KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning

Bill Yuchen Lin[†] **Xinyue Chen**[‡] **Jamin Chen**[†] **Xiang Ren**[†]
{yuchen.lin,jaminche,xiangren}@usc.edu, kiwisher@sjtu.edu.cn

[†]Computer Science Department, University of Southern California
[‡]Computer Science Department, Shanghai Jiao Tong University

Abstract

Commonsense reasoning aims to empower machines with the human ability to make presumptions about ordinary situations in our daily life. In this paper, we propose a textual inference framework for answering commonsense questions, which effectively utilizes external, structured commonsense knowledge graphs to perform *explainable* inferences. The framework first grounds a question-answer pair from the semantic space to the knowledge-based symbolic space as a schema graph, a related sub-graph of external knowledge graphs. It represents schema graphs with a novel knowledge-aware graph network module named KagNet, and finally scores answers with graph representations. Our model is based on graph convolutional networks and LSTMs, with a hierarchical path-based attention mechanism. The intermediate attention scores make it transparent and interpretable, which thus produce trustworthy inferences. Using ConceptNet as the only external resource for Bert-based models, we achieved state-of-the-art performance on the CommonsenseQA, a large-scale dataset for commonsense reasoning. We open-source our code[1] to the community for future research in knowledge-aware commonsense reasoning.

1 Introduction

Human beings are rational and a major component of rationality is the ability to reason. Reasoning is the process of combining facts and beliefs to make new decisions (Johnson-Laird, 1980), as well as the ability to manipulate knowledge to draw inferences (Hudson and Manning, 2018). Commonsense reasoning utilizes the basic knowledge that reflects our natural understanding of the world and human behaviors, which is common to all humans.

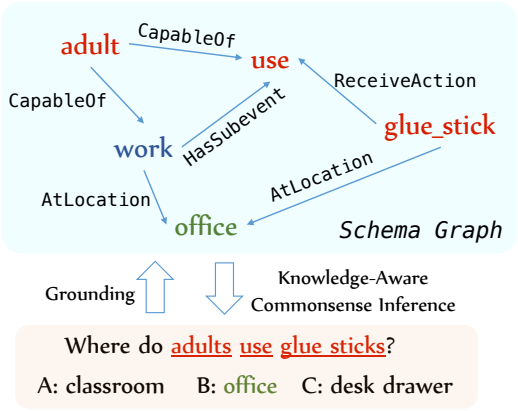

Figure 1: An example of using external commonsense knowledge (symbolic space) for inference in natural language commonsense questions (semantic space).

Empowering machines with the ability to perform commonsense reasoning has been seen as the bottleneck of artificial general intelligence (Davis and Marcus, 2015). Recently, there have been a few emerging large-scale datasets for testing machine commonsense with various focuses (Zellers et al., 2018; Sap et al., 2019b; Zellers et al., 2019). In a typical dataset, CommonsenseQA (Talmor et al., 2019), given a question like *"Where do adults use glue sticks?"*, with the answer choices being {classroom(✗), office (✓), desk drawer (✗)}, a commonsense reasoner is expected to differentiate the correct choice from other "distractive" candidates. False choices are usually highly related to the question context, but just less possible in real-world scenarios, making the task even more challenging. This paper aims to tackle the research question of how we can teach machines to make such commonsense inferences, particularly in the question-answering setting.

It has been shown that simply fine-tuning large, pre-trained language models such as Gpt (Radford et al., 2018) and Bert (Devlin et al., 2019)

[1] https://github.com/INK-USC/KagNet

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2829–2839, Hong Kong, China, November 3–7, 2019.

can be a very strong baseline method. However, there still exists a large gap between performance of said baselines and human performance. Reasoning with neural models is also lacking in transparency and interpretability. There is no clear way as to how they manage to answer commonsense questions, thus making their inferences dubious.

