Pretrained language models have been shown to store knowledge in their parameters and have achieved reasonable performance in commonsense knowledge base completion (CKBC) tasks. However, CKBC is knowledge-intensive and it is reported that pretrained language models’ performance in knowledge-intensive tasks are limited because of their incapability of accessing and manipulating knowledge. As a result, we hypothesize that providing retrieved passages that contain relevant knowledge as additional input to the CKBC task will improve performance. In particular, we draw insights from Case-Based Reasoning (CBR) – which aims to solve a new problem by reasoning with retrieved relevant cases, and investigate the direct application of it to CKBC. On two benchmark datasets, we demonstrate through automatic and human evaluations that our End-to-end Case-Based Reasoning Framework (ECBRF) generates more valid, informative, and novel knowledge than the state-of-the-art COMET model for CKBC in both the fully supervised and few-shot settings. We provide insights on why previous retrieval-based methods only achieve merely the same performance with COMET. From the perspective of CBR, our framework addresses a fundamental question on whether CBR methodology can be utilized to improve deep learning models.

Table 1: Commonsense knowledge base tuples. Examples are from ConceptNet and Atomic.
(i.e., commonsense knowledge). An example of reporting bias from Table 1 is that people rarely say “when a person wins a big Jackpot, he/she will want to get its money” because it’s too obvious and meaningless to say. Therefore, instead of retrieving passages from the web, we propose that benefits can still be gained by retrieving relevant knowledge from a “case base” of existing commonsense knowledge tuples\(^1\) and using the retrieved knowledge as non-parametric knowledge (i.e., beyond that represented in the model parameters) to augment the current CKBC input example. In addition, to prevent ECBRF from overfitting to some commonly retrieved cases, we also propose random mask as a training strategy that randomly masks the retrieved cases during training, which functions similar to dropout (Srivastava et al., 2014) and further improves the performance of the framework. We also propose a reverse demonstration strategy that empirically can help the framework better utilize the retrieved knowledge.

Although past attempts suggest that similar retrieval-based methods cannot improve the performance of CKBC (Wang et al., 2021), on two benchmark datasets, we demonstrate through automatic and human evaluations that our End-to-end Case-Based Reasoning Framework (ECBRF) generates more valid, informative, and novel knowledge than (1) the state-of-the-art COMET model (Bosselut et al., 2019) for CKBC which employs no case retrieval, and (2) a baseline model that employs random case retrieval – on both fully supervised and few-shot settings. We also provide an analysis on why different conclusions are reached.

In addition, our framework draws insight from Case-Based Reasoning (CBR), and also has contributions to the CBR research.

CBR is a subject in classical AI that solves a new problem by reusing the solutions of retrieved seen similar problems stored in the case base (Aamodt and Plaza, 1994). CBR’s methodology has the procedure of four steps — case retrieval, reuse, revise and retain.

Past years of accomplishment in deep learning (DL) have led to enthusiasm in the CBR community to apply DL in the service of CBR. However, based on the observation that many challenges remain in DL where CBR has advantages (e.g. few-shot learning), some CBR researchers (Leake and Crandall, 2020) advocate using CBR to complement DL. However, past works on using CBR to complement DL only limit to shallow Neural Networks (NN) (Liao et al., 2018; Leake et al., 2021; Ye et al., 2021, 2022). The latest work even suggests that in many tasks NN itself outperforms CBR-complemented NN (Ye et al., 2022), which raises fundamental questions on whether CBR methodology is useful for DL.

Our work addresses this doubt by being the first to show a concrete implementation of the integration of the full methodology of CBR to PLM (as one typical model in DL) and show that the integration method can benefit from multiple steps in the methodology of CBR, and can lead to better performance over PLM itself in both fully supervised settings and few-shot settings on CKBC. Notably our proposed framework has a larger advantage in few-shot settings, where CBR methods typically have advantage. We also find that the generation of our framework is largely related to the retrieved case especially when they are similar, which exhibits strong case-based reasoning patterns. In addition, a detailed analysis of our framework from a CBR perspective is provided in section 6.

