Human Parity on CommonsenseQA: Augmenting Self-Attention with External Attention

Yichong Xu, Chenguang Zhu, Shuohang Wang, Siqi Sun, Hao Cheng,
Xiaodong Liu, Jianfeng Gao, Pengcheng He, Michael Zeng and **Xuedong Huang**
Microsoft Corporation
{yicxu,chezhu,shuowa,siqi.sun,chehao,xiaodl,jfgao,penhe,nzeng,xdh}@microsoft.com

Abstract

Most of today's AI systems focus on using self-attention mechanisms and transformer architectures on large amounts of diverse data to achieve impressive performance gains. In this paper, we propose to augment the transformer architecture with an external attention mechanism to bring external knowledge and context to bear. By integrating external information into the prediction process, we hope to reduce the need for ever-larger models and increase the democratization of AI systems. We find that the proposed external attention mechanism can significantly improve the performance of existing AI systems, allowing practitioners to easily customize foundation AI models to many diverse downstream applications. In particular, we focus on the task of Commonsense Reasoning, demonstrating that the proposed external attention mechanism can augment existing transformer models and significantly improve the model's reasoning capabilities. The proposed system, Knowledgeable External Attention for commonsense Reasoning (KEAR), reaches human parity on the open CommonsenseQA research benchmark with an accuracy of 89.4% in comparison to the human accuracy of 88.9%.

1 Introduction

Transformers [Vaswani *et al.*, 2017] have revolutionized many areas of AI with state-of-the-art performance in a wide range of tasks [Devlin *et al.*, 2018; Dosovitskiy *et al.*, 2020]. The most notable and effective component in a Transformer model is the self-attention mechanism, which enables the model to dynamically leverage different parts of the input for computation, with no information loss for even the most distant parts of the input. With the success of pre-trained models [Devlin *et al.*, 2018; Liu *et al.*, 2019], the Transformer and its self-attention mechanism have been widely adopted as the cornerstone of foundation models trained on huge amounts of data [Bommasani *et al.*, 2021].

One phenomenon found during the development of Transformer models is that models with larger sizes tend to have better learning abilities, especially when combined with large-scale data. This has prompted the recent boom of super large Transformer models, ranging from BERT [Devlin *et al.*, 2018] with 110 million parameters, to GPT-3 [Brown *et al.*, 2020] with 175 billion parameters. Nevertheless, numerous studies have shown that the corresponding understanding and generation capabilities of these huge models are still behind humans [Bommasani *et al.*, 2021]. Furthermore, the sheer size of these models already poses serious practical challenges in utilization, deployment, interpretation, and environmental impact [Patterson *et al.*, 2021]. Thus, the recent "scaling-up" approach to Transformer-based NLP modeling is unsustainable and has been questioned in recent studies [Bommasani *et al.*, 2021].

In this paper, we take a step back and examine the mechanism of current Transformer-based models. Self-attention was designed to allow the model to better analyze the inner structure of input data, and the model is trained to have its parameters grasp and memorize all the content and patterns of the training data. When the model is given a novel input X, the implicitly stored knowledge in the parameters about related information is activated to facilitate the analysis of X. This could partly explain why larger models pre-trained with more data have an advantage in performance.

While Transformer models process input by looking *inward* via self-attention, we propose to make the model look *outward* by providing it with related context and knowledge from various sources. We then let the model conduct self-attention on the input while also computing external attention to the knowledge (Figure 1). As the context and knowledge can usually be stored in a non-parametric and symbolic way (e.g., plain text, knowledge graph and dictionary entries), even moderately-sized Transformer models can perform exceptionally well on NLP tasks. This approach allows one to shrink the size of Transformer-based foundation models, which is critical to the accessibility and democratization of AI technology. This approach is also analogous to the way humans conduct intelligence; we often resort to search engines, dictionaries, or information from other people to navigate the world.

Another benefit of external attention is that, as the related knowledge is stored outside of the model, practitioners can easily update the knowledge source to change the behavior of their models. For example, one could add or delete entries from a knowledge graph or rewrite certain paragraphs in Wikipedia. By explicitly representing knowledge, the decision process of the model becomes much more transparent and explainable.

