KFCNet: Knowledge Filtering and Contrastive Learning Network for Generative Commonsense Reasoning

Haonan Li[1*], **Yeyun Gong**[2†], **Jian Jiao**[3], **Ruofei Zhang**[3], **Timothy Baldwin**[1], **Nan Duan**[2]
[1]School of Computing and Information Systems, The University of Melbourne
[2]Microsoft Research Asia [3]Microsoft
haonanl5@student.unimelb.edu.au, tb@ldwin.net
{yegong,Jian.Jiao,nanduan,bzhang}@microsoft.com

Abstract

Pre-trained language models have led to substantial gains over a broad range of natural language processing (NLP) tasks, but have been shown to have limitations for natural language generation tasks with high-quality requirements on the output, such as commonsense generation and ad keyword generation. In this work, we present a novel **K**nowledge **F**iltering and **C**ontrastive learning **Net**work (KFCNet) which references external knowledge and achieves better generation performance. Specifically, we propose a BERT-based filter model to remove low-quality candidates, and apply contrastive learning separately to each of the encoder and decoder, within a general encoder–decoder architecture. The encoder contrastive module helps to capture global target semantics during encoding, and the decoder contrastive module enhances the utility of retrieved prototypes while learning general features. Extensive experiments on the CommonGen benchmark show that our model outperforms the previous state of the art by a large margin: +6.6 points (42.5 vs. 35.9) for BLEU-4, +3.7 points (33.3 vs. 29.6) for SPICE, and +1.3 points (18.3 vs. 17.0) for CIDEr. We further verify the effectiveness of the proposed contrastive module on ad keyword generation, and show that our model has potential commercial value.

1 Introduction

Pre-trained language models have achieved impressive results across a wide range of NLP tasks (Devlin et al., 2019; Yang et al., 2019; Sun et al., 2019; Liu et al., 2019; Lewis et al., 2020a; Qi et al., 2020; He et al., 2020b). However, their ability to accurately reflect factual knowledge or perform logical inference is still limited. To investigate the ability of systems to capture commonsense knowledge, datasets such as CommonsenseQA (Talmor et al., 2019), SWAG (Zellers et al., 2018), and WinoGrande (Sakaguchi et al., 2020) have been proposed. Separate to these discriminative tasks that require models to choose the correct option from multiple candidates, CommonGen (Lin et al., 2020) is framed as a generation task, and requires the system to generate a logical and coherent sentence describing an everyday scenario based on a concept set. Experiments show that state-of-the-art generation models are not adequate or accurate enough to generate plausible sentences or reflect commonsense assumptions in this setting.

External knowledge provides not only information about the sorts of relationships that hold between concepts, to potentially guide generation models in capturing the implicit logic between concepts, but also interpretability. Inspired by Lewis et al. (2020b) and Fan et al. (2020), we adopt a retrieval-and-generation framework, and propose a BERT-filter and two contrastive learning modules for retrieval and generation, respectively.

For retrieval, previous research (Lewis et al., 2020b) has shown that traditional sparse vector space models, such as TF-IDF and BM25, perform better than dense representation-based retrieval on heavily entity-centric tasks such as FEVER (Thorne et al., 2018). However, while using sparse vector space retrieval models can retrieve relevant prototypes that contain a set of concepts, there can be significant domain mismatches between the retrieved results and target distribution, making it difficult for generation models to bridge between prototypes and targets. We argue that a two-stage retrieval strategy alleviates this issue by combining sparse vector space search and dense representation filters. First, a sparse vector retrieval model is used to find passage candidates with high coverage of concept words, and then a dense vector-based filter is applied to score the candidates, and filter

[*]Work done during an internship at Microsoft Research Asia.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2918–2928
November 7–11, 2021.

out low-quality prototypes.

For generation, we apply contrastive learning to each of the encoder and decoder, in a general encoder–decoder architecture. The core idea of contrastive learning is to construct positive and negative samples from an anchor sample, and draw together the anchor and positive samples while pushing away the anchor from all negative samples in the embedding space during training. Given that high-quality prototypes can be used as clusters of positive samples, we propose a decoder contrastive module that minimizes the distance between decoded sentence representations with distinct prototypes retrieved from the same concept set. Common scenario information and abstract concept relationships can be learned based on the contrasts between different prototypes. Moreover, we propose an encoder contrastive module to force the encoder to learn sentence representations, and save it into a global token which is visible to the decoder during decoding. In this way, global sentence-level semantics can be captured better.