Merely relying on pre-training large language models on corpora cannot provide well-defined or reusable structures for explainable commonsense reasoning. We argue that it would be more beneficial to propose reasoners that can exploit commonsense knowledge bases (Speer et al., 2017; Tandon et al., 2017; Sap et al., 2019a). Knowledge-aware models can explicitly incorporate external knowledge as relational inductive biases (Battaglia et al., 2018) to enhance their reasoning capacity, as well as to increase the transparency of model behaviors for more interpretable results. Furthermore, a knowledge-centric approach is extensible through commonsense knowledge acquisition techniques (Li et al., 2016; Xu et al., 2018).

We propose a knowledge-aware reasoning framework for learning to answer commonsense questions, which has two major steps: schema graph grounding (§3) and graph modeling for inference (§4). As shown in Fig. 1, for each pair of question and answer candidate, we retrieve a graph from external knowledge graphs (e.g. `ConceptNet`) in order to capture the relevant knowledge for determining the plausibility of a given answer choice. The graphs are named "schema graphs" inspired by the `schema theory` proposed by Gestalt psychologists (Axelrod, 1973). The grounded schema graphs are usually much more complicated and noisier, unlike the ideal case shown in the figure.

Therefore, we propose a knowledge-aware graph network module to further effectively model schema graphs. Our model `KagNet` is a combination of graph convolutional networks (Kipf and Welling, 2017) and LSTMs, with a hierarchical path-based attention mechanism, which forms a `GCN-LSTM-HPA` architecture for path-based relational graph representation. Experiments show that our framework achieved a new state-of-the-art performance[2] on the `CommonsenseQA` dataset. Our model also works better then other methods with limited supervision, and provides human-readable results via intermediate attention scores.

[2]The highest score on the leaderboard as of the time when we submitted the paper (May 2019).

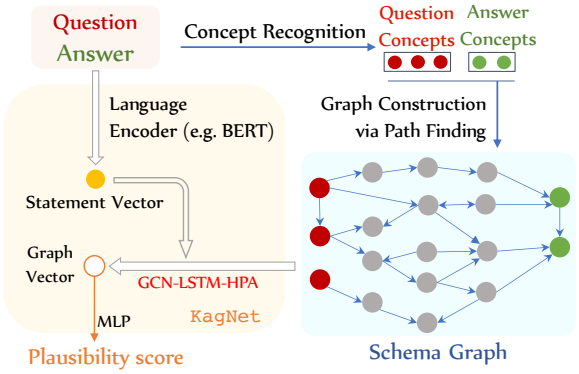

Figure 2: The overall workflow of the proposed framework with knowledge-aware graph network module.

2 Overview

In this section, we first formalize the commonsense question answering problem in a knowledge-aware setting, and then introduce the overall workflow of our framework.

2.1 Problem statement

Given a commonsense-required natural language question q and a set of N candidate answers $\{a_i\}$, the task is to choose one answer from the set. From a knowledge-aware perspective, we additionally assume that the question q and choices $\{a_i\}$ can be grounded as a **schema graph** (denoted as g) extracted from a large external knowledge graph G, which is helpful for measuring the plausibility of answer candidates. The knowledge graph $G = (V, E)$ can be defined as a fixed set of **concepts** V, and **typed edges** E describing semantic relations between concepts. Therefore, our goal is to effectively ground and model schema graphs to improve the reasoning process.

2.2 Reasoning Workflow

As shown in Fig. 2, our framework accepts a pair of question and answer (QA-pair) denoted as q and a. It first recognizes the mentioned concepts within them respectively from the concept set V of the knowledge graph. We then algorithmically construct the schema graph g by finding paths between pairs of mentioned concepts (§3).

The grounded schema graph is further encoded with our proposed knowledge-aware graph network module (§4). We first use a model-agnostic language encoder, which can either be trainable or a fixed feature extractor, to represent the QA-pair

as a statement vector. The statement vector serves as an additional input to a GCN-LSTM-HPA architecture for path-based attentive graph modeling to obtain a graph vector. The graph vector is finally fed into a simple multi-layer perceptron to score this QA-pair into a scalar ranging from 0 to 1, representing the plausibility of the inference. The answer candidate with the maximum plausibility score to the same question becomes the final choice of our framework.

