Distinguish Before Answer: Generating Contrastive Explanation as Knowledge for Commonsense Question Answering

Qianglong Chen1,2; Guohai Xu1; Ming Yan2; Ji Zhang2; Fei Huang2; Luo Si2; Yin Zhang1†

1College of Computer Science and Technology, Zhejiang University, China
2DAMO Academy, Alibaba Group, China,
{chenqianglong,zhangyin98}@zju.edu.cn, 
{guohai.xgh,ym119608,zj122146,f.huang,luo.si}@alibaba-inc.com

Abstract

Existing knowledge-enhanced methods have achieved remarkable results in certain Q&A tasks via obtaining diverse knowledge from different knowledge bases. However, limited by the properties of retrieved knowledge, they still have trouble benefiting from both the knowledge relevance and distinguishment simultaneously. To address the challenge, we propose CPACE, a Concept-centric Prompt-based Contrastive Explanation Generation model, which aims to convert obtained symbolic knowledge into the contrastive explanation for better distinguishing the differences among given candidates. Firstly, following previous works, we retrieve different types of symbolic knowledge with a concept-centric knowledge extraction module. After that, we generate corresponding contrastive explanation using acquired symbolic knowledge and explanation prompt as guidance for better modeling the knowledge distinguishment and interpretability. Finally, we regard the generated contrastive explanation as external knowledge for downstream task enhancement. We conduct a series of experiments on three widely-used question-answering datasets: CSQA, QASC, and OBQA. Experimental results demonstrate that with the help of generated contrastive explanation, our CPACE model achieves new SOTA on CSQA (89.8% on the testing set, 0.9% higher than human performance), and gains impressive improvement on QASC and OBQA (4.2% and 3.5%, respectively).

1 Introduction

In recent years, a large number of knowledge enhanced pre-trained language models (KE-PLMs) (Zhang et al., 2019; Liu et al., 2020; Wang et al., 2021b,c) have been proposed to improve performance on a wide variety of NLP tasks (Wei et al., 2021). However, the implicit knowledge learned in PLMs can not be effectively used for these knowledge-driven QA tasks, especially in commonsense question answering. Some works (Lv et al.,

![Figure 1: A motivating example for our CPACE model.](image-url)

To provide more distinguishing information, we can convert the acquired symbolic knowledge into contrastive explanation and use them for inference enhancement.

2020; Wang et al., 2020; Chen et al., 2020; Xu et al., 2021; Wang et al., 2021a) explicitly retrieve knowledge from different knowledge sources, including WordNet (Miller, 1995), Wikidata (Vrandečić and Krötzsch, 2014) and ConceptNet (Speer et al., 2017), then integrate them into downstream models for Q&A. These methods enjoy the ability to utilize diverse knowledge, but inevitably introduce irrelevant or even noisy knowledge which will hurt the performance of model. Other works...
consider PLMs as knowledge bases (Petroni et al., 2019; Roberts et al., 2020; Heinzerling and Inui, 2021; Wang et al., 2021a), which elicit potential knowledge via prompt from PLMs (Paranjape et al., 2021; Liu et al., 2022). These approaches can obtain relevant knowledge from PLMs, however the generated knowledge from PLMs is generally common and lacks specific and distinguishing information for enhancement. It is an important direction to explore “how to provide discriminative information to models to help them distinguish candidates before answering?”.

Inspired by previous studies (Chen et al., 2021; Paranjape et al., 2021; Jacovi et al., 2021), contrastive explanation can provide the information to explain “WHY A NOT B” for given input and prediction, which naturally has distinguishing property. As shown in Figure 1, given question, candidates and retrieved symbolic knowledge, we generate contrastive explanation for each candidate to provide discriminative information among them for inference enhancement.

Therefore, in this paper, we propose a Concept-centric Prompt-based Contrastive Explanation generation (CPACE) model, a distinguish before answer architecture, to obtain high-quality incorporated knowledge and distinguish the differences among candidates. Specifically, our model consists of three parts, namely symbolic knowledge acquisition module, contrastive explanation generation module and explanation enhanced inference module. Firstly, given the question and candidates, we use a trained concept recognizer to detect concepts appearing in input. Then, with identified concepts, we extract diverse symbolic concept-centric knowledge from different types of knowledge bases. After that, we take the retrieved knowledge and a pre-defined explanation prompt as guidance for a fine-tuned generative pre-trained language model to generate contrastive explanation. The process of generation can filter irrelevant knowledge and convert selected symbolic knowledge into more specific and distinguishing information according to question and candidates. Finally, we use the generated contrastive explanation as external knowledge for enhancement. It is worth noting that contrastive explanation, as the final form of incorporated knowledge, not only meet distinguishing property, but also makes it easier for human to understand and is better interpretable.