The main contributions of this work are threefold: (1) we demonstrate that adding a high-quality matching model to the word overlap-based retriever benefits entity-centric retrieval tasks; (2) we propose two contrastive learning modules that can be applied to a general encoder–decoder generation model; and (3) we conduct experiments on CommonGen and an ad keyword generation task, and show that our method achieves large-scale improvements on both tasks.

2 Related Work

2.1 Knowledge Enhanced Generation

There is significant work on incorporating external knowledge from knowledge bases and incorporating retrieved information in language generation tasks (Weston et al., 2018; Cao et al., 2018; Guan et al., 2019; Hossain et al., 2020). Lewis et al. (2020b) explore a general-purpose fine-tuning recipe for retrieval-augmented generation that combines a dense passage retriever (Karpukhin et al., 2020) with a BART (Lewis et al., 2020a) generator. For commonsense generation, Liu et al. (2020) propose a knowledge graph-augmented language generation model that encompasses concepts from a knowledge graph, and produces more logical and natural sentences. Fan et al. (2020) propose to retrieve prototypes based on sparse vector similarity, and introduce a scaling module and a prototype position indicator to explicitly deal with retrieval noise.

This paper proposes a two-stage retrieval strategy and differs in applying contrastive learning to make better use of prototypes for generation.

2.2 Contrastive Learning

Recently, contrastive learning has achieved remarkable results in many self-supervised and supervised learning tasks, primarily for computer vision. The two key elements of contrastive learning are: (1) the construction of positive and negative samples; and (2) the learning framework.

2.2.1 Sample Construction

Usually in contrastive learning, positive samples are augmented forms of anchor data points, and negative samples are augmented forms of other data points. In NLP, Meng et al. (2021) create positive samples by masking and cropping tokens from sentences; Gunel et al. (2020) and Fang and Xie (2020) use back-translation to create positive augmentations of original sentences; Chi et al. (2020) and Wei et al. (2021) regard parallel sentences distributed in one or multiple languages as different views of the same semantics to learn cross-lingual representations; and Gao et al. (2021) demonstrate that constructing positive pairs with only standard dropout as minimal data augmentation works surprisingly well on the NLI task. Distinct from these methods, we propose to create positive sample pairs from retrieved results.

2.2.2 Learning Framework

Previous contrastive learning methods have required either specialized architectures (Bachman et al., 2019; Hénaff, 2020) or a memory bank to store large volumes of negative samples (Wu et al., 2018; Tian et al., 2020). Chen et al. (2020) present a simple framework consisting of a feature extraction module, and a non-linear transformation module, which outperforms previous work on ImageNet (Russakovsky et al., 2015) without using a specialized architecture or a memory bank. However, it requires a large batch size to yield high performance, which is computationally prohibitive. Moco (He et al., 2020a) addresses this issue by maintaining a queue of data samples as the memory bank, and enqueuing encoded representations of the current mini-batch and dequeuing the oldest representations on each iteration. They further propose a momentum encoder to maintain the consistency

of representations in the queue. In this work, we use the Moco framework to train our contrastive learning modules.

3 Method

In this section, we detail our method: **K**nowledge **F**iltering and **C**ontrastive learning **Net**work (KFC-Net). First, we introduce our prototype retrieval strategy together with the knowledge filter model. Then we present our generative model, based on an encoder–decoder architecture with two contrastive learning modules. Finally, we show how we adapt the Moco contrastive learning framework, and deal with multiple positive samples.

3.1 Task Formulation

We use S to denote a set of concepts, where $S = \{c_1, c_2, ..., c_m\}$, $c_i \in \mathcal{C}$ and $\mathcal{C}$ is the concept vocabulary, and use $\mathcal{X}$ to denote all possible concept sets. The commonsense generation task is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps the concept set S to a sentence T, where $T = \{t_1, t_2, ..., t_n\} \in \mathcal{Y}$, and $\mathcal{Y}$ is the target sentence space. The generated sentence must be a plausible sentence that describes a common scenario in our daily life based on the contents of S.

3.2 Prototype Retrieval and Filtering

In order to retrieve prototypes that contain the concepts in a given concept set while keeping the retrieval results and target sentences semantically as similar as possible, we use a two-stage retrieval strategy combining sparse vector space search and dense representation matching. In Stage 1, a sparse vector retrieval model is used to retrieve N candidate prototypes from the corpus $\mathcal{D}$, where $N \ll |\mathcal{D}|$. Then in Stage 2, a dense representation-based scorer is used to score the candidates, and the top-k scored candidates are chosen as the final prototypes.