3 Schema Graph Grounding

The grounding stage is three-fold: recognizing concepts mentioned in text (§3.1), constructing schema graphs by retrieving paths in the knowledge graph (§3.2), and pruning noisy paths (§3.3).

3.1 Concept Recognition

We match tokens in questions and answers to sets of mentioned concepts ($\mathcal{C}_q$ and $\mathcal{C}_a$ respectively) from the knowledge graph G (for this paper we chose to use ConceptNet due to its generality).

A naive approach to mentioned concept recognition is to exactly match n-grams in sentences with the surface tokens of concepts in V. For example, in the question "*Sitting too close to watch tv can cause what sort of pain?*", the exact matching result $\mathcal{C}_q$ would be {sitting, close, watch_tv, watch, tv, sort, pain, etc.}. We are aware of the fact that such retrieved mentioned concepts are not always perfect (e.g. "sort" is not a semantically related concept, "close" is a polysemous concept). How to efficiently retrieve contextually-related knowledge from noisy knowledge resources is still an open research question by itself (Weissenborn et al., 2017; Khashabi et al., 2017), and thus most prior works choose to stop here (Zhong et al., 2018; Wang et al., 2019b). We enhance this straightforward approach with some rules, such as soft matching with lemmatization and filtering of stop words, and further deal with noise by pruning paths (§3.3) and reducing their importance with attention mechanisms (§4.3).

3.2 Schema Graph Construction

ConceptNet. Before diving into the construction of schema graphs, we would like to briefly introduce our target knowledge graph ConceptNet. ConceptNet can be seen as a large set of triples of the form (h, r, t), like (ice, HasProperty, cold), where h and t represent head and tail concepts in the concept set V and r is a certain relation type from the pre-defined set R. We delete and merge the original 42 relation types into 17 types, in order to increase the density of the knowledge graph[3] for grounding and modeling.

Sub-graph Matching via Path Finding. We define a schema graph as a sub-graph g of the whole knowledge graph G, which represents the related knowledge for reasoning a given question-answer pair with minimal additional concepts and edges. One may want to find a minimal spanning sub-graph covering all the question and answer concepts, which is actually the NP-complete "Steiner tree problem" in graphs (Garey and Johnson, 1977). Due to the incompleteness and tremendous size of ConceptNet, we find that it is impractical to retrieve a comprehensive but helpful set of knowledge facts this way. Therefore, we propose a straightforward yet effective graph construction algorithm via path finding among mentioned concepts ($\mathcal{C}_q \cup \mathcal{C}_a$).

Specifically, for each question concept $c_i \in \mathcal{C}_q$ and answer concept $c_j \in \mathcal{C}_a$, we can efficiently find paths between them that are shorter than k concepts[4]. Then, we add edges, if any, between the concept pairs within $\mathcal{C}_q$ or $\mathcal{C}_a$.

3.3 Path Pruning via KG Embedding

To prune irrelevant paths from potentially noisy schema graphs, we first utilize knowledge graph embedding (KGE) techniques, like TransE (Wang et al., 2014), to pre-train concept embeddings $\mathbf{V}$ and relation type embeddings $\mathbf{R}$, which are also used as initialization for KagNet (§4). In order to measure the quality of a path, we decompose it into a set of triples, the confidence of which can be directly measured by the scoring function of the KGE method (i.e. the confidence of triple classification). Thus, we score a path with the multiplication product of the scores of each triple in the path, and then we empirically set a threshold for pruning (§5.3).

4 Knowledge-Aware Graph Network

The core component of our reasoning framework is the knowledge-aware graph network module KagNet. The KagNet first encodes plain structures of schema graphs with graph convolutional networks (§4.1) to accommodate pre-trained con-

[3] The full mapping list is in the appendix.