The contributions are summarized as follows:

- Based on previous exploration of contrastive explanation, we first propose a CPACE model to unify the retrieved knowledge into contrastive explanation, which can distinguish the difference among answers before prediction.
- To better adapt contrastive explanation to question answering tasks, we develop a concept-centric prompt-based generator, which can leverage concept-centric knowledge and explanation prompt as guidance.
- Our CPACE model achieves new SOTA on CSQA leaderboard, which surprisingly surpasses human performance. Experimental results demonstrate the generalization of our methods on QASC and OBQA datasets and the effectiveness of contrastive explanation as another type of unified knowledge form for knowledge enhancement.

2 Task Formulation and Overall Workflow

Here, we introduce the commonsense question answering task and the workflow of our CPACE model. Given a question stem $Q$, the task is to find the correct answer $a$ from a finite set of choices $A = \{a_1, a_2, ..., a_n\}$. As shown in Figure 2, our approach can be divided into three steps. The first step is symbolic knowledge acquisition, we build a concept recognizer to identify a concept set $C$ from the given question $Q$ and candidates $A$, then we take them as queries to extract diverse symbolic knowledge $K_{\text{symbolic}}$ from several knowledge bases $KBs$, as shown in Section 3.1:

$$C = \text{Recognition}(Q, A)$$ (1)

$$K_{\text{symbolic}} = \text{Extraction}(C, KBs)$$ (2)

The second step is contrastive explanation generation, where we generate contrastive explanation $K_{ce}$ with CPACE generator, given $Q, A, K_{\text{symbolic}}, C$ and explanation prompt $P$, as shown in Section 3.2:

$$K_{ce} = \text{Generation}(Q, A, K_{\text{symbolic}}, C, P)$$ (3)

The final step is explanation enhanced inference, we obtain the predicted answers $a$ from a standard inference model enhanced with $K_{ce}$, as presented in Section 3.3:

$$a = \text{Inference}(Q, A, K_{ce})$$ (4)
3 Approach

3.1 Symbolic Knowledge Acquisition

3.1.1 Concept Recognition

Considering the concepts represent the key information of examples in semantic level, some works (Chen et al., 2021; Antognini and Faltings, 2021; Stowe et al., 2021) build a connection with external knowledge through concepts. Inspired by these studies, we employ a concept recognizer to detect the concepts from given question and candidates, which can ensure the retrieved symbolic knowledge is more concept-centric and relevant to the input in external knowledge extraction.

We first formulate concept recognition as a token-level sequence labeling task (Thorne et al., 2019), where 1 indicates a concept token and 0 indicates a background token. For the concept recognizer, we adopt RoBERTa-large as the encoder with a CRF layer. We construct the input sentence $S=[CLS]Q[SEP]A[SEP]$, where [SEP] is special token to separate question and candidates. Given a sentence $S = \{t_1, t_2, ..., t_n\}$, the task is to find a set of concepts $C = \{c_1, ..., c_m\}$. Limited by the scale of training corpus, we collect several similar datasets for concept recognizer training, including CommonGen (Lin et al., 2020), e-SNLI (Camburu et al., 2018) and CSQA (Talmor et al., 2019), all of which contained the annotated concepts or tokens in examples. The statistics of these datasets are shown in Table 1. While the CommonGen dataset is annotated to generate sentence with given concepts, we invert the target sentence into an input and use the given concepts as target. If there are more than 3 identified concepts in question stem, the top 3 concepts will be selected based on the score ranking mechanism for subsequent use. Otherwise, we select all identified concepts.

3.1.2 External Knowledge Extraction

After obtaining a group of concepts, we use them as anchors to retrieve relevant external symbolic knowledge. Following previous works (Chen et al., 2020; Xu et al., 2021), we choose ConceptNet and Cambridge Dictionary as knowledge bases for triples and definitions extraction.

**Triples Extraction** To extract relationships between concepts, being similar to Jession (2020), we find the path from the question concept to the candidate concept in ConceptNet. If there are more than one path, we choose the shortest. If there is no straightforward path between question concept and candidate concept, but we can find other triples in the ConceptNet with candidate concept. We define a score function and use it to compute the final score of each triples and chose the highest,

$$score_j = w_j \times \frac{N}{N_k}$$ (5)

where $w_j$ denotes the weight of $j_{th}$ triple in ConceptNet, $N$ is the total number of triples related to
candidate concepts, all $N$ triples related to candidate concepts are divided into multiple relation groups by clustering, and $N_k$ is the number of triples contained in $k$th relation group.

**Definitions Extraction** To extract definitions of concepts, following recent works (Chen et al., 2020; Xu et al., 2021), we obtain them from Cambridge Dictionary. For each concept, we choose its first definition entry in Dictionary as the description. When the closest matching definition entry is selected as the concept description in the dictionary, if there are multiple forms of definition entries, the priority order selected as the concept description is: the original form of the concept itself > the lemma form by Spacy > base word (last word). Finally, we concatenate the triples and concept definitions as external concept-centric knowledge, specifically, we take Triples [SEP] Definitions [SEP] as the Concept-centric Knowledge for contrastive explanation generation and downstream inference.