3.2.1 Stage 1

Given a concept set $S = \{c_1, c_2, ..., c_m\}$, we first split corpus $\mathcal{D}$ into $m + 1$ parts $\{d_0, d_1, ..., d_m\}$ according to the number of concepts the sentence contains, where sentences in d_i contain i concepts in x. Given that most concepts in $\mathcal{C}$ are verb and noun lemmas, we pre-process based on lemmatization and stemming. Then we choose N sentences as candidates from the parts, prioritized such that $d_m > d_{m-1} ... > d_1$.

3.2.2 Stage 2

After retrieving N candidate prototypes, a scorer is used to rank the candidates and filter out candidates that are far from the targets in embedding space. In this work, we use a BERT-based model as an encoder, and use the hidden state of the `[CLS]` token as the sample representation. The representation is then feed into a multi-layer perceptron with a scalar output as follows:

$$r_{cls} = \text{BERT}(S) \tag{1}$$

$$score = \text{MLP}(r_{cls}) \tag{2}$$

where $S = \texttt{[CLS]} + concept\ set + \texttt{[SEP]} + sentence + \texttt{[SEP]}$ is the training sample created from the concept set and candidates/targets, $r_{cls} \in \mathbb{R}^d$ is the sample representation, and $score \in [0, 1]$ is the final score of the sample. Theoretically, the label of the training set can be any real number in the range $[0, 1]$, but we find that it is sufficient to train the scorer as a simple binary classifier. We create the positive samples by combining a concept set with each corresponding target sentence, and create the negative samples by combining the concept set with a different candidate prototype or random sentence from $\mathcal{D}$. During inference, we score all candidates and choose the candidate with the highest score as the prototype.

For all experiments in this paper, we set $k = 2$ and $N = 8$. $k = 2$ means that for each input, we construct one positive sample which is widely used in contrastive leaning work. $N = 8$ is because experience shows that at least 2 high-quality prototypes can be retrieved with 8 candidates.

3.3 Contrastive Learning for Generation

3.3.1 Encoder–decoder Architecture

The encoder–decoder architecture is widely used in generation tasks. Compared to single decoder generation models such as GPT-2 (Radford et al., 2019) where words are conditioned only on left context, models using an encoder–decoder framework such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020) enable bidirectional interactions with an encoder, and auto-regressive generation with a decoder. In this work, we use BART with an auto-regressive objective for generation as shown in the middle of Figure 1, and propose separate contrastive modules for the encoder and decoder, corresponding to the right and left substructures in the figure.

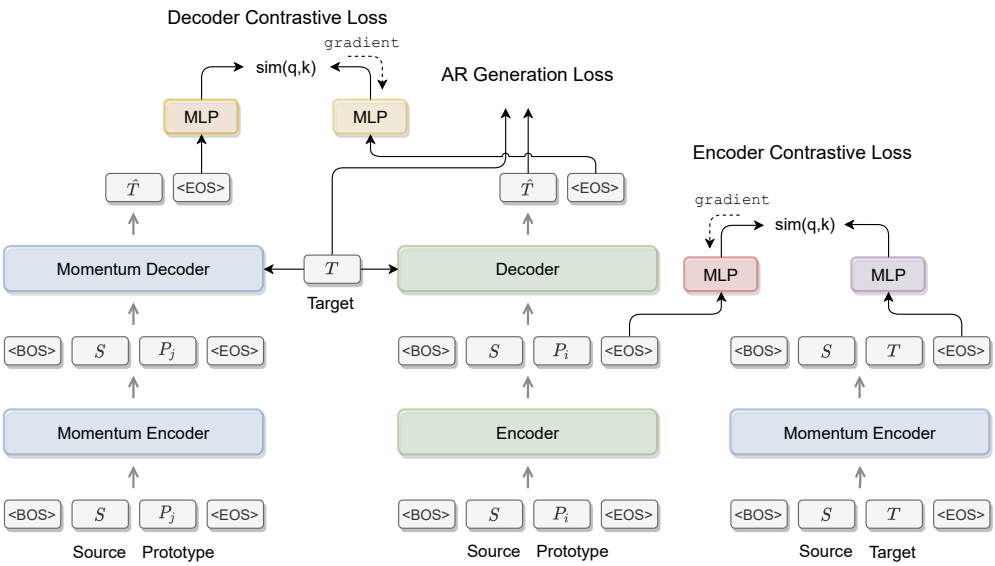

Figure 1: Generation model structure with contrastive learning.