Our contributions can be summarized as follows: (1) Drawing insights from CBR, we introduce a new end-to-end framework for CKBC task. We also propose training strategies that can better utilize the retrieved knowledge. (2) We conduct extensive experiments on the CKBC task in various settings (e.g. fully supervised and few-shot), and the results consistently demonstrate that our proposed framework achieves substantial improvements over the state-of-the-art baseline methods. (3) From the perspective of the CBR community, whether CBR methodology can be used to improve DL models remains a fundamental research question. We address this doubt by being the first to show a concrete implementation of the integration of the full methodology of CBR to PLM (as a typical model in DL), and showing that such integration can achieve better performance than single PLM. A thorough analysis of the integration from a CBR perspective is also provided.

2 Related Work

Case-Based Reasoning  CBR is a subject in classical AI which consists of 4 sub-processes in its methodology: retrieve, reuse, revise and retain (Aamodt and Plaza, 1994). Leake and Crandall (2020) advocate using CBR to complement the

\(^1\)Initialized with tuples from training set or external data.
challenges in deep learning (e.g., few-shot learning). Appendix A.4 provides more detailed related works relevant to this line. Specifically, our framework is inspired by Watson (1999)’s proposal that compared to CBR being described as an artificial intelligence technology, it is better to describe CBR as a methodology for problem solving, that may use any appropriate technology. Here we treat CBR as a methodology and deep learning as technology that uses CBR as the general high-level process and deep learning as components of the process.

Commonsense Knowledge Base Completion
Here we mainly describe works that use text generation models for this task. Li et al. (2016) propose models to evaluate the full knowledge tuple rather than generate new knowledge. Saito et al. (2018) make an extension by proposing a joint model for the completion and generation of commonsense tuples. However, their work focuses on augmenting knowledge base completion model, rather than to increase coverage in commonsense knowledge base construction. Yao et al. (2019) and Malaviya et al. (2020) focus on link prediction and ranking of knowledge, which is a different task with our generative CKBC task. Sap et al. (2019) use LSTM (Hochreiter and Schmidhuber, 1997) to generate commonsense knowledge and Bosselut et al. (2019) further leverage pre-trained language models to generate commonsense knowledge. Gabriel et al. (2021) present the task of discourse-aware commonsense inference and proposes a memory-based model to generate commonsense knowledge that is more coherent with context. Wang et al. (2021) give an analysis on knowledge capacity, transferability, and induction of pretrained language models to perform generalizable commonsense inference. Da et al. (2021) analyze the few-shot learning ability of pretrained language models for CKBC task. Unlike these works, we propose a model that can improve the performance of generative CKBC tasks in both fully supervised settings and few-shot settings.

Language Model Prompting
First developed by the GPT series (Brown et al., 2020), retrieved data are used as augmented input to improve few-shot performance of remarkable large models. However, past research suggest that such in-context learning cannot improve the CKBC task (Wang et al., 2021), and we are the first to show how in-context learning is useful for CKBC. In addition, such large models are hard to obtain and Brown et al. (2020) do not explore the finetuning performance, neither do they explore the full CBR methodology’s effect on PLM. Gao et al. (2021) use prompting and also incorporate demonstrations into context to improve few-shot performance. Their work, however, only focuses on classification tasks and regression tasks, which is different from the CKBC. Similar to our work, Das et al. (2021) use retrieved cases as prompt to improve the performance of PLM. However, they only focus on question answering task and do not integrate the full methodology of CBR, missing important steps such as retain.

3 Task Definition
In the generative CKBC task, a knowledge data instance is represented as a tuple of subject, relation, and object: \((\text{sub}, \text{rel}, \text{obj})\). All \(\text{sub}\) and \(\text{obj}\) are in natural language phrases (Figure 1). \(\text{rel}\) can be used as either a special token or the corresponding natural language phrases (Bosselut et al., 2019). Here we use \(\text{rel}\) as natural language phrases. The task is that given a pair of \(\text{sub}\) and \(\text{rel}\), the goal is to generate the corresponding \(\text{obj}\).