### 3.2 Contrastive Explanation Generation

In this part, we present how to generate contrastive explanations, given the question, candidates, and the retrieved knowledge, from data collection and generator training aspects.

**Data Collection** Firstly, for contrastive explanation generator training, the most important thing is to collect a certain number of annotated contrastive explanation datasets. We firstly collect some explanation-related datasets with the following principles in order: 1) whether the dataset directly contains contrastive explanations; 2) if not, can the dataset provide explanations for different candidates, i.e., positive and negative explanations; 3) if not, does the explanation of the dataset contain factual knowledge to distinguish different candidates or labels. Therefore, we choose the training set of ECQA (Aggarwal et al., 2021), eQASC (Jhamtani and Clark, 2020) and e-SNLI (Camburu et al., 2018) for generator training. The statistics of datasets are shown in Table 1.

**Generator Training** With the collected datasets, we train a contrastive explanation generator by fine-tuning a generative language model (GLM). In this work, we use BART-base as the backbone. In the fine-tuning stage, different from concatenating question stem and candidates in ECQA and eQASC, the hypothesis and premise sentence in e-SNLI are used as original input of GLM. The target is the explanation text. Moreover, different from previous works only consider original questions and candidates as input for fine-tuning, we also take the concepts and external symbolic knowledge to enhance the input for the prompt-based generation. As shown in Figure 3, the input is organized as follows: Task Prefix [SEP] [Question;Candidates] [SEP] Concepts [SEP] Concept-centric Knowledge [SEP] Explanation Prompt, where Task Prefix is “Generate the contrastive explanation for this question”, Concept-centric Knowledge represents extracted symbolic knowledge (triples and definitions of concepts) shown in section 3.1.2, and Explanation Prompt are the selected discrete prompts constructed by human, which are shown in Table 9.

Different from previous work (Paranjape et al., 2021) constructs Cloze prompt patterns for comparing the differences between two candidates, we consider whole contrastive explanation among all candidates and construct different discrete explanation prompts for guidance, for example, “Given concept sets, the difference among them is”. We use a list of templates $t_1, ..., t_p$ to generate a list of candidate explanations $e_1, ..., e_p$ for each input dur-
ing fine-tuning and select the best prompt for generation, $p$ denotes the number of the templates. It is worth noting that we firstly leverage the extracted symbolic knowledge and concepts to improve the quality of generated contrastive explanation, which is ignored in Paranjape et al. (2021).

### 3.3 Explanation Enhanced Inference

As shown in step 3 of Figure 2, given original question, we use the generated contrastive explanation as external knowledge to enhance the inference model, such as ALBERT and DeBERTaV3. Other types of knowledge can also be incorporated, which is optional. The objective function is defined as follows:

$$L_{ce} = -\frac{1}{T} \sum_{i=1}^{T} y_i \log \text{softmax}(h_i)$$

$$\text{softmax}(h_i) = \frac{\exp(h_i)}{\sum_{j=1}^{T} \exp(h_j)}$$

where $i$ represents the $i_{th}$ example, $h_i$ represents the hidden state after task-specific layer (MLP), $y_i$ represents the label of $i_{th}$ example, $T$ represents the total number of examples.

### 4 Experiments

#### 4.1 Datasets

**CSQA & ECQA** CommonsenseQA (CSQA) (Talmor et al., 2019) is proposed to explore the commonsense understanding ability of PLMs. To explore the interpretability of question-answering models, ECQA (Aggarwal et al., 2021) is proposed with the positive and negative explanations annotated for each question in CSQA. Here, we construct the positive and negative explanations as the ground truth contrastive explanation.

**QASC & OBQA** To further validate the generalization of our CPACE model, we evaluate the effectiveness of generated contrastive explanation on QASC (Khot et al., 2020) and OpenbookQA (OBQA) (Mihaylov et al., 2018). The statistics of above datasets are shown in Table 2.

#### 4.2 Experimental Setting

We choose BART-base (Lewis et al., 2020) as the pre-trained generative language model, which is the backbone of our contrastive explanation generator. For the framework, we use Pytorch 1.11. We use the AdamW (Loshchilov and Hutter, 2019) for optimization and set the warmup fraction to 0.1, and weight decay to 0.01. Meanwhile, we set the epoch to 10. For the learning rate, we search from $1e^{-5}$, $5e^{-5}$, $1e^{-4}$ and the best batch size we choose is 32. We set the max-length of output in the generator to 256. For the automatic evaluation, we use the ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) as the metric to measure the quality of generated explanation. For the inference models, we use ALBERT-xxlarge-v2 (Lan et al., 2020) and DeBERTaV3 (He et al., 2021) as backbone respectively, which are enhanced with contrastive explanation. For each experiment, we run 5 times and report the average and we use RTX6000 with 40G memory for training and inference.