Typical generation inputs only contain the source sentence, which is the concept set S in CommonGen. The difference here is that we append one of the retrieved prototypes $p_i \in P$ to S to enhance the input.

3.3.2 Encoder Contrastive Loss

Although BART learns bidirectional interactions using a transformer-based encoder and implements cross-attention at each layer of the decoder, the global target information is not explicitly learned during encoding, meaning that the decoder needs to find important local information for the current step during each timestep of decoding, without having access to the global goal of generation. Here, we propose to force the model to learn global target information during encoding and save it to a special token, using source–target contrastive learning. The special token is visible to the decoder via cross-attention during each timestep of decoding. Specifically, given a concept set $S = \{c_1, c_2, ..., c_m\}$ and a target sentence $T = \{t_1, t_2, ..., t_n\}$, where $c_i \in \mathcal{C}$, we denote the retrieved prototypes as $\{p_1, p_2, ..., p_k\}$, where each p_i is a complete prototype. We construct the input for encoder contrastive learning by concatenating S with T and S with p_i, respectively. As illustrated in Figure 1, the concatenation of S and p_i will be used as the input to the main encoder, which is followed by a decoder with gradient and auto-regressive generation loss, and the concatenation of S and p_i will be used as the input to another encoder without a decoder or gradient.

3.3.3 Decoder Contrastive Loss

At the retrieval step, multiple high-quality retrieval results are collected as prototypes to augment generation. Although these retrieved results substantially boost external information, they inevitably introduce noise. In order to learn general information associated with the concept set and eliminate noise in the prototypes, we propose a decoder contrastive learning module, which we apply to the sentence representation at the decoder output. Formally, we concatenate S with p_j ($j \neq i$), which is a different prototype for S from the one used in the main-branch BART model. Note also that different from the main-branch model, here the gradient is not back-propagated.

3.4 Momentum Contrast with Memory Bank

Most existing training methods greatly limit the number of in-batch negative samples, limiting the potential of contrastive learning. To enable large-scale interactions between negative samples, we follow Moco (He et al., 2020a) in maintaining a dictionary as a queue of encoded/decoded data samples. The keys of the dictionary are samples from data after encoding/decoding and the queries are samples in current mini-batches after encoding/decoding during training. Learning is formulated as minimizing the contrastive loss, which makes the query similar with its matching key and dissimilar to other keys.

3.4.1 Memory Bank as Queues

We use two dictionaries to store the representations of the encoder and decoder output, respectively. In each training iteration, the newest encoded representations are enqueued and the oldest are dequeued, to maintain a fixed queue size. For each sample, the number of contrast pairs is the size of the queue, where usually only the matching key in the same mini-batch is positive, and all others are negative.

3.4.2 Dealing with Multiple Positive Samples

In the CommonGen task, the mapping between source sequences and gold targets can be many-to-many. N independent sample pairs are created for 1-to-N and N-to-1 source–target pairs, which can be distributed across mini-batches. However, these N samples should be all regarded as mutually positive. To enable interactions between positive samples intra- and inter-mini-batch, we assign each source–target pair an identity, which indicates pairs that share the same source or target. These identities are saved in another queue that is synchronously updated with an encoder and decoder memory bank.

3.4.3 Momentum Updated Parameters

To keep the consistency of representations in the memory banks, we update the parameters of the key encoder and decoder with momentum. Formally, denote the parameters of the query encoder and decoder as θ_q^e and θ_q^d, and the parameters of the key encoder and decoder as θ_k^e and θ_k^d. The parameters θ_q^e and θ_q^d are updated by back-propagation, and the parameters θ_k^e and θ_k^d are updated by:

$$\theta_k^e \leftarrow m\theta_k^e + (1-m)\theta_q^e \tag{3}$$

$$\theta_k^d \leftarrow m\theta_k^d + (1-m)\theta_q^d \tag{4}$$

Here, $m \in [0, 1)$ is a momentum coefficient which is set to be close to 1. In this way, the parameters of the key encoder and decoder evolve more smoothly than those of the query, which maintains the consistency of key representations in the memory bank.