[4] We set $k = 4$ in experiments to gather three-hop paths.

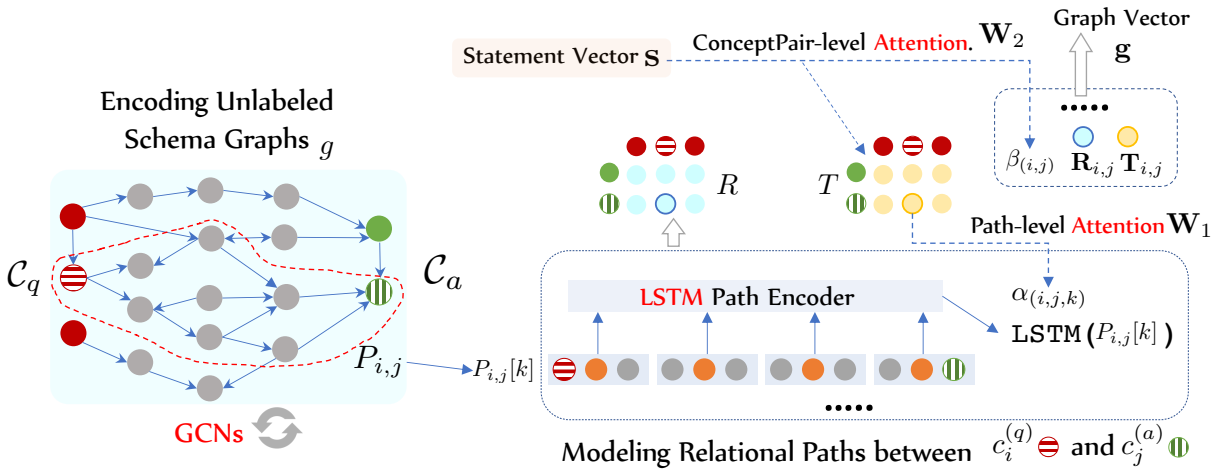

Figure 3: Illustration of the GCN−LSTM−HPA architecture for the proposed KAGNET module.

cept embeddings in their particular context within schema graphs. It then utilizes LSTMs to encode the paths between $\mathcal{C}_q$ and $\mathcal{C}_a$, capturing multi-hop relational information (§4.2). Finally, we apply a hierarchical path-based attention mechanism (§4.3) to complete the GCN−LSTM−HPA architecture, which models relational schema graphs with respect to the paths between question and answer concepts.

4.1 Graph Convolutional Networks

Graph convolutional networks (GCNs) encode graph-structured data by updating node vectors via pooling features of their adjacent nodes (Kipf and Welling, 2017). Our intuition for applying GCNs to schema graphs is to 1) contextually refine the concept vectors and 2) capture structural patterns of schema graphs for generalization.

Although we have obtained concept vectors by pre-training (§3.3), the representations of concepts still need to be further accommodated to their specific schema graphs context. Think of polysemous concepts such as "close" (§3.1), which can either be a verb concept like in "close the door" or an adjective concept meaning "a short distance apart". Using GCNs to update the concept vector with their neighbors is thus helpful for disambiguation and contextualized concept embedding. Also, the pattern of schema graph structures provides potentially valuable information for reasoning. For instance, shorter and denser connections between question and answer concepts could mean higher plausibility under specific contexts.

As many works show (Marcheggiani and Titov, 2017; Zhang et al., 2018), relational GCNs (Schlichtkrull et al., 2018) usually over-parameterize the model and cannot effectively utilize multi-hop relational information. We thus apply GCNs on the plain version (unlabeled, non-directional) of schema graphs, ignoring relation types on the edges. Specifically, the vector for concept $c_i \in \mathcal{V}_g$ in the schema graph g is initialized by their pre-trained embeddings at first ($h_i^{(0)} = \mathbf{V}_i$). Then, we update them at the $(l+1)$-th layer by pooling features of their neighboring nodes (N_i) and their own at the l-th layer with an non-linear activation function σ:

$$h_i^{(l+1)} = \sigma(W_{self}^{(l)} h_i^{(l)} + \sum_{j \in N_i} \frac{1}{|N_i|} W^{(l)} h_j^{(l)})$$