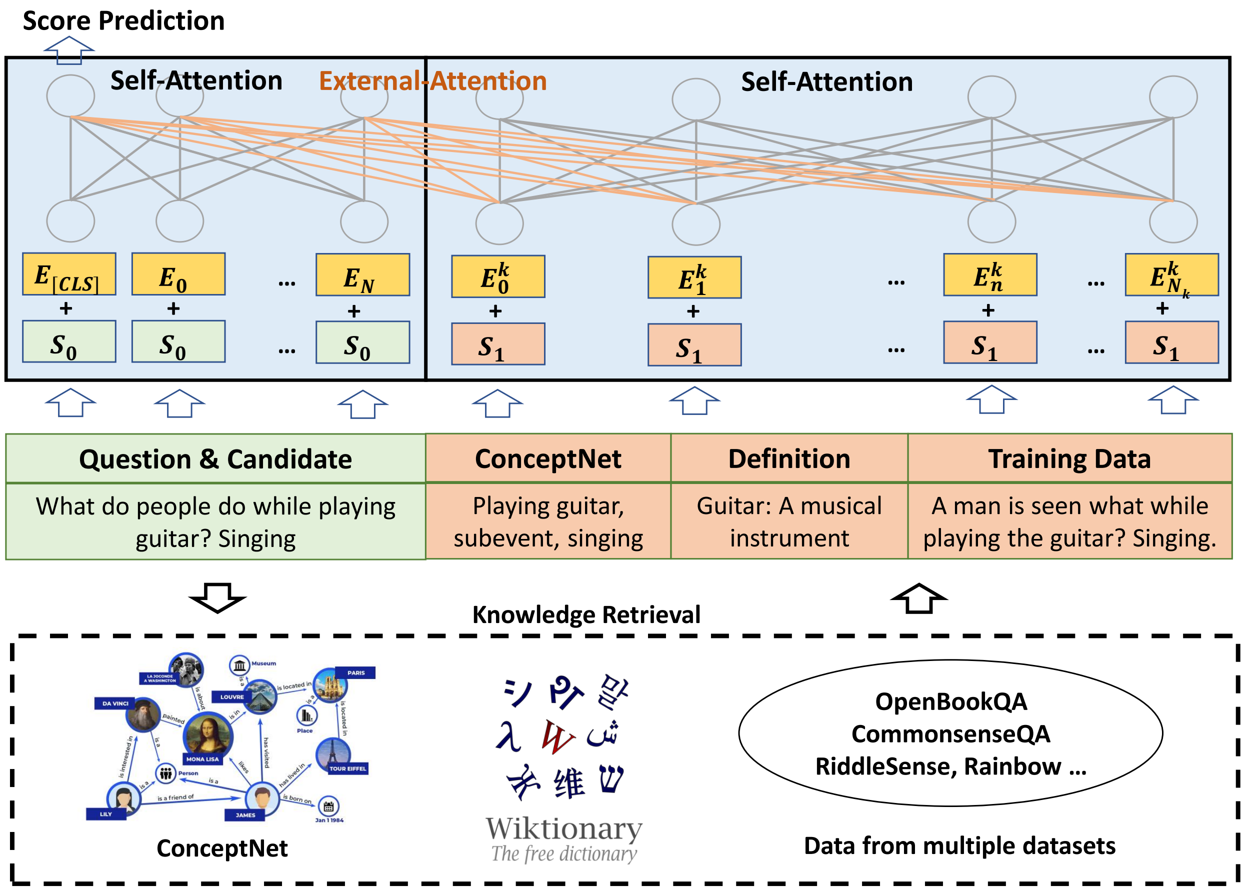

Figure 1: Our proposed method of Knowledgeable External Attention for commonsense Reasoning (KEAR). Related knowledge is retrieved from external sources, e.g., knowledge graph, dictionary and training data, using the input as the key and then integrated with the input. While additional external attention layers can be added to the Transformer blocks, we adopt text-level concatenation for external attention, incurring no structural change to the model architecture.