3.5 Training Objective

Consider a batch of query-key pairs $\{(q_1, k_1), (q_2, k_2), ..., (q_n, k_n)\}$, where there is only one positive key k_i for a given query q_i. After encoding, we fetch the representation of the last <EOS> tokens and apply a projection to it as:

$$a_i^{eos} = \text{Encoder}(q_i) \tag{5}$$

$$z_i = \text{Proj}(a_i^{eos}) \tag{6}$$

The encoder contrastive loss function, called InfoNCE, is as follows:

$$\mathcal{L}_{E_i} = -\log \frac{\exp(\text{sim}(z_{q_i}, z_{k_i})/\tau)}{\sum_{j \in M} \exp(\text{sim}(z_{q_i}, z_{k_j})/\tau)} \tag{7}$$

where $\text{sim}(,)$ denotes the similarity function, τ is a temperature hyper-parameter, and M denotes all indices in the memory bank. The denominator has $|M|$ total terms, including one positive and $|M|-1$ negative samples. Intuitively, the loss function is the log loss of an $|M|$-way softmax classifier that tries to classify q_i according to the positive k_i. Eqn (7) is only able to deal with the case of a single positive key existing for each query. To generalize it to an arbitrary number of positives, inspired by SupCon (Khosla et al., 2020), we consider the following loss functions:

$$\mathcal{L}_{E_i}^{out} = -\sum_{p \in P(i)} \frac{1}{|P(i)|} \log \mathcal{L}_{E_{i,p}}^{single} \tag{8}$$

$$\mathcal{L}_{E_i}^{in} = -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \mathcal{L}_{E_{i,p}}^{single} \right\} \tag{9}$$

$$\mathcal{L}_{E_{i,p}}^{single} = \frac{\exp(\text{sim}(z_{q_i}, z_{k_p})/\tau)}{\sum_{j \in M} \exp(\text{sim}(z_{q_i}, z_{k_j})/\tau)} \tag{10}$$

Here, $P(i)$ denotes all positive indices of the sample i, Eqn (8) summarizes the positive samples outside of the log function, and Eqn (9) summarizes those inside it.

The decoder contrastive loss $\mathcal{L}_D$ can be obtained in the same way, except that the sentence representation is fetched from the <EOS> token after decoding. During training, we try to minimize the sum of the encoder contrastive loss, decoder contrastive loss, and the decoder auto-regressive generation loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_E + \lambda_2 \mathcal{L}_D \tag{11}$$

Here, $\mathcal{L}_{CE}$ denotes the cross-entropy generation loss, and λ_1 and λ_2 are tunable scalars.

During inference, we discard the momentum encoder and decoder, together with the projection layers.

	$\mathcal{D}_{in}$	$\mathcal{D}_{out}$
# Sentences	4,118,993	70,245,048
# Length (avg)	11.10	18.76
# Missing concept (avg)		
size=3	0.40	0.48
size=4	0.56	1.00
size=5	0.80	1.98

Table 1: Statistics of the two corpora. "Missing concept" indicates the number of missing concepts in the top-2 retrieved sentences, broken down by concept-set size.

4 Experiments

4.1 Datasets

CommomGen (Lin et al., 2020) contains 32,651/993/1,497 unique training/development/test concept sets, corresponding to 67,389 and 4,018 English target sentences in the training and development sets, meaning that one concept-set can map to multiple target sentences. The percentage of concept-sets in the development and test sets that are unseen in the training set are 99.60%, and 100.00% respectively, making the dataset challenging for compositional generalization.

4.2 Prototype Collection

4.2.1 In-domain Corpus

As CommonGen was created from visually-grounded caption datasets that describe everyday scenarios, we build an in-domain corpus from datasets of image captions, video captions, and natural language inference. In detail, we extracted sentences from ActivityNet (Krishna et al., 2017), VaTeX (Wang et al., 2019), Conceptual Captions (Sharma et al., 2018), SNLI (Bowman et al., 2015), and MNLI (Williams et al., 2018) as the in-domain corpus ($\mathcal{D}_{in}$).

4.2.2 Out-of-domain Corpus

In addition to in-domain experiments, we create an out-of-domain corpus ($\mathcal{D}_{out}$) from Wikipedia,[1] using SpaCy[2] as our sentence tokenizer.

For both corpora, sentences with fewer than 5 tokens or more than 100 tokens were removed. Table 1 shows the basic statistics of the two corpora. Although $\mathcal{D}_{out}$ is much larger than $\mathcal{D}_{in}$, sentences retrieved from $\mathcal{D}_{in}$ contain more required concepts than those from $\mathcal{D}_{out}$ on average. Specifically, for concept-sets of size 4 and 5, the retrieved sentences from $\mathcal{D}_{out}$ have 0.44 and 1.18, respectively, more relevant concepts than $\mathcal{D}_{in}$.