4.2 Relational Path Encoding

In order to capture the relational information in schema graphs, we propose an LSTM-based path encoder on top of the outputs of GCNs. Recall that our graph representation has a special purpose: "*to measure the plausibility of a candidate answer to a given question*". Thus, we propose to represent graphs with respect to the paths between question concepts $\mathcal{C}_q$ and answer concepts $\mathcal{C}_a$.

We denote the k-th path between i-th question concept $c_i^{(q)} \in \mathcal{C}_q$ and j-th answer concept $c_j^{(a)} \in \mathcal{C}_a$ as $P_{i,j}[k]$, which is a sequence of triples:

$$P_{i,j}[k] = [(c_i^{(q)}, r_0, t_0), ..., (t_{n-1}, r_n, c_j^{(a)})]$$

Note that the relations are represented with trainable relation vectors (initialized with pre-trained relation embeddings), and concept vectors are the GCNs' outputs ($h^{(l)}$). Thus, each triple can be represented by the concatenation of the three

corresponding vectors. We employ LSTM networks to encode these paths as sequences of triple vectors, taking the concatenation of the first and the last hidden states:

$$\mathbf{R}_{i,j} = \frac{1}{|P_{i,j}|} \sum_{k} \texttt{LSTM}(P_{i,j}[k])$$

The above $\mathbf{R}_{i,j}$ can be viewed as the latent relation between the question concept $c_i^{(q)}$ and the answer concept $c_j^{(a)}$, for which we aggregate the representations of all the paths between them in the schema graph. Now we can finalize the vector representation of a schema graph $\mathbf{g}$ by aggregating all vectors in the matrix $\mathbf{R}$ using mean pooling:

$$\mathbf{T}_{i,j} = \texttt{MLP}([\mathbf{s}\ ;\ \mathbf{c_q^{(i)}}\ ;\ \mathbf{c_a^{(j)}}])$$
$$\mathbf{g} = \frac{\sum_{i,j}[\mathbf{R}_{i,j}\ ;\ \mathbf{T}_{i,j}]}{|\mathcal{C}_q| \times |\mathcal{C}_a|}$$

, where $[\cdot\ ;\ \cdot]$ means concatenation of two vectors.

The statement vector $\mathbf{s}$ in the above equation is obtained from a certain language encoder, which can either be a trainable sequence encoder like LSTM or features extracted from pre-trained universal language encoders like GPT/BERT). To encode a question-answer pair with universal language encoders, we simply create a sentence combining the question and the answer with a special token ("`question`+ [sep] + `answer`"), and then use the vector of '[cls]' as suggested by prior works (Talmor et al., 2019)..

We concatenate $\mathbf{R}_{i,j}$ with an additional vector $\mathbf{T}_{i,j}$ before doing average pooling. The $\mathbf{T}_{i,j}$ is inspired from the `Relation Network` (Santoro et al., 2017), which also encodes the latent relational information yet from the context in the statement s instead of the schema graph g. Simply put, we want to combine the relational representations of a pair of question/answer concepts from both the schema graph side (symbolic space) and the language side (semantic space).

Finally, the plausibility score of the answer candidate a to the question q can be computed as $\texttt{score}(q, a) = \texttt{sigmoid}(\texttt{MLP}(\mathbf{g}))$.