#### 4.3 Baselines

**Pre-trained Language Models** For the comparison, we choose some PLMs as baselines to validate the effectiveness of backbone encoder, including DeBERTa (He et al., 2020, 2021), ALBERT (Lan et al., 2020), T5 (Raffel et al., 2020), UnifiedQA (Khashabi et al., 2020), RoBERTa (Liu et al., 2019) and BERT (Devlin et al., 2019).

**Knowledge Enhanced Methods** In knowledge-driven Q&A tasks, the most effective methods are external knowledge-enhanced. Here, we select representative approaches as baselines, including $BERT+OMCS$ \(^3\), $RoBERTa+MHGRN$ (Feng et al., 2020), $RoBERTa+AIR$ (Yadav et al., 2020),

---

\(^3\)https://drive.google.com/file/d/1sGJBV38aG706EAR7SF7lyWcqi9ocG9i

---

Table 1: Statistics of ECQA, eQASC, e-SNLI and CommonGen, used for CPACE generator training and concept identifier training.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Has Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECQA</td>
<td>9,741</td>
<td>1,221</td>
<td>✓</td>
</tr>
<tr>
<td>eQASC</td>
<td>8,134</td>
<td>926</td>
<td>✓</td>
</tr>
<tr>
<td>e-SNLI</td>
<td>549,369</td>
<td>9,843</td>
<td>✓</td>
</tr>
<tr>
<td>CommonGen</td>
<td>67,389</td>
<td>4,018</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 2: Statistics of CSQA, QASC and OBQA, used for QA task inference.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Number of candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSQA</td>
<td>9,741</td>
<td>1,221</td>
<td>1,140</td>
<td>5</td>
</tr>
<tr>
<td>QASC</td>
<td>8,134</td>
<td>926</td>
<td>920</td>
<td>8</td>
</tr>
<tr>
<td>OBQA</td>
<td>4,957</td>
<td>500</td>
<td>500</td>
<td>4</td>
</tr>
</tbody>
</table>

---

13211
of baselines, our CPACE model achieve the best performance on CSQA.

Specifically, in part 1 of Table 3, experimental results demonstrate the selection of pre-trained language models (PLM) is important for commonsense question answering. PLM with optimal pre-training tasks and large parameters achieves better results. While RoBERTa-large only achieves 72.5%, DeBERTa obtains 79.6% on CSQA, which adopts disentangled attention for decoding enhancement and has 1.5B parameters. In part 2 of Table 3, incorporating triples and concept definitions helps a lot to improve the performance of PLMs on CSQA. Compared with ALBERT, ALBERT+DESC-KCR achieves 83.3% on CSQA, which gains 7.8% improvement. Meanwhile, other works attempt to generate triples or relationships with PLMs, as shown in part 3 of Table 3. While ALBERT + PathGenerator only achieves 75.6% via dynamically generating structured evidence, our CPACE model achieves 87.4% in single model setting via generating contrastive explanations. Furthermore, while KEAR leverages external knowledge and retrieved training example for knowledge enhancement, our CPACE model outperforms KEAR and achieves first place on CSQA leaderboard.

Overall, while ALBERT achieves 73.5% on CSQA test set, existing knowledge-enhanced methods achieve 3.8%-7.2% improvement and our CPACE model improves over 13.9%. It indicates the generated contrastive explanation can be another efficient way for knowledge enhancement instead of retrieving triples, definitions, and training examples. It is noted that while KEAR joins human party via extra training examples retrieval and using over 39 models for ensemble, we only use 5 models for ensemble and propose a contrastive explanation generator, which is easier to follow.

4.5 Generalization of CPACE

To further measure the generality of CPACE, we evaluate our model on QASC and OBQA datasets. As shown in Table 4, we select some representative baselines for comparison, including UnifiedQA, RoBERTa+AIR and GenMC. Although ALBERT only gets 71.8% and 72.5% on QASC and OBQA, ALBERT + KD achieves 80.3% and 83.2% respectively, which only retrieves symbolic knowledge from KBs. With our CPACE model, we can further improve by 3.4% and 2.9%, respectively. The experimental results show that our CPACE model can...
Table 4: Results on development set of QASC and OBQA, demonstrating the generalization of CPACE.