[1] English Wikipedia dump from May 01, 2020.

[2] https://spacy.io/

4.3 Experimental Setup

4.3.1 Implementation Details

We employ the pre-trained BART-large model as the base generation model, and initialize the momentum encoder and decoder by copying parameters from the base model. We use the Adam optimizer with $\beta_{1,2} = (0.9, 0.999)$, $\epsilon = 1e-6$, and 0.1 weight decay, with the initial learning rate setting selected from $\{8e-6, 1e-5, 3e-5, 5e-5\}$. We use the polynomial decay learning rate scheduler with 500 warmup steps, and set dropout to 0.1. We set the max tokens per batch to 3000 and max batch-size to 48, with 15k total updates. For the auto-regressive generation loss, we use cross-entropy loss with 0.1 label-smoothing penalty. During decoding, we use beam search with size 5, and 1.0 length penalty.

For contrastive learning, we use an MLP as the projection network, with a single hidden layer of 1024d and the output size of 128d. We use Eqn (8) as the loss function, with similarity measured by dot-product, and set the temperature to 0.1. The queue size of the memory bank is set to 4096, and the momentum coefficient is set to 0.999.

4.3.2 Baselines

We use several state-of-the-art pre-trained language generation models as baselines: GPT-2 (Radford et al., 2019), BERT-Gen (Bao et al., 2020), UniLM (Dong et al., 2019), UniLM-v2 (Bao et al., 2020), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020a). All models are fine-tuned in seq2seq mode. We also compare our model with two strong baselines that use external knowledge: EKI (Fan et al., 2020) and KG-BART (Liu et al., 2020).

4.3.3 Evaluation Metrics

To evaluate generation performance, we use BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), in addition to evaluation metrics for captioning tasks, namely CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). As all metrics score the output in the range $[0, 100]$, we also present the average score across all metrics.

Model	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE	Overall
GPT-2 (Radford et al., 2019)	17.18	39.28	30.70	21.10	26.20	12.15	25.90	24.64
BERT-Gen (Bao et al., 2020)	18.05	40.49	30.40	21.10	27.30	12.49	27.30	25.30
UniLM (Dong et al., 2019)	21.48	43.87	38.30	27.70	29.70	14.85	30.20	29.44
UniLM-v2 (Bao et al., 2020)	18.24	40.62	31.30	22.10	28.10	13.10	28.10	25.93
T5 (Raffel et al., 2020)	22.01	42.97	39.00	28.60	30.10	14.96	31.60	29.89
BART (Lewis et al., 2020a)	22.23	41.98	36.30	26.30	30.90	13.92	30.60	28.89
EKI-out (Fan et al., 2020)	24.36	45.42	42.90	32.10	32.00	16.80	32.50	32.29
KFCNet-out	24.10	45.59	44.09	34.20	32.83	17.39	33.11	33.04
KG-BART (Liu et al., 2020)	23.38	44.54	42.10	30.90	32.40	16.83	32.70	31.83
EKI (Fan et al., 2020)	25.43	46.53	46.00	36.10	33.80	17.80	33.40	34.15
KFCNet w/o FC	25.16	46.13	50.22	41.97	36.22	18.85	35.90	36.35
KFCNet w/o C	25.91	46.81	54.75	47.33	38.19	20.21	38.20	38.77
KFCNet	**26.81**	**47.52**	**57.33**	**51.46**	**38.92**	**20.98**	**39.15**	**40.31**

Table 2: Overall performance of the different models on CommonGen (v1.0). Models in the first block are fine-tuned pre-trained language models without external knowledge; models in the second block use out-of-domain knowledge; models in the last two blocks use in-domain knowledge, where the KG-BART uses ConceptNet, and both EKI and KFCNet (our model) use the in-domain prototype corpus as a knowledge base.

5 Results

Table 2 presents the experimental results across all the metrics.[3] We observe the following: (1) Methods in the 2nd, 3rd, and 4th blocks of Table 2 that use external knowledge outperform the fine-tuned pre-trained language models in the first block. This demonstrates that external knowledge benefits commonsense reasoning and generation. (2) The overall performance of EKI and our method (KFCNet) that both use natural sentences as prototypes is better than KG-BART, which incorporates structured knowledge from knowledge bases. We hypothesize that this is because pre-trained language models like BART can more easily exploit natural language samples than structured information, even with elaborate modules for information fusion. (3) Prototypes retrieved from the in-domain corpus result in better performance than those from the out-of-domain corpus. (4) Simply fine-tuning BART on our retrieved prototypes beats previous published SOTA on several metrics, and using filtered prototypes boosts the performance again. This on the one hand shows that the quality of prototypes has a large impact on generation, and on the other hand, indicates our retrieval method is better than that of EKI, and our filter helps in selecting good prototypes. (5) Our KFCNet achieves new state-of-the-art results which surpass all other methods by a large margin.