4 Methodology
We start by formalizing our framework as a retrieve-then-predict generative process. Then in Section 4.2, we describe our ECBRF’s modules for the generative process in detail. Finally, we present a hybrid training strategy for better regularization.

Figure 1 describes our method. In the figure, “query” stands for a \(\text{sub}\) and \(\text{rel}\) pair which is used as input to ECBRF to generate \(\text{obj}\). “Case Base” is initialized with knowledge triples from the training set. “Cases” means the retrieved knowledge triples from the “Case Base”. “In-context demonstrations” stand for the retrieved cases that are used for input augmentation (concatenate with the query). The subject, relation, and object of the retrieved cases are sub-scripted with “r” (e.g., \(\text{sub}_r\)).

4.1 ECBRF’s Generative Process
ECBRF takes \(x\) as input and learns a distribution \(p(y|x)\) over possible outputs \(y\). Here \(x\) consists of \(\text{sub}\) and \(\text{rel}\), and \(y\) consists of \(\text{obj}\). More specifically, ECBRF decomposes \(p(y|x)\) into two steps: retrieve and predict. Given an input \(x\), we first retrieve similar cases \(z_1, z_2, \ldots\) (each case \(z_i\) consists of \(\text{sub}_i, \text{rel}_i, \text{obj}_i\)) from case base \(Z\), while \((x, y) \notin z_i\). We model this as a sample from the
We now detail two key components – the neural knowledge retriever, which models $p_q(z|x)$; and case-augmented encoder, which models $p_r(y|s,x)$. 

**Neural Knowledge Retriever** The retriever uses max inner product search (MIPS) to retrieve $z$. Specifically, the retriever is defined using a dense inner product model:

$$p_q(z|x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')}$$  \hspace{1cm} (4)

Then we condition on both the supporting set $s$ and the query $x$ to generate the output $y$, modeled as $p(y|s,x)$. To obtain the overall likelihood of generating $y$, we treat $s$ as a latent variable and marginalize $s$ via a top-$k$ approximation, yielding:

$$p(y|x) = \sum_{s \in \text{top}-k(p_q(.|x))} p_0(s|x)p_r(y|s,x)$$  \hspace{1cm} (3)

$\text{Case-Augmented Encoder}$ Given an input $x$ and a supporting set $s$, the case-augmented encoder defines:

$$p_r(y|s,x) = \prod_{i}^{N} p_r(y_i|x,s,y_{1:i-1})$$  \hspace{1cm} (6)

We use BART (Lewis et al., 2020a) and GPT-2 (Radford et al., 2019) as the base model for case-augmented encoder.
We evaluate ECBRF using two automatic common-sense knowledge base completion benchmarks — ConceptNet (Speer et al., 2017) and ATOMIC (Sap et al., 2019). Apart from using the entire train set for training, we also conduct experiments in the few-shot settings — where the model is only trained with 5 to 320 knowledge tuples\(^2\). Considering that fine-tuning on small datasets can suffer from instability (Dodge et al., 2020) and results may change dramatically given a new sample of data, we measure average performance across 3 different randomly sampled partitions of train data for few-shot experiments.

\(^2\)Note that for the few-shot settings, our ECBRF’s case base is also initialized with 5 to 320 tuples

## 4.3 Training Method

Since the purpose of in-context demonstration is only to provide ancillary information, the model should be able to predict \(obj\) \(w/\) or \(w/o\) it. Therefore here we design a specific training strategy for ECBRF — during training, we randomly mask out in-context demonstrations and only keep the \((sub, rel)\) query for some training examples with probability \(p_{\text{mask}}\). It functions similar to dropout to prevent overly relying on retrieved cases.

## 5 Experiments & Analysis

In this section, we introduce the experiment datasets and evaluation details, as well as experiment setups and the experiment results, measured with automatic and human evaluations.