<table>
<thead>
<tr>
<th>Model</th>
<th>QASC</th>
<th>OBQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>68.4</td>
<td>64.1</td>
</tr>
<tr>
<td>UnifiedQA</td>
<td>66.6</td>
<td>70.5</td>
</tr>
<tr>
<td>GenMC</td>
<td>67.6</td>
<td>71.6</td>
</tr>
<tr>
<td>ALBERT</td>
<td>71.8</td>
<td>72.5</td>
</tr>
<tr>
<td>ALBERT + KD</td>
<td>80.3</td>
<td>83.2</td>
</tr>
<tr>
<td>RoBERTa + AIR</td>
<td>81.4</td>
<td>81.7</td>
</tr>
<tr>
<td>CPACE</td>
<td><strong>83.7</strong></td>
<td><strong>86.1</strong></td>
</tr>
</tbody>
</table>

Table 5: Ablation study of generator on development set of CSQA. We adopt ALBERT as the inference model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BART</td>
<td>78.3</td>
</tr>
<tr>
<td>BART + Concept</td>
<td>79.1</td>
</tr>
<tr>
<td>BART + Explanation Prompt</td>
<td>82.4</td>
</tr>
<tr>
<td>BART + Concept-centric Knowledge</td>
<td>83.5</td>
</tr>
<tr>
<td>BART + All</td>
<td><strong>85.2</strong></td>
</tr>
</tbody>
</table>

be used not only for commonsense question answering but also for other open-domain Q&A. Meanwhile, we present the case study in Appendix C.

4.6 Ablation Study

Analysis of Contrastive Explanation Generator As shown in Table 5, we use BART-base as the backbone to evaluate the effectiveness of concepts, prefix prompt, and retrieved concept-centric knowledge (triples and definitions of concepts) in the generator. Only with the fine-tuned BART-base as the generator, the generated explanation enhanced inference model can achieve 78.3% on CSQA development set. Since concepts represent the key information of a given sentence, with identified concepts, the generator can get some benefits. When taking concepts as enhanced input, we can obtain 0.8% improvement. When taking explanation prompt as a formal constraint, we get an improvement of 4.1%, which fully shows the necessity of contrastive explanation prompt as constraint. Meanwhile, enhanced with the external concept-centric knowledge, we can gain 5.2% improvement, which indicates concept-centric knowledge is equally important in contrastive explanation generation. Finally, with the incorporated of above three kinds of knowledge, the inference model can be improved by 6.9%.

Analysis of Inference Encoder In this part, we use ALBERT-xxlarge-v2 and DeBERTaV3 as the inference encoder. As shown in Table 6, ALBERT achieves 73.8% on CSQA, the DeBERTa achieves 84.6%, which indicates a better inference backbone is of importance in the downstream task. Then, we take the concept, retrieved concept-centric knowledge and generated contrastive explanation as different types of extra knowledge to enhance the inference model, respectively. While concept can only bring about 1.5% and 0.2% improvement for ALBERT and DeBERTa, we can get 10.4% and 0.5% improvement through triples and concept definitions, respectively. With the generated contrastive explanation, we can get a great improvement, which is 11.4% and 3.3% respectively. It demonstrates that generated contrastive explanation is much more effective than retrieved symbolic knowledge. Compared with adding ground-truth contrastive explanation, which achieves 11.7% and 9.2% improvement respectively, there is still some room for improvement.

4.7 Evaluation of Contrastive Explanation

As shown in Table 7, following Shwartz et al. (2020), we present the human evaluation of generated contrastive explanation in four aspects, including 1) Relevant, whether the generated explanation is relevant to current example, 2) Factual, if the explanation contains factual evidence, 3) Distinguishing, if the explanation can provide distinguishing information to improve inference, and 4) Grammatical, whether the generated explanation is grammatical.

We sample 100 explanations from generated contrastive explanation on CSQA and evaluate the score of the generated explanation from above aspects. We use five students as annotators and report
Table 7: Human evaluation of generated contrastive explanation on development set of CSQA.

<table>
<thead>
<tr>
<th>Model</th>
<th>Relevant</th>
<th>Factual</th>
<th>Distinguishing</th>
<th>Grammatical</th>
</tr>
</thead>
<tbody>
<tr>
<td>BART</td>
<td>68.2</td>
<td>47.0</td>
<td>26.0</td>
<td>83.2</td>
</tr>
<tr>
<td>BART + Concept</td>
<td>71.0</td>
<td>48.2</td>
<td>28.2</td>
<td>83.6</td>
</tr>
<tr>
<td>BART + Explanation Prompt</td>
<td>72.4</td>
<td>48.6</td>
<td>29.1</td>
<td>83.7</td>
</tr>
<tr>
<td>BART + Concept-centric Knowledge</td>
<td>75.3</td>
<td>52.2</td>
<td>50.1</td>
<td>84.2</td>
</tr>
<tr>
<td>BART + All</td>
<td><strong>80.3</strong></td>
<td><strong>54.6</strong></td>
<td><strong>53.4</strong></td>
<td><strong>87.5</strong></td>
</tr>
</tbody>
</table>

the average. As we can see, taking the given example as input of BART-base, we only get high grammatical score but with low distinguishing score. When we take concept and explanation prompt for enhancement, it can slightly improve the the relevance and distinguishment. Enhanced with concept-centric knowledge, it can improve the distinguishing score over 20%, which is much more helpful. Furthermore, we can use all the above knowledge to get the best performance. Besides, we demonstrate automatic evaluation of contrastive explanation, which is shown in Appendix B.