In this paper, we use the commonsense reasoning task CommonsenseQA [Talmor *et al.*, 2019] as a case study in leveraging external attention to obtain and integrate information related to the input. Given a commonsense question and a choice, we retrieve knowledge from three external sources: a knowledge graph (ConceptNet), a dictionary (Wiktionary), and labeled training data (CommonsenseQA and 16 related QA datasets). The retrieved knowledge is directly appended to the input and sent to the language model with no revision to the underlying architecture. We show that with the proposed external attention, the accuracy of commonsense reasoning using a DeBERTa-xxlarge model [He *et al.*, 2020] can be significantly boosted from 83.8% to 90.8% on the dev set, while fine-tuned large-scale models like GPT-3 can only achieve 73.0%. The ensembled version of our model, Knowledgeable External Attention for commonsense Reasoning (KEAR), reaches an accuracy of 93.4% on the dev set and 89.4% on the test set, surpassing human performance (88.9%) for the first time [Talmor *et al.*, 2019].

The benefits of our approach extend beyond commonsense reasoning. First, the external attention dramatically reduces our system's dependence on large-scale models, i.e., achieving human parity with models up to 1.5B parameters. Second, the external information is obtained via computationally efficient methods, such as information retrieval and word matching, adding little computational cost to the main model. Third, the text-level concatenation of input and knowledge leads no change to the Transformer model, enabling existing systems to easily adopt this new external attention mechanism.

2 Method

We first describe our external attention framework in Sec 2.1. Next, we describe our external knowledge sources in Sec 2.2. Last, we present additional modeling techniques for improving commonsense reasoning in Sec 2.3.

Problem Formulation. We focus on the multiple-choice question answering task in this paper, where the goal is to select the correct answer from a given list $c_1, c_2, ..., c_n$ for a commonsense question q. The output of the model is a distribution $\mathcal{P}$ on $\{1, 2, ..., n\}$.

2.1 External Attention

Self Attention. The majority of recent language models are based on the Transformer architecture [Vaswani *et al.*, 2017].

One of the most important components in Transformer is the self-attention mechanism, which can be formulated as

$$\boldsymbol{Q} = \boldsymbol{H}_l \boldsymbol{W}_q, \boldsymbol{K} = \boldsymbol{H}_l \boldsymbol{W}_k, \boldsymbol{V} = \boldsymbol{H}_l \boldsymbol{W}_v,$$
$$\boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}, \boldsymbol{H}_{l+1} = \text{softmax}(\boldsymbol{A})V, \quad (1)$$

where $\boldsymbol{H}_l \in \mathbb{R}^{N \times d}$ is the input hidden vectors to the l-th Transformer layer, $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{d \times d}$ are projection matrices, N is the input length and d is the hidden vector's dimension. The inputs to the first Transformer layer are usually the embeddings of the tokenized input text, denoted as $\boldsymbol{H}_0 = \boldsymbol{X} = [x_1, x_2, ..., x_N]$[1]. In the multi-choice question answering context, the input text is a concatenation of the question and a specific choice.

External Attention. For commonsense question answering, the required information needed to answer the question is usually absent from the input. Thus, we need to integrate external knowledge into the model. In this work, we denote the extra knowledge in text format as $\boldsymbol{K} = [x_1^K, x_2^K, ..., x_{N_k}^K]$. There are many ways to integrate the external knowledge into the model, such as using graph neural networks [Lin *et al.*, 2019]. In this paper we simply concatenate the knowledge to the input text: $\boldsymbol{H}_0 = [\boldsymbol{X}; \boldsymbol{K}] = [x_1, ..., x_N, x_1^K, ..., x_{N_k}^K]$. The advantage of this input-level integration is that the existing model architecture does not need to be modified. Then, applying self-attention on $\boldsymbol{H}_0$ can make the model freely reason between the knowledge text and the question/choices, therefore equipping the model with enhanced reasoning capacity.

2.2 External Knowledge Sources

The knowledge to append to the input for external attention is crucial for getting the correct prediction. For commonsense reasoning, we collect three external knowledge sources to complement the input questions and choices.