### 5.1 Datasets and statistics

We evaluate ECBRF using two automatic common-sense knowledge base completion benchmarks — ConceptNet (Speer et al., 2017) and ATOMIC (Sap et al., 2019). Apart from using the entire train set for training, we also conduct experiments in the few-shot settings — where the model is only trained with 5 to 320 knowledge tuples\(^2\). Considering that fine-tuning on small datasets can suffer from instability (Dodge et al., 2020) and results may change dramatically given a new sample of data, we measure average performance across 3 different randomly sampled partitions of train data for few-shot experiments.

### 5.2 Evaluation Details

For automatic evaluation metrics, following Bosselut et al. (2019), we use BLEU-2, perplexity, and novelty metrics (including \(\%N/T\)-sro, \(\%N/T\)-o, and \(\%N/U\)-o). Following prior work (Luo et al., 2017), to calculate BLEU scores of generations, we firstly filter out all stop words using the standard nltk package in both generation and ground truth and then calculate BLEU \(^3\). Specifically for novelty metrics, we report the proportion of all generated tuples that are novel tuple (\(\%N/T\)-sro) (here novel means unseen in train set), have a novel \(obj\) (\(\%N/T\)-o), and the proportion of the set of unique \(obj\) in all generated objects (\(\%N/U\)-o).

In addition to automatic evaluation, we also perform human evaluation, including validness, informativeness, and preference score. For validness and informativeness, following Gabriel et al. (2021), the score is based on a 5-point Likert scale (with 5 points the highest score). For validness, following Gabriel et al. (2021), we judge the validness of the generated new knowledge by the likelihood of inferences based on a 5-point Likert scale (with 5 points the highest score). Specifically, obviously true (5), generally true (4), plausible (3), neutral or unclear or basically a repetition (sub-sentence) of the query (2), and doesn’t make sense (1). For informativeness, the rating standard is also based on a 5-point Likert scale. Specifically, rich in relevant details (5), has relevant details (4), some details are provided (3), basically a repetition (sub-sentence) of the query (2), unfinished generation (1). For preference score, We ask the human raters to compare the generations between ECBRF and COMET. Specifically, a valid generation with more information provided will be assigned 1.0 point, and a generation that is not valid or with less information will be assigned 0.0 instead. However, if the two generations perform comparably, both generations will be assigned 0.5 points.

Following Bosselut et al. (2019), for each experiment and for each model, we sample 100 generations for human evaluation. Each generation will be rated by three undergraduate students. During the evaluation the order of the two generations to be compared are randomized for each selection, therefore human raters have no clue on which choice is associated with which model.

\(^3\)During eval, we find that in generated object phrases there are many stop words like “the” and “to”, which do not contain much information but largely influence BLEU score.
Table 4: An example to show that BLEU is not a perfect metric for CKBC.

| sub: PersonX spends ___ working; rel: As a result, others feel Ground truth: ['happy', 'happy to have x in their life'] COMET’s generation: happy (BLEU: 31.62) ECBRF’s generation: satisfied with person’s work (BLEU: 0.00) |

Table 5: Novelty evaluation results using %N/T-sro, %N/T-o, and %N/U-o. Scores with significant improvement (≥ 10%) are boldfaced.

5.3 Experimental Setup

Baselines We use COMET (Bosselut et al., 2019) as our baseline. COMET is originally implemented with GPT (Radford et al.), a pretrained language model as the base model and uses subject and relation as direct input and uses the generation result as object. Here we compare two versions of COMET, one is GPT-2 (Radford et al., 2019) based and another is BART (Lewis et al., 2020a) based. Both GPT-2 and BART are more powerful pretrained language models than GPT. We leave the details of hyperparameters in Appendix A.1.
**subject**: PersonX wants to play with PersonY
**ref**: Before, this person needed

One retrieved case by ECBRF:
PersonX plays tennis with PersonY’s friend,
Before, this person needed, get a tennis racket

**COMET’s generation**: to have a game
**ECBRF’s generation**: to find a tennis court

<table>
<thead>
<tr>
<th>Table 6</th>
<th>An example to show how the retrieved cases influence the generated obj.</th>
</tr>
</thead>
</table>

**obj** is not in the ground truth set. Table 4 shows one typical example from ATOMIC that shows although ECBRF’s generations are reasonable, they only receive low BLEU scores (More examples in Appendix A.2). From our observations (representative example shown in Table 4), we find that ECBRF tends to generate more novel (shown in Table 5) and informative (shown in Table 3) sentences which is usually not included in the limited set of ground truth. As a result, BLEU with a limited set of ground truths will assign these novel and informative generations with low scores. Therefore we argue that although BLEU can provide good insights on the quality of generation, it is not a perfect metric for CKBC, and we should refer to human evaluation for more accurate evaluation.

