASFAA 2023, Tianjin, China, April 17–20, 2023, Proceedings, Part II*, pages 622–638. Springer, 2023.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[Yang *et al.*, 2018] Yang Yang, Yi-Feng Wu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. Complex object classification: A multi-modal multi-instance multi-label deep network with optimal transport. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2594–2603, London, UK, 2018.

[Ye *et al.*, 2022] Deming Ye, Yankai Lin, Peng Li, Maosong Sun, and Zhiyuan Liu. A simple but effective pluggable entity lookup table for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 523–529, 2022.

[Yoran *et al.*, 2022] Ori Yoran, Alon Talmor, and Jonathan Berant. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6016–6031, 2022.

[Yu *et al.*, 2022] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11630–11638, Jun. 2022.

[Yue *et al.*, 2022] Linan Yue, Qi Liu, Yichao Du, Yanqing An, Li Wang, and Enhong Chen. Dare: Disentanglement-augmented rationale extraction. *Advances in Neural Information Processing Systems*, 35:26603–26617, 2022.

[Zhang *et al.*, 2019] Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5773–5780, 2019.

[Zhang *et al.*, 2021] Kai Zhang, Qi Liu, Hao Qian, Biao Xiang, Qing Cui, Jun Zhou, and Enhong Chen. Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):377–389, 2021.

[Zhang *et al.*, 2022] Kai Zhang, Kun Zhang, Mengdi Zhang, Hongke Zhao, Qi Liu, Wei Wu, and Enhong Chen. Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3599–3610, 2022.