Knowledge Graph. Knowledge graphs (KG) contain curated facts that can help with commonsense reasoning which might not appear in a general corpus. We follow KCR[2] to retrieve relevant relation triples in the ConceptNet graph [Speer *et al.*, 2017]. Suppose the question entity is e_q and the choice contains entity e_c[3]. If there is a direct edge r from e_q to e_c in ConceptNet, we choose this triple (e_q, r, e_c). Otherwise, we retrieve all the triples originating from e_c. We score each triple j by the product of its confidence w_j (provided by ConceptNet) and the defined relation type weight t_{r_j}: $s_j = w_j \cdot t_{r_j} = w_j \cdot \frac{N}{N_{r_j}}$, where r_j is the relation type of j, N is the total number of triples originating from e_c, N_{r_j} is the number of triples with relation r_j among these triples. We then choose the triple with highest weight. Finally, if the selected triple is (e_1, r, e_2), we denote the knowledge from the KG as $\boldsymbol{K}_{\text{KG}}(q, c) = [e_1 \ r \ e_2]$.

Dictionary. Although pre-trained language models are exposed to large-scale text data, the long tail distribution of words means that the quality of a word's representation is highly dependent on that word's frequency in the pre-training corpus. Dictionaries, on the other hand, can provide accurate semantic explanation of words regardless of their frequency in datasets. To help understand key concepts in the question and answer, we follow DEKCOR [Xu *et al.*, 2021] to use the Wiktionary definitions of the question and answer concepts as external knowledge. For every concept, we fetch the first (most frequent) definition from Wiktionary using its closest lexical match. Let d_q be the definition text for e_q and d_c be the definition text for e_c, we denote the dictionary knowledge as $\boldsymbol{K}_{\text{dict}}(q, c) = [e_q : d_q; e_c : d_c]$.

Training Data. Although recent language models are giant in terms of the number of parameters, recent studies show that they cannot perfectly memorize all the details of their training data [Wang *et al.*, 2022].

To tackle this challenge, we propose to retrieve relevant questions and answers from the training data as additional knowledge. We use BM25 [Schütze *et al.*, 2008] to retrieve top M relevant questions and answers from the training data. For each question q, we index the concatenation of the question text, the ground-truth choice c^*, ConceptNet triples and Wiktionary definitions: $[q; c^*; \boldsymbol{K}_{\text{KG}}(q, c^*); \boldsymbol{K}_{\text{dict}}(q, c^*)]$. For a new question q' and a potential choice c', we similarly build a query $[q'; c'; \boldsymbol{K}_{\text{KG}}(q', c'); \boldsymbol{K}_{\text{dict}}(q', c')]$ to retrieve most similar questions from the training set. For each retrieved question from the training data, we drop the knowledge part and employ the retrieved question and the ground-truth answer as external knowledge. Suppose the retrieved questions and (correct) answers are $\{(q_1, c_1^*), (q_2, c_2^*), ..., (q_M, c_M^*)\}$, we denote the knowledge from training data as $\boldsymbol{K}_{\text{train}} = [q_1 \ c_1^*; q_2 \ c_2^*; \cdots; q_M \ c_M^*]$. During training, for each question q we filter itself from the retrieved results to avoid data leakage.

Different from Wang *et al.* [2022] where the retrieval questions are only obtained from the same dataset, we experiment with three sources of training data for retrieval: i) CSQA training data, ii) CSQA+OBQA+RiddleSense, a small collection of datasets focusing on ConceptNet knowledge, and iii) a pool of 17 datasets focusing on commonsense reasoning (we describe details of these 17 datasets in the Appendix). Since most datasets do not provide the question and choice entity e_q, e_c for every question-choice pair, we use entity linking to find all entities $E_q = \{e_q^{(1)}, ..., e_q^{(n_q)}\}, E_c = \{e_c^{(1)}, ..., e_c^{(n_c)}\}$ appearing in the question and choice text respectively. We select the entity with the maximum length in E_q and E_c as the question and choice entity for Wiktionary definitions. For ConceptNet triples, we find edges between E_q and E_c and choose the one with the maximum total length.

Finally, we concatenate the retrieved knowledge from our three sources to form a final knowledge input: $\boldsymbol{K} = [\boldsymbol{K}_{\text{KG}}; \boldsymbol{K}_{\text{dict}}; \boldsymbol{K}_{\text{train}}]$. In practice, the semicolon is replaced by the separator token (e.g., `[SEP]`). We name our knowledge retrieval and integration technology as Knowledgeable External Attention for commonsense Reasoning (KEAR), shown in Figure 1.

[1] We do not differentiate between tokens and their embeddings in the following discussion. Following previous work, we prepend a `[CLS]` token to the input.