4.3 Hierarchical Attention Mechanism

A natural argument against the above `GCN-LSTM-mean` architecture is that mean pooling over the path vectors does not always make sense, since some paths are more important than others for reasoning. Also, it is usually not the case that all pairs of question and answer concepts equally contribute to the reasoning. Therefore, we propose a hierarchical path-based attention mechanism to selectively aggregate important path vectors and then more important question-answer concept pairs. This core idea is similar to the work of Yang et al. (2016), which proposes a document encoder that has two levels of attention mechanisms applied at the word- and sentence-level. In our case, we have path-level and concept-pair-level attention for learning to contextually model graph representations. We learn a parameter matrix $\mathbf{W}_1$ for path-level attention scores, and the importance of the path $P_{i,j}[k]$ is denoted as $\hat{\alpha}_{(i,j,\cdot)}$.

$$\alpha_{(i,j,k)} = \mathbf{T}_{i,j}\ \mathbf{W}_1\ \texttt{LSTM}(P_{i,j}[k]),$$
$$\hat{\alpha}_{(i,j,\cdot)} = \texttt{SoftMax}(\alpha_{(i,j,\cdot)}),$$
$$\hat{\mathbf{R}}_{i,j} = \sum_{k} \hat{\alpha}_{(i,j,k)} \cdot \texttt{LSTM}(P_{i,j}[k]).$$

Afterwards, we similarly obtain the attention over concept-pairs.

$$\beta_{(i,j)} = \mathbf{s}\ \mathbf{W}_2\ \mathbf{T}_{i,j}$$
$$\hat{\beta}_{(\cdot,\cdot)} = \texttt{SoftMax}(\beta_{(\cdot,\cdot)})$$
$$\hat{\mathbf{g}} = \sum_{i,j} \hat{\beta}_{(i,j)}[\hat{\mathbf{R}}_{i,j}\ ;\ \mathbf{T}_{i,j}]$$

The whole `GCN-LSTM-HPA` architecture is illustrated in Figure 3. To sum up, we claim that the KAGNET is a graph neural network module with the `GCN-LSTM-HPA` architecture that models relational graphs for relational reasoning under the context of both knowledge symbolic space and language semantic space.

5 Experiments

We introduce our setups of the `CommonsenseQA` dataset (Talmor et al., 2019), present the baseline methods, and finally analyze experimental results.

5.1 Dataset and Experiment Setup

The `CommonsenseQA` dataset consists of 12,102 (v1.11) natural language questions in total that require human commonsense reasoning ability to answer, where each question has five candidate answers (hard mode). The authors also release an easy version of the dataset by picking two random terms/phrases for sanity check. `CommonsenseQA` is directly gathered from real human annotators and covers a broad range of

Model	10(%) of IHtrain		50(%) of IHtrain		100(%) of IHtrain	
	IHdev-Acc.(%)	IHtest-Acc.(%)	IHdev-Acc.(%)	IHtest-Acc.(%)	IHdev-Acc.(%)	IHtest-Acc.(%)
Random guess	20.0	20.0	20.0	20.0	20.0	20.0
GPT-FINETUNING	27.55	26.51	32.46	31.28	47.35	45.58
GPT-**KAGNET**	28.13	**26.98**	33.72	**32.33**	48.95	**46.79**
BERT-BASE-FINETUNING	30.11	29.78	38.66	36.83	53.48	53.26
BERT-BASE-**KAGNET**	31.05	**30.94**	40.32	**39.01**	55.57	**56.19**
BERT-LARGE-FINETUNING	35.71	32.88	55.45	49.88	60.61	55.84
BERT-LARGE-**KAGNET**	36.82	**33.91**	58.73	**51.13**	62.35	**57.16**
Human Performance	-	*88.9*	-	*88.9*	-	*88.9*

Table 1: Comparisons with large pre-trained language model fine-tuning with different amount of training data.

types of commonsense, including spatial, social, causal, physical, temporal, etc. To the best of our knowledge, CommonsenseQA may be the most suitable choice for us to evaluate supervised learning models for question answering.

For the comparisons with the reported results in the CommonsenseQA's paper and leaderboard, we use the official split (9,741/1,221/1,140) named (OFtrain/OFdev/OFtest). Note that the performance on OFtest can only be tested by submitting predictions to the organizers. To efficiently test other baseline methods and ablation studies, we choose to use randomly selected 1,241 examples from the training data as our in-house data, forming an (8,500/1,221/1,241) split denoted as (IHtrain/IHdev/IHtest). All experiments are using the random-split setting as the authors suggested, and three or more random states are tested on development sets to pick the best-performing one.