5 Related Work

5.1 Knowledge Enhanced Methods
To alleviate the knowledge insufficiency problem, many knowledge-enhanced works have been proposed (Chen et al., 2022a; Wang et al., 2022; Liu et al., 2020; Wang et al., 2021c,b; Chen et al., 2020; Zhang et al., 2022; Chen et al., 2022b; Sun et al., 2021), which can be roughly categorized into explicit symbolic knowledge retrieval based and implicit knowledge generation based. In the former works, researchers (Lv et al., 2020; Chen et al., 2020; Xu et al., 2021) mainly focus on acquiring relevant knowledge from different knowledge bases, including ConceptNet (Speer et al., 2017), Wikipedia, and dictionaries. These methods enjoy the benefits of diverse knowledge but inevitably introduce irrelevant or even noisy knowledge. In the latter works, attempts (Petroni et al., 2019; Gao et al., 2021; Schick and Schütze, 2021; Zhong et al., 2021; Chen et al., 2022a) have been made to explore the possibility of using pre-trained language models (Devlin et al., 2019; Peters et al., 2018) as a knowledge base. While Petroni et al. (2019) first regard PLMs as knowledge bases, other works (Gao et al., 2021; Schick and Schütze, 2021; Zhong et al., 2021) use different prompt-based methods to elicit potential knowledge from PLMs. However, limited by the pre-training corpus, the generated knowledge from PLMs lacks specific information. To takes both advantages of symbolic knowledge retrieval and knowledge generation, we propose the distinguish before answer framework to generate contrastive explanation.

5.2 Contrastive Explanation
Contrastive explanations clarify why an event occurred in contrast to another, which are inherently intuitive to humans to both produce and comprehend (Jacovi et al., 2021). Compared to other explanations, Miller (2019) first suggests contrastive explanations are more effective in human learning. While Liang et al. (2020) first leverage expert annotated contrastive explanations for active learning to improve data efficiency, Jacovi et al. (2021) propose a method to produce contrastive explanations automatically in the latent space via input token/span for 3-label classification. Meanwhile, Chen et al. (2021) generate contrastive explanation with counterfactual examples for natural language inference. Different from Paranjape et al. (2021) uses human templates to prompt PLMs to generate contrastive explanations, we focus on generating contrastive explanation with retrieved symbolic knowledge to distinguish the candidates before prediction, which is ignored in previous works. Liu et al. (2022) and Huang et al. (2022) are the concurrent works, focusing on improving question answering with generated knowledge and clues.

6 Conclusion
In this paper, we propose a CPACE model, which unify the retrieved knowledge into contrastive explanation, to provide more discriminative information for model enhancement. We firstly consider concept-centric knowledge and explanation prompt as guidance for contrastive explanation generation. Our CPACE model achieves a new SOTA on CSQA leaderboard, which surprisingly surpasses human performance. In addition, we verify the effectiveness and generalization of CPACE on other datasets.
In the future, we will explore a unified contrastive explanation generation framework for NLP tasks.

7 Limitations

Limited by the scale of annotated contrastive explanation corpus, our CPACE model is only fine-tuned on approximate datasets selected with some designed principles. The performance of our method can be further improved with sufficient high-quality contrastive explanation annotated datasets over more NLP tasks. Moreover, in this paper, we mainly explore the effectiveness of the CPACE model for multiple-choice commonsense question-answering tasks, which is our goal, while previous retrieved-augmented methods cannot provide highly relevant knowledge or context for reasoning. Due to the fact that the contrastive explanation is designed to provide distinguishing information among given options \([a_1, a_2, \ldots, a_n]\) or labels, there are no given candidates or labels in generative commonsense question-answering tasks, therefore, our CPACE model cannot directly fit to other generative QA benchmark datasets. However, in our work, we provide some insights for future exploration, that is, generating question-specific distinguishing knowledge with a contrastive explanation generator can improve the performance and interpretation of current reasoning models. Meanwhile, although we validate the generalization of CPACE on other QA tasks, including QASC and OBQA, the effectiveness of our model in other NLP tasks requiring contrastive knowledge enhancement, such as open domain dialogue, needs to be further explored. In the future, following the CPACE model, we will explore a unified contrastive explanation generation framework for the generative commonsense question answering tasks via generating the chain-of-thoughts with a large generative language model-based generator, such as InstructGPT (Ouyang et al., 2022), BLOOM (Scao et al., 2022) etc., or generating top-N possible candidates and ranking them with distinguishing knowledge, which is beyond the scope of this paper to explore and is also our future work.