[2] https://github.com/jessionlin/csqa

[3] In CommonsenseQA dataset, both e_q and e_c are provided. Otherwise, we use entity linking to find related knowledge graph nodes to the input text (see "Training Data" part later in this section).

2.3 General Methods to Improve Commonsense Reasoning

Prior works have proposed other methods to improve general NLU performance, and it is therefore natural to wonder if these methods also work for commonsense reasoning. Here, we explore two general methods for improving commonsense reasoning performance: i) using different text encoders and ii) virtual adversarial learning.

Text Encoders. Previous methods for natural language understanding (NLU) have tried using BERT [Devlin *et al.*, 2018], RoBERTa [Liu *et al.*, 2019], ALBERT [Lan *et al.*, 2019], T5 [Raffel *et al.*, 2019], ELECTRA [Clark *et al.*, 2020] and DeBERTa [He *et al.*, 2020] as the text encoder, achieving state-of-the-art performance on the GLUE benchmark [Wang *et al.*, 2019]. Thus, we evaluate these models as encoders for the commonsense reasoning task.

Virtual Adversarial Training (VAT). Previous works show that virtual adversarial training (VAT) can improve the performance for general NLU and question answering tasks [Jiang *et al.*, 2020]. In the multiple-choice commonsense reasoning task, the goal is to minimize the cross-entropy loss:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\mathrm{CE}(f(x;\theta), y)] \tag{2}$$

where f produces the model prediction (distribution $\mathcal{P}$ on the choices), θ represents the model parameters, y is the one-hot ground-truth answer vector, CE is cross-entropy, and D is the empirical data distribution. VAT first finds the update δ that leads to the largest change in the predicted distribution, subject to a L_2-norm constraint. Then, a consistency regularization loss term is added to minimize the difference in the function's output when compared to the input variation δ:

$$\begin{aligned}\min_{\theta}\ & \mathbb{E}_{(x,y)\sim D}[\mathrm{CE}(f(x;\theta), y) + \\ & \alpha \max_{\|\delta\|_2 \le \varepsilon} \mathrm{CE}(f(x;\theta), f(x+\delta;\theta))],\end{aligned} \tag{3}$$

where α and ε are hyperparameters.

3 Experiments

We present our empirical results in this section. Each of our three external knowledge sources can boost the commonsense reasoning performance, and combining all the three techniques helps us reach the human parity on the CommonsenseQA benchmark.

3.1 Setup

Data. We focus on the CommonsenseQA (CSQA, Talmor *et al.*, 2019) benchmark. CommonsenseQA is a widely used multiple-choice question answering dataset that requires commonsense knowledge. It contains 12k questions created using ConceptNet [Speer *et al.*, 2017]. For an edge (subject, relation, object) in ConceptNet, Talmor *et al.* [2019] retrieve other object concepts with the same subject and relation as distractors for a question. A human worker is then asked to i) write a question containing the subject and with the object as the correct answer, ii) pick the most distractive answer from the retrieved concepts, and iii) write another distractor for the question. The final question contains five choices, with one correct choice, two random retrieved concepts, one human-picked concept, and one human-curated answer.

We present details of the 17 datasets that we use for training data retrieval in Table 1. All the datasets are multiple-choice or classification datasets related to commonsense reasoning, and we include dataset details in the appendix.

Dataset	Task	#Train	#Label
αNLI	NLI	170k	2
SWAG	MC	73.5k	4
RACE-Middle	MRC	87.9k	4
CODAH	MC	1672	4
RiddleSense	MC	3512	5
SciTail	NLI	23.6k	2
Com2Sense	MC	808	2
AI2Science	MC	1232	4
WinoGrade	CoRef	40.4k	2
CSQA	MC	9741	5
CSQA2.0	CLF	9264	2
ASQ	MC	8872	2
OBQA	MC	4960	4
PhysicalIQA	MC	16.1k	2
SocialIQA	MC	33.4k	3
CosmosQA	MRC	25.3k	4
HellaSWAG	NSP	39.9k	4

Table 1: The datasets used for training data retrieval. NLI stands for natural language inference, MC is multiple choice, MRC is machine reading comprehension, CLF is classification, NSP is next sentence prediction.