[3]Note that the latest test set (v1.1) adds one more human reference to each example in the test set (v1.0), but is not publicly available. Additionally, EKI and KG-BART were evaluated on v1.0, so this is what we use for our experiments.

Model	BLEU-4	CIDEr	SPICE
w/o Retrieval	26.30	13.92	30.60
BM25	36.84	17.33	32.96
+Lemma & Stem	41.97	18.85	35.90
+BERT Filter	**47.33**	**20.21**	**38.20**

Table 3: Results for fine-tuning BART based on different retrieval strategies over the test set.

5.1 Ablation study

To better understand the impact of the different modules in KFCNet, we perform a number of ablation experiments.

5.1.1 Retrieval and Filter

Prototype retrieval is a key part of any retrieval-based generation model. To assess the effectiveness of the retrieval-and-filter mechanism, we retrieve prototypes from the in-domain corpus and run ablations on a single BART model. Table 3 shows the results. Compared to models without retrieval, using prototypes retrieved by a simple BM25 model improves generation performance, which we suggest is due to the retrieved prototypes helping the model to better capture relationships

Model	BLEU-4	CIDEr	SPICE
KFCNet	**36.10**	**17.96**	33.89
$-C_D$	34.09	17.24	**33.97**
$-C_E - C_D$	30.82	16.20	33.16

Table 4: Contrastive ablation study on CommonGen development set. C_E and C_D denote the encoder and decoder contrastive modules, respectively.

Sim	Sum	BLEU-4	CIDEr	SPICE
$(\cdot)$	Out	**34.11**	**16.77**	33.59
$(\cdot)$	In	32.45	16.54	**33.79**
$\cos(,)$	Out	32.49	16.62	33.73
$\cos(,)$	In	33.52	16.64	33.39

Table 5: Comparison of different similarity functions and positive sample summation locations. $(\cdot)$ denotes dot-product similarity, and $\cos(,)$ denotes cosine similarity.

between concepts, and construct a coherent scenario. With word lemmatization and stemming, the variety of the retrieval results increases, resulting in better prototypes. Adding a BERT filter boosts the performance again, achieving +5.38, +1.36, and +2.30 absolute improvements for BLEU-4, CIDEr, and SPICE. This verifies the effectiveness of using a high-quality matching model as an auxiliary module for a word overlap-based retriever.

5.1.2 Contrastive Learning

The contrastive loss plays an important role in our model. We perform an ablation study on the development set of CommonGen, by comparing the model without the contrastive module, using only encoder contrastive learning, and using both encoder and decoder contrastive learning. As shown in Table 4, using only encoder contrastive learning leads to improvements over the baseline BART model, and adding decoder contrastive learning further improves results based on BLEU-4 and CIDEr.

5.2 Similarity Function and Summation Location

We further compare the performance of different similarity functions and positive summation locations, as mentioned in Section 3.5. The results in Table 5 demonstrate that the combination of dot-product similarity with summation outside of the log function performs best, consistent with the findings of Khosla et al. (2020).

Model	BLEU-4	CIDEr	SPICE
Human	46.49	37.64	52.43
KFCNet	**42.45**	**18.37**	**33.27**
RE-T5	40.86	17.66	31.07
KG-BART	33.86	16.92	29.63
EKI-BART	35.94	16.99	29.58
T5-Large	31.96	15.12	28.85
BART	31.82	13.97	27.99
UniLM	30.61	14.88	27.42
BERT-Gen	23.46	12.60	24.82
GPT-2	26.83	12.18	23.56

Table 6: Final CommonGen leaderboard results, using SPICE to rank the methods.

5.3 Model Efficiency

5.3.1 Retrieval

The prototype retrieval is done separately from the generation model, and the retrieval time consists of 2 parts: (1) sparse vector matching time, in the form of BM25 search; and (2) BERT filter inference, for fine-grained selection, noting that only a few candidates (8 in our experiments) are left after stage 1, which can be processed in a single mini-batch.

5.3.2 Contrastive Module

During training, the momentum encoder and decoder parameters are updated by Eqn (3) and there are no gradients or back-propagation in these modules. Therefore it takes no more than double the training time without contrastive modules. During inference, the contrastive modules are disabled, and hence the efficiency does not decrease.