Acknowledgments

We thank the anonymous reviewers for their helpful comments on this paper. This work was supported by the NSFC under Grant No. 62072399, Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ23F020009, Chinese Knowledge Center for Engineering Sciences and Technology, MoE Engineering Research Center of Digital Library, Alibaba Group, Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, and the Fundamental Research Funds for the Central Universities.

References


Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022a. Dictbert: Dictionary description knowledge enhanced language


Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. **ELECTRA**: Pre-training text encoders as discriminators rather than generators. In *ICLR*.


A Details of Baselines

A.1 Baselines

BERT  BERT (Devlin et al., 2019) is the traditional pre-trained language model with mask language modeling and next sentence prediction pre-training tasks, which is used in most NLP tasks.

RoBERTa  RoBERTa (Liu et al., 2019) further optimizes BERT via pre-training on more corpus and removing next sentence prediction task.

ALBERT  ALBERT (Lan et al., 2020) is proposed to lower memory consumption and increase the training speed of BERT and focus on modeling inter-sentence coherence via a self-supervised loss, which is also widely used as backbone.

T5  To explore the landscape of transfer learning techniques for NLP, T5 (Raffel et al., 2020) introduces a unified framework that converts all text-based language problems into a text-to-text format and achieves new SOTA on many benchmarks.

UnifiedQA  UnifiedQA (Khashabi et al., 2020) is proposed to cross the boundaries among QA tasks via a single pre-trained QA model.

DeBERTa  To improve the BERT and RoBERTa models, He et al. (2020) propose DeBERTa with disentangled attention mechanism and an enhanced mask decoder. And they (He et al., 2021) further optimize DeBERTa via ELECTRA-style (Clark et al., 2020) pre-training with gradient-disentangled embedding sharing.

BERT+OMCS  BERT+OMCS finetunes BERT-large “whole word masking” model on the Open Mind Common Sense (OMCS) corpus used for creating ConceptNet.

TeGBERT  TeGBERT is a multi-modal learning method for commonsense reasoning, where paths are searched from a given question and choice with ConceptNet with triple scoring and triples are pretrained with kg2vec such as transE (Bordes et al., 2013).

RoBERTa+MHGRN  Feng et al. (2020) propose RoBERTa+MHGRN to equip pre-trained language models with a multi-hop graph relation network (MHGRN) module, which performs multi-hop, multi-relational reasoning over sub-graphs extracted from external knowledge graphs.

RoBERTa+AIR  RoBERTa+AIR (Yadav et al., 2020) is a method with alignment-based iterative retriever, which retrieves high-quality evidence sentences from unstructured knowledge bases and achieves new SOTA on QASC.

QA-GNN  Yasunaga et al. (2021) proposed QA-GNN to identify relevant knowledge from large KGs, and perform joint reasoning over the QA context and KG via relevance scoring and joint reasoning.

GenMC  Huang et al. (2022) propose a generation-enhanced multiple-choice question answering (MCQA) model, GenMC, which generates a clue from the question and then leverages the clue to enhance a reader for MCQA. It outperforms text-to-text models on multiple MCQA datasets.

ALBERT+Headhunter  Li et al. (2021) utilizes a self-attention module to re-distribute the importance of knowledge for common-sense reasoning, where top k commonsense knowledge are extracted from OMCS and they employ a Self-Attention module to interact with each triple representation.

ALBERT+KCR  Jesson (2020) propose a knowledge base method ALBERT+KCR to enhance text encoder, where they extract relevant triples from ConceptNet.

ALBERT+KD  ALBERT+KD combines ConceptNet and dictionary definitions for inference, where they use python’s networkx library to frame ConceptNet and use the Oxford dictionary to extract the definitions of concepts.

ALBERT+DESC-KCR  Xu et al. (2021) employ external entity descriptions to provide contextual information for knowledge understanding and retrieve descriptions of related concepts from Wiktionary and feed them as additional input to pre-trained language models.

ALBERT+PathGenerator  Wang et al. (2020) augment a general commonsense QA framework with a knowledgeable path generator, where the generator learns to connect a pair of entities in text with a dynamic and multi-hop relational path.

ALBERT+HGN  Yan et al. (2021) propose Hybrid Graph Network to jointly contextualize extracted and generated knowledge by reasoning over both within a unified graph structure.
KEAR  

Xu et al. (2022) propose KEAR to retrieve labeled examples from several question answering datasets and augment them as external knowledge for inference.

B Automatic Evaluation of Contrastive Explanation

As shown in Table 8, we also present the BLEU and ROUGE results of contrastive explanation generated with different types of knowledge. While the BLEU metric focuses on the precision of text, the ROUGE metric mainly evaluates the recall performance of generated text, which denotes the text can provide more relevant contextual information for given questions. Since we use the generated text as external knowledge for inference enhancement, the recall performance in the generation is much more important. With concepts and prompt constrained, we obtain better generated explanation text in the BLEU metric, while we can acquire better generated explanation text in the ROUGE metric with the external concept-centric symbolic knowledge (triples and definitions).