Model Setup. We feed the input text into a pretrained text encoder (e.g., DeBERTa) and take the representation $\boldsymbol{v} \in \mathbb{R}^d$ of the `[CLS]` token, where d is the dimension of the encoder. We set the segment id as 0 for the question and answer text, and 1 for the appended knowledge text. The position embedding simply runs from 1 to l, where l is the total input length including the external knowledge. The embedding of `[CLS]` is projected to a scalar with learned weights and the final prediction is computed via a softmax over all five choices for a question. We then minimize the cross entropy error during training.

Implementation Details. We finetune the model using the AdamW optimizer [Loshchilov and Hutter, 2017]. The batch size is set to 48 or smaller to fit the batch onto a single GPU. We train the model for 10 epochs and take the best result on the dev set. We choose the weight decay in $\{0, 0.01, 0.1\}$. The learning rate are chosen from $\{1e-5, 2e-5, 3e-6\}$ for all encoders except for DeBERTa; following the DeBERTa paper [He *et al.*, 2020] we use a smaller learning rate, chosen from $\{4e-6, 6e-6, 9e-6\}$. We use the DeBERTa v2 model and choose from the pretrained model or model finetuned on MNLI. We also try out the recent DeBERTa V3 model [He *et al.*, 2021] which combines DeBERTa with adversarial pretraining. For VAT, we choose the weight multiplier $\alpha \in \{0.1, 1.0, 10.0\}$ and set input variation norm $\varepsilon = 1e-5$ (see Eqn. 3). For retrieving from training data, we choose the data

Encoder	**CSQA**	**MNLI**	**#Para**
Fine-tuned GPT-3	73.0	82.1	175B
RoBERTa-large	76.7	90.2	355M
ALBERT-xxlarge	81.2	90.6	235M
ELECTRA-base	75.0	88.8	110M
ELECTRA-large	81.3	90.9	335M
DeBERTa-xlarge	82.9	91.7	900M
DeBERTa-xxlarge	83.8	91.7	1.5B
DeBERTaV3-large	84.6	91.8	418M
T5-11B	83.5[1]	91.3	11B

Table 2: CSQA dev set accuracy for various encoders. We append the accuracy on MNLI dataset (in-domain) for each encoder as a reference. MNLI scores are from the corresponding GitHub repositories. [1]: from Liu *et al.* [2021].

source with the best validation set performance from the three retrieval source datasets. We set number of retrieved questions $M = 10$. We run each experiment with 3 different seeds and present results from the best run.

3.2 Effects of Individual Components

General Methods. As shown in Table 2, there is a positive correlation between general performance on NLI tasks and commonsense reasoning abilities on CommonsenseQA. Notice that the fine-tuned GPT-3 model with 175 billion parameters could only achieve 73.0% on the dev set of CommonsenseQA. Based on these results, we choose ELECTRA-large and DeBERTa variants [He *et al.*, 2020, 2021] as the encoders for subsequent experimentation.

For virtual adversarial training, we find that VAT can improve commonsense reasoning accuracy for ELECTRA, improving the result from 81.3% to 82.1%. It does not show much improvement for DeBERTa models in our experiment. Therefore, we apply VAT to ELECTRA in subsequent experiments.

Method	**E-l+VAT**	**D-xxl**	**DV3-l**
Base	82.1	83.8	84.6
+ KG	85.2	86.4	86.7
+ Dictionary	83.8	84.0	85.1
+ Training data	84.0	86.4	87.1

Table 3: Applying external attention to different knowledge sources. E-l+VAT stands for ELECTRA-large with VAT, D-xxl stands for DeBERTa-xxlarge, DV3-l stands for DeBERTaV3-large.

Effect of External Attention. As shown in Table 3, all of the proposed knowledge sources bring gains in commonsense reasoning accuracy across all base encoder models. The dictionary, knowledge graph and training data bring 0.5%, 2.1%, and 2.5% improvement, respectively, when DeBERTaV3-large [He *et al.*, 2021] is the base encoder model.

We find that the best training data retrieval source depends on the exact encoders and the techniques applied, and we show a comparison in Table 4. In general, the 17-dataset pool achieves the best performance for DeBERTa, but for ELECTRA retrieving from the CSQA training set alone can get the best performance. Table 3 and 4 demonstrate the effectiveness of our proposed knowledge retrieval and concatenation methods.