In human evaluation including these examples (Table 3), we show that while receiving a lower BLEU when using large train sets (320-shot and full), ECBRF receives a higher human evaluation score in ATOMIC dataset. This difference indicates that when trained on ATOMIC, ECBRF tends to generate more novel but valid **obj** (too novel that many generated **obj** are out of the scope of the ground truth set), and the **novelty** results in Table 5 show consistency with this indication. Table 3 shows that ECBRF consistently outperforms COMET in **preference score**, **validness**, and **informativeness** in human evaluation.

Table 5 shows that the generated **obj** of ECBRF are generally **novel**, especially in ATOMIC dataset. The reason %N/T-sro score is always 100.0 in ATOMIC is that the (sub, rel) pairs in ATOMIC’s train set and test set do not overlap.

Now we answer the question on why previous research to use in-context demonstrations (ICD) with finetuning for CKBC reaches to the conclusion that it can only reaches comparable performance with COMET (Wang et al., 2021). Firstly, ICD is more useful in few-shot settings, while they only explore finetuning on the full train set. Secondly, with ICD models tend to generate more novel **obj**, which is more possible to not be included in the ground truth set. Thirdly, the stop words can largely influence the BLEU results, therefore stop words should be removed first. Fourthly, human evaluation is the most precise metric for the task, so that only evaluation with automatic metrics could be misleading.

### 5.5 Ablation Study of ECBRF

In Table 2, we show some ablation studies of ECBRF. “ECBRF w/o rand mask” stands for the ECBRF model without using random mask (\(p_{mask} = 0\)); “ECBRF w/o reverse demonstrations” stands for the ECBRF model without using reverse demonstrations; and “ECBRF w/ rand retrieval” represents an ECBRF model that randomly retrieves cases instead of MIPS search.

Both tables for automatic evaluation show that using “reverse demonstration” can generally lead to performance gain in both BLEU and perplexity, and using “random mask” can generally lead to better performance in BLEU (BLEU is the most important metric as its a generation task). From the tables, we also observe that ECBRF with MIPS retrieval consistently leads to better performance than ECBRF with random retrieval.

**5.6 Qualitative Analysis on How Retrieved Cases Influence **obj** Generation**

Table 6 shows one example of the generation of ECBRF and COMET (more examples are shown in Appendix A.3). It shows that ECBRF’s generation is related to the retrieved case, exhibiting the case-based reasoning ability of **reusing** the retrieved old experience to solve new problems.

### 6 Further Analysis from Perspective of CBR

CBR methodology contains 4 sub-processes, which are **retrieve**, **reuse**, **revise** and **retain**. More specifically, when given a new problem, the method first **retrieves** the most similar cases, then **reuses** the information in that case to solve the new problem by proposing a new solution, then **revises** the proposed solution according to the feedback of adopting it in real application scenarios (**revise** step usually involves human’s effort), and finally select high quality revised solutions together with their problems as new cases to **retain** to case base. We provide an analysis of how the high-level methodology of CBR (**retrieve**, **reuse**, **revise** and **retain**) shapes the design and how the selection details of CBR-
related components improve the performance of our end-to-end deep learning framework.

**Step 1: Retrieve**  Retrieve is an important step since the effectiveness of a CBR system largely relies on its ability to retrieve useful previous cases (Montazemi and Gupta, 1997). Here we use neural knowledge retriever (DPR) for retrieving the most similar cases. Table 2 shows that ECBRF with DPR retrieval generally outperforms ECBRF with random retrieval, which is consistent with insights from CBR.