5.4 Final Leaderboard Results

Table 6 shows the final evaluation results on the latest test set with additional human references (v1.1).[4] Note that the model in second place (RE-T5) expands the original training data and does continuous pretraining before fine-tuning on CommonGen. Our method, KFCNet, performs best on all metrics. Among all fine-tuned methods, KFCNet beats the previous state-of-the-art by a large margin: +6.51 (18.11%) for BLEU-4, +1.38 (8.12%) for CIDEr, and +3.64 (11.95%) for SPICE.

[4] https://inklab.usc.edu/CommonGen/leaderboard.html

Model	ROUGE-2/L	BLEU-3/4	AVG
BART	33.03/60.17	31.61/25.03	38.96
$+R$	33.68/60.31	31.69/24.75	39.70
$+C_{E,D}$	**35.18/61.24**	**33.60/26.78**	**41.44**

Table 7: Experimental results on ad keyword generation.

5.5 Experiment on Keyword Generation

To test the effectiveness of the proposed contrastive learning modules, we constructed a real-world adword dataset, based on an advertising platform (Edelman et al., 2007). The goal is to display a list of ads that matches the user intent, for which the first step is to retrieve relevant keywords provided by advertisers given a user query. The dataset contains 72,876 training samples, 10,000 dev samples, and 10,000 test samples from a major search engine, with each sample corresponding to a query–keyword pair. Titles of the top-two web search results of the query from the search engine are kept as prototypes.

We fine-tune BART models following the same sequence generation experimental design. The results are shown in Table 7.

From the first two lines, we see that directly appending the retrieved information to the source does not lead to noticeable improvements, almost certainly because of noise in the retrieved results. However, our contrastive modules alleviate the effects of noise, and beat BART on all metrics.

6 Conclusion

In this paper, we present KFCNet: a novel knowledge filtering and contrastive learning model for retrieval-augmented sequence generation, which achieves state-of-the-art results on the CommonGen benchmark. Two contrastive learning modules are proposed to capture global target semantics and learn general features from multiple retrieved prototypes. A prototype retrieval ablation study showed the effectiveness of the proposed filter for filtering low-quality candidates, and further experiments on ad keyword generation showed that our model has potential commercial value. In the future, we plan to extend the contrastive module to more general settings, such as natural language understanding and representation learning.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: semantic propositional image caption evaluation. In *ECCV - 14th European Conference, Part V*, volume 9909, pages 382–398.

Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 15509–15519.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL*, pages 65–72.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 642–652.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 632–642.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 152–161.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119, pages 1597–1607.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. *CoRR*, abs/2007.07834.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou,

and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 13042–13054.

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259.

Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuanjing Huang, Nan Duan, and Ruofei Zhang. 2020. An enhanced knowledge injection model for commonsense generation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING*, pages 2014–2025.

Hongchao Fang and Pengtao Xie. 2020. CERT: contrastive self-supervised learning for language understanding. *CoRR*, abs/2005.12766.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6473–6480.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *CoRR*, abs/2011.01403.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020a. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 9726–9735.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020b. DeBERTa: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Olivier J. Hénaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119, pages 4182–4192.

Nabil Hossain, Marjan Ghazvininejad, and Luke Zettlemoyer. 2020. Simple and effective retrieve-edit-rerank text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 2532–2538.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 6769–6781.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems, NeurIPS*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *International Conference on Computer Vision, ICCV*, pages 706–715.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 7871–7880.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems, NeurIPS*.

Bill Yuchen Lin, Ming Shen, Wangchunshu Zhou, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Conference on Automated Knowledge Base Construction, AKBC*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2020. KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. *CoRR*, abs/2009.12677.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: correcting and contrasting text sequences for language model pretraining. *CoRR*, abs/2102.08473.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 311–318.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP*, pages 2401–2410.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial Winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8732–8740.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 2556–2565.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. 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, NAACL-HLT*, pages 4149–4158.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 809–819.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *ECCV - 16th European Conference, Proceedings, Part XI*, volume 12356, pages 776–794.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4566–4575.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VaTeX: A large-scale, high-quality multilingual dataset for video-and-language research. In *International Conference on Computer Vision, ICCV*, pages 4580–4590.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jason Weston, Emily Dinan, and Alexander H. Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP*, pages 87–92.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT)*, pages 1112–1122.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3733–3742.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 5754–5764.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 93–104.