Model	**CSQA**	**3-Data**	**17-Data**
E-l+VAT	**84.0**	82.9	82.8
D-xxl	86.2	86.1	**86.4**
DV3-l	87.0	**87.1**	**87.1**
E-l+VAT + KG + Dict	**88.5**	88.2	87.1
D-xxl + KG + Dict	89.8	90.5	**90.8**
DV3-l + KG + Dict	91.0	**91.2**	**91.2**

Table 4: Performance on CSQA dev set of model w.r.t source of training data retrieval. E-l+VAT stands for ELECTRA-large with VAT, D-xxl stands for DeBERTa-xxlarge, DV3-l stands for DeBERTaV3-large.

Method	**Dev Acc(%)**
ELECTRA-large + VAT + KEAR	88.7
DeBERTa-xxlarge + KEAR	90.8
DeBERTaV3-large + KEAR	**91.2**
Ensemble (39 models w/ KEAR)	**93.4**

Table 5: CSQA dev set results with different encoders and ensembles.

Method	Single	Ensemble
BERT+OMCS	62.5	-
RoBERTa	72.1	72.5
RoBERTa+KEDGN	-	74.4
ALBERT	-	76.5
RoBERTa+MHGRN	75.4	76.5
ALBERT + HGN	77.3	80.0
T5	78.1	-
UnifiedQA	79.1	-
ALBERT+KCR	79.5	-
ALBERT + KD	80.3	80.9
ALBERT + SFR	-	81.8
DEKCOR	80.7	83.3
Human	-	88.9
KEAR (ours)	**86.1**	**89.4**

Table 6: Results on test set from the leaderboard. The human performance is ensemble of 5 workers [Talmor *et al.*, 2019].

3.3 Combining the Techniques

Table 5 shows the results of KEAR, which combines the best techniques in previous experiments, i.e., best encoders and external attention to all knowledge sources, to further boost the performance. The best single model (DeBERTaV3-large + KEAR) achieves 91.2% accuracy on the dev set. To get the best performance, we train KEAR models with ELECTRA large, DeBERTa xlarge (900M), xxlarge (1.5B) and V3 large as encoders with 12 different seeds, resulting in 48 models in total. We rank the models by their dev set performance as $M_1, M_2, ..., M_{48}$. The ensemble prediction uses a majority vote on individual predictions. We picked the first

Question	What has a surface with many sides?
Choices	**A) tetrahedron**, B) object, C) geometry problem, D) lake, *E) triangle*
KG	surface AtLocation {tetrahedron, object, geometry problem, lake}
Dictionary	surface: The overside or up-side of a flat object such as a table, or of a liquid. tetrahedron: A polyhedron with four faces.
Training Data	The four equal sides were all a smooth surface, he had carved and sanded a perfect what? tetrahedron.
Question	What is a treat that your dog will enjoy?
Choices	A) salad, B) petted, C) affection, **D) bone**, *E) lots of attention*
KG	dog Desires {petted, affection, bone, lots of attention}
Dictionary	dog: A mammal that has been domesticated for thousands of years. bone: A composite material making up the skeleton of most vertebrates.
Training Data	What do dogs like to eat? bones.

Table 7: Case study for the effect of external attention. We list two questions from the CSQA dataset with our retrieved knowledge (both retrieved training data questions are from CSQA). The correct answer is in **bold**, and our KEAR model selected the correct choice for both questions. We highlight the wrong choice that a DeBERTa 1.5B model chooses in *italic*.

N models such that the dev set performance of ensembling $M_1, M_2, ..., M_N$ is the best. We ended up with 39 models with 12 ELECTRA models, 12 DeBERTaV3 models, 11 DeBERTa-xxlarge models and 4 DeBERTa-xlarge models. Our ensemble model reaches 93.4% accuracy on the dev set. Table 6 shows the official leaderboard result on the hidden test set. Our ensemble model exceeds the previously best DEKCOR model by over 6% and exceeds the human performance (88.9%) by 0.5%.