**Step 2: Reuse**  Here we use case-augmented encoder to automatically reuse the retrieved cases.

Figure 2a shows the effects of number of retrieved cases. When the number of demonstrations is zero, prompt is also removed since no in-context demonstration is used, which makes the model exactly the same as COMET. We observe that when case-augmented encoder uses 3 cases, it reaches the best performance in both BLEU and perplexity.

Figure 2b shows the effects of \( p_{mask} \). Only when \( p_{mask} \) is 1.0, in-context demonstrations are not used at test time, which makes the model the same as COMET. As we gradually increase \( p_{mask} \), perplexity keeps improving and BLEU-2 reaches the global maximum when \( p_{mask} \) is 0.3. It is also interesting to see empirically how the case-augmented encoder gradually learns to reuse the retrieved cases to increase the performance of the deep learning model as we gradually decrease \( p_{mask} \).

In CBR, the reuse of the retrieved case’s solution contains two steps: (a) find the difference between the past and the current queries and (b) adapt the retrieved solution to the current query (Aamodt and Plaza, 1994). So it is important to know the difference between the past queries and the current input query for better adaptation. Table 7 shows the comparison between the result of only using \( obj_j \) (the retrieved cases’ object phrases) as in-context demonstrations and the result of ECBRF (uses both \( sub_r \), \( rel_r \) and \( obj_r \)), and we observe that ECBRF performs better. This result indicates that deep learning based case-augmented encoder is possible to automatically learn and reason from the difference between the past queries and the current input query for reuse.

We use prompt to indicate the role of retrieved cases and current query in input. Table 7 shows that case-augmented encoder with the prompt performs better, indicating that usage of a prompt can help the model better reuse the retrieved cases.

**Step 3 & 4: Revise and Retain**  Since revise typically involves human efforts, here we simulate revise and retain and see their effect on our framework. The result of revise and retain is a larger case base with more high quality data, and the parameters of the model for revise are not necessarily updated according to the new data. Here we simulate the effect of revise and retain by first training ECBRF in a low-resource experiment (with a small case base), then at test time we expand the case base to the full train set. Table 7 shows that, at test time, ECBRF with access to a larger case base substantially outperforms ECBRF (with access to only a small case base), although the parameters have not been updated with the new data. This result demonstrates that our framework can benefit from CBR’s methodology as revise and retain.

### 7 Conclusion

Drawing insights from CBR, we propose an end-to-end framework for the CKBC task. We demonstrate through automatic and human evaluations that our framework generates more valid, informative, and novel knowledge than the state-of-the-art COMET model in both the fully supervised and few-shot settings. From the perspective of CBR, our framework addresses a fundamental question on whether CBR methodology can be utilized to improve deep learning models.
8 Limitations

From the perspective of CBR, we have shown through experiments that our framework can perform retrieve and reuse steps, and can benefit from revise and retain steps. But the revise step in CBR typically involves human efforts, and this paper does not focus on addressing this challenge. As a result, our framework might still need manual efforts to benefit from revise and retain.

However, human efforts could be more efficiently utilized for revise than writing new data from scratch. Since comparing with requesting the workers to write the knowledge from scratch, revising the existing generations of ECBRF could be much faster.

References


Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. 2021. Understanding few-shot commonsense knowledge models. AKBC.


Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. Communications of the ACM.


### A Appendix

#### A.1 Hyperparameters

We use a common training hyperparameters setup for PLM. Specifically, learning rate is 1e-5, and batch size is 8 in both ConceptNet and ATOMIC experiments, for both COMET and ECBRF. The remainder of our training hyperparameters are the same as COMET (Bosselut et al., 2019). All BLEU and perplexity numbers are the averaged results of three different runs.

For decoding hyperparameters, we use top-k decoding and we derive the best decoding hyperparameters in the validation set for each model in each dataset.

For BART experiment, ConceptNet dataset, COMET: k: 50; length penalty: 1.0; temperature: 0.7; ECBRF: k: 50; length penalty: 1.0; temperature: 0.8.