C Case Study

As shown in Table 10, we present a case study of our model. Given a question *Where can you find a magazine* and a set of candidates {A: doctor, B: bookstore, C: market, D: train, E: mortuary}, the true answer is *bookstore*. We first identify the concepts from the input example, including concepts in question stem and answer candidates. Then, we can extract the triples from ConceptNet. As shown in Table 10, both four candidates have same relation with *magazine*, which can not further filter the true answer. With adding the concepts descriptions, we can further distinguish the candidates with same relations. However, the description of concepts is not clear enough for explanation, compared with the annotated contrastive explanation. With our CPACE generator, we can obtain the generated contrastive explanation with concepts, symbolic knowledge and prompt enhanced, as Table 10 shown. Compared with the extracted triples and concept descriptions, the generated contrastive explanation is much easier to understand for user, while only with BART-base, we can only obtain candidates related explanation without considering question concepts. As shown in Table 10, the concepts and symbolic knowledge can help the generative language model concentrated on the key difference of candidates according to question concepts. Meanwhile, with the generated contrastive explanation for enhancement, we can infer the predicted answer is *bookstore*.
Table 9: Prompts we constructed for CPACE generator.

<table>
<thead>
<tr>
<th>Prompt Patterns</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given concept sets ([OPT_1, ..., OPT_n]), the difference among them is ...</td>
<td>CSQA QASC</td>
</tr>
<tr>
<td>Given ([OPT_1, ..., OPT_n]), while ([OPT_1]] can ..., ..., ([OPT_n]] can not ...</td>
<td>OBQA</td>
</tr>
<tr>
<td>The main difference among the concepts ([OPT_1, ..., OPT_n]] is ...</td>
<td></td>
</tr>
<tr>
<td>Given concepts, ([OPT_1]] can, but ([OPT_2, ..., OPT_n]] can not ...</td>
<td></td>
</tr>
<tr>
<td>Given concepts, while ([OPT_1]] can, ([OPT_2, ..., OPT_n]] can not ...</td>
<td></td>
</tr>
<tr>
<td>Given concepts, while ([OPT_1]] can not, however ([OPT_2, ..., OPT_n]] can ...</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Case study of CAPCE generator on CSQA dev set.

<table>
<thead>
<tr>
<th>Input Example</th>
<th>Where can you find a magazine. A. doctor B.bookstore, C.market D.train station E.mortuary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labels</td>
<td>B</td>
</tr>
<tr>
<td>Predicted Labels</td>
<td>B</td>
</tr>
</tbody>
</table>

**Step 1:**

**Step 1.1: Identified Concepts**

magazine, doctor, bookstore, market, train station, mortuary

**Step 1.2: Triples from ConceptNet**

magazine AtLocation doctor
magazine AtLocation bookstore
magazine AtLocation market
magazine AtLocation train station

**Step 1.2: Concept Description from Dictionary**

dr **doctor**: A physician; a member of medical profession; one who is trained and licensed to heal the sick or injured
bookstore **bookstore**: A store where books are bought and sold
market **market**: A gathering of people for the purchase and sale of merchandise at a set time.
train station **train station**: A place where trains stop for passengers to embark and disembark.
mortuary **mortuary**: of or relating to death or a funeral; funeral;

**Step 2:**

**Step 2.1: Generated Contrastive Explanation only with BART-base**

A store is a place where people can find magazines along with many other printed works.
Doctor is a physician.
You can buy something at market.
At train station, you can take a train.
Mortuary can be found in funeral.

**Step 2: Generated Contrastive Explanation with CPACE**

Many other printed works can be found at a bookstore.
You would find magazines along side.
Doctor is not a place where you can find various printed works.
At market, magazines are not found.
At train station, there are no printed works so no magazines are found. Mortuary do not have magazines.

**Golden Contrastive Explanation**

Bookstores have a variety of reading material including books, magazines, novels, etc.
The doctor is not a place. Market sells various items one of which is printed works.
Train stations do not have various printed works.
Mortuary has dead bodies.
ACL 2023 Responsible NLP Checklist

A  For every submission:

☐ A1. Did you describe the limitations of your work?
   Left blank.

☐ A2. Did you discuss any potential risks of your work?
   Left blank.

☐ A3. Do the abstract and introduction summarize the paper’s main claims?
   Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
   Left blank.

B  ☐ Did you use or create scientific artifacts?

   Left blank.

☐ B1. Did you cite the creators of artifacts you used?
   Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
   Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
   Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
   Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
   Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
   Left blank.

C  ☐ Did you run computational experiments?

   Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
   Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.
C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants’ demographic (e.g., country of residence)?
Left blank.

D3. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.