3.4 Case Study

We present two examples from CSQA in Table 7 to illustrate how the model can reason between all retrieved knowledge sources to get the correct answer. For the first question, the knowledge graph helps rule out the wrong answer *triangle* since it does not have a surface. The dictionary and training data attention further confirms that a tetrahedron has four sides/faces, which is the correct answer. For the second question, again knowledge graph rules out "salad" since a dog does not desire salads. The dictionary attention results suggest that bones are important for a dog, and training data attention suggests that bones are good food for a dog. This leads to the correct answer (bone). This suggests that all three knowledge sources are critical for getting the correct answer. Having access to all three knowledge sources makes it easier for model reasoning to get the correct answer.

4 Related work

Many previous works have proposed ways of incorporating external knowledge sources into Transformer architectures. For commonsense question answering, specialized knowledge graphs like ConceptNet [Speer *et al.*, 2017] and ATOMIC [Sap *et al.*, 2019] are the most popular choices for external knowledge [Chang *et al.*, 2021; Yao *et al.*, 2022; Song *et al.*, 2021]. Lin *et al.* [2019] construct a scheme graph from concepts in the question and choices and uses an LSTM to reason on paths between question and choice concepts. Yasunaga *et al.* [2021] construct a joint graph containing the QA context and KG, then use graph neural networks to reason over the two knowledge sources.

Another line of work explores less structured knowledge such as Wikipedia and dictionaries for commonsense reasoning [Xu *et al.*, 2021; Lv *et al.*, 2020]. Bhakthavatsalam *et al.* [2020] combine the knowledge from ConceptNet, WordNet, and other corpora to form 3.5M generic statements and show that this knowledge can help boost accuracy and explanation quality. Mitra *et al.* [2020] compares several ways of incorporating external knowledge from a relevant corpus for commensense question answering.

Recently, there are approaches to generate facts from pretrained language models to complement missing facts in the external knowledge source. Bosselut *et al.* [2019] finetune a pretrained model on ATOMIC for commonsense knowledge graph completion. Liu *et al.* [2021] directly prompt the GPT-3 model [Brown *et al.*, 2020] to get knowledge for reasoning.

Beyond commonsense reasoning, external knowledge can also help boost performance on other language processing tasks like open domain question answering [Yu *et al.*, 2021], conversational QA [Qin *et al.*, 2019], and text generation [Yu *et al.*, 2020]. Compared with prior work that uses extra modules (e.g., GNNs) or extra models (e.g., GPT-3), our external attention framework is extremely lightweight. It operates via a combination of non-parametric retrieval and text concatenation, which we show is highly effective, able to surpass human parity on the CommonsenseQA task.

5 Conclusion

We propose external attention as a lightweight framework for retrieving and integrating external knowledge for language understanding. Compared with self-attention which benefits from ever-increasing model sizes, external attention can bring related information from external sources to supplement the input. We demonstrate that this strategy can lead to considerable gains in performance with little additional computational cost. By leveraging knowledge from knowledge graphs, dictionaries, and training data, we show that our technology, KEAR, achieves human parity on the CommonsenseQA benchmark for the first time. For future work, we will apply the technique to other NLP tasks to improve language model performance with external knowledge.

References

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. Genericskb: A knowledge base of generic statements. *arXiv preprint arXiv:2005.00660*, 2020.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur. Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks, 2021.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2020.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, 2020.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. 2020.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. How additional knowledge can improve natural language commonsense question answering?, 2020.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, William B Dolan, Yejin Choi, and Jianfeng Gao. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5427–5436, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035, 2019.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.

Chenguang Song, Nianwen Ning, Yunlei Zhang, and Bin Wu. Knowledge augmented transformer for adversarial multidomain multiclassification multimodal fake news detection. *Neurocomputing*, 462:88–100, 2021.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI conference on artificial intelligence (AAAI)*, 2017.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, pages 5998–6008, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. Training data is more valuable than you think: A simple and effective method by retrieving from training data. *Association for Computational Linguistics (ACL)*, 2022.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. Fusing context into knowledge graph for commonsense question answering. In *Association for Computational Linguistics (ACL)*, 2021.

Yunzhi Yao, Shaohan Huang, Ningyu Zhang, Li Dong, Furu Wei, and Huajun Chen. Kformer: Knowledge injection in transformer feed-forward layers, 2022.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*, 2021.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *arXiv preprint arXiv:2010.04389*, 2020.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. *arXiv preprint arXiv:2110.04330*, 2021.