For BART experiment, ATOMIC dataset, COMET: k: 50; length penalty: 1.2; temperature: 0.7. ECBRF: k: 50; length penalty: 1.2; temperature: 1.2.

For GPT-2 experiment, ConceptNet dataset, for both COMET and ECBRF: k: 50; length penalty: 1.0; temperature: 0.25.

For GPT-2 experiment, ATOMIC dataset, for both COMET and ECBRF: k: 50; length penalty: 1.2; temperature: 0.25.

#### A.2 Examples that BLEU not a Perfect Metric

Table 8 shows three examples with three different rel from ATOMIC that shows although ECBRF’s generations are reasonable, but they only receive low BLEU scores.

This table shows that sometimes a good generation is assigned with a low BLEU score, especially when the generation is novel and unseen from the ground truth set. Table 5 shows that ECBRF produces more novel generations compared with the COMET baseline, which might make ECBRF suffer more from the imperfectness of the BLEU score.

#### A.3 Examples on How Retrieved Cases Influence ECBRF

Table 9 shows three examples of the generation of ECBRF and COMET.

This table shows that many ECBRF’s generations are related to the retrieved case, exhibiting the case-based reasoning ability of reusing the retrieved old experience to solve new problems.

#### A.4 More Related Works on CBR

Leake and Crandall (2020) advocate using CBR to complement the challenges in deep learning (e.g., few-shot learning). However, past works on using CBR to complement DL only limit to shallow Neural Networks (NN) (Liao et al., 2018; Leake et al., 2021; Ye et al., 2021, 2022). The latest work even suggests that in many tasks NN itself outperforms CBR-complemented NN (Ye et al., 2022), which raises fundamental questions on whether CBR methodology is useful for DL. Other relevant works include only use CBR to improve the explainability of deep learning models (Keane et al., 2021), or use CBR to improve the performance of symbolic reasoning (Das et al., 2020a,b).
Example 1
sub: PersonX spends ___ working
rel: As a result, others feel
ground truth: ['happy', 'happy to have x in their life']
COMET’s generation: happy (BLEU score: 31.62)
ECBRF’s generation: satisfied with personx’s work (BLEU score: 0.00)

Example 2
sub: PersonX expects another ___
rel: This person then
ground truth: ['prepares themselves', 'gains knowledge']
COMET’s generation: gains knowledge (BLEU score: 100.00)
ECBRF’s generation: wants to find out what it is that they are going to do next (BLEU score: 0.00)

Example 3
sub: PersonX spends the ___ working
rel: This person is seen as
ground truth: ['diligent', 'tired', 'hardworking']
COMET’s generation: hardworking (BLEU score: 31.62)
ECBRF’s generation: dedicated (BLEU score: 0.00)

Table 8: Examples to show that BLEU is not a perfect metric for CKBC. This table shows that sometimes a good generation is assigned with a low BLEU score, especially when the generation is novel and unseen from the ground truth set. Table 5 shows that ECBRF produces more novel generations compared with the COMET baseline, which might make ECBRF suffer more from the imperfectness of the BLEU score.

Example 1
sub: PersonX wants to play with PersonY
rel: Before, this person needed
COMET’s generation: to have a game
Retrieved cases by ECBRF include:
(PersonX plays tennis with PersonY’s friend, Before, this person needed, get a tennis racket)
ECBRF’s generation: to find a tennis court

Example 2
sub: PersonX advances another ___
rel: This person then
COMET’s generation: PersonX gains knowledge
Retrieved cases by ECBRF include:
(PersonX marries the king’s ___, This person then, he becomes king)
ECBRF’s generation: becomes more powerful

Example 3
sub: PersonX smiles broadly
rel: As a result, this person wants to
COMET’s generation: to smile back
Retrieved cases by ECBRF include:
(PersonX grins like a cheshire cat, As a result, this person wants to, express their feelings)
ECBRF’s generation: to tell others about the good time they had

Table 9: Examples to show how the retrieved cases influence the generated obj. This table shows that many ECBRF’s generations are related to the retrieved case, exhibiting the case-based reasoning ability of reusing the retrieved old experience to solve new problems.