5.2 Compared Methods

We consider two different kinds of baseline methods as follows:

• *Knowledge-agnostic Methods.* These methods either use no external resources or only use unstructured textual corpora as additional information, including gathering textual snippets from search engine or large pre-trained language models like BERT-LARGE. QABILINEAR, QACOMPARE, ESIM are three supervised learning models for natural language inference that can be equipped with different word embeddings including GloVe and ELMO. BIDAF++ utilizes Google web snippets as context and is further augmented with a self-attention layer while using ELMO as input features. GPT/BERT-LARGE are fine-tuning methods with an additional linear layer for classification as the authors suggested. They both add a special token '[sep]' to the input and use the hidden state of the '[cls]' as the input to the linear layer. More details about them can be found in the dataset paper (Talmor et al., 2019).

• *Knowledge-aware Methods.* We also *adopt* some recently proposed methods of incorporating knowledge graphs for question answering. KV-MEM (Mihaylov and Frank, 2018) is a method that incorporates retrieved triples from ConceptNet at the word-level, which uses a key-valued memory module to improve the representation of each token individually by learning an attentive aggregation of related triple vectors. CBPT (Zhong et al., 2018) is a plug-in method of assembling the predictions of any models with a straightforward method of utilizing pre-trained concept embeddings from ConceptNet. TEXTGRAPHCAT (Wang et al., 2019c) concatenates the graph-based and text-based representations of the statement and then feed it into a classifier. We create sentence template for generating sentences and then feed retrieved triples as additional text inputs as a baseline method TRIPLESTRING. Rajani et al. (2019) propose to collect human explanations for commonsense reasoning from annotators as additional knowledge (COS-E), and then train a language model based on such human annotations for improving the model performance.

5.3 Implementation Details of KagNet

Our best (tested on OFdev) settings of KAGNET have two GCN layers (100 dim, 50dim respectively), and one bidirectional LSTMs (128dim) . We pre-train KGE using TransE (100 dimension) initialized with GloVe embeddings. The statement encoder in use is BERT-LARGE, which works as a pre-trained sentence encoder to obtain fixed features for each pair of question and answer candidate. The paths are pruned with path-score threshold set to *0.15*, keeping *67.21%* of the original

Model	OFdev-Acc.(%)	OFtest-Acc.(%)
Random guess	20.0	20.0
BIDAF++	-	32.0
QACOMPARE+GLOVE	-	25.7
QABLINEAR+GLOVE	-	31.5
ESIM+ELMO	-	32.8
ESIM+GLOVE	-	34.1
GPT-FINETUNING	47.11	45.5
BERT-BASE-FINETUNING	53.57	53.0
BERT-LARGE-FINETUNING	62.34	56.7
COS-E (w/ additional annotations)	-	58.2
KAGNET (Ours)	64.46	**58.9**
Human Performance	-	*88.9*

Table 2: Comparison with official benchmark baseline methods using the official split on the leaderboard.

	Easy Mode		Hard Mode	
Model	IHdev.(%)	IHtest.(%)	IHdev.(%)	IHtest.(%)
Random guess	33.3	33.3	20.0	20.0
BLSTMS	80.15	78.01	34.79	32.12
+ KV-MN	81.71	79.63	35.70	33.43
+ CSPT	81.79	80.01	35.31	33.61
+ TEXTGRAPHCAT	82.68	81.03	34.72	33.15
+ TRIPLESTRING	79.11	76.02	33.19	31.02
+ **KAGNET**	83.26	**82.15**	36.38	**34.57**
Human Performance	-	*99.5*	-	*88.9*

Table 3: Comparisons with knowledge-aware baseline methods using the in-house split (both easy and hard mode) on top of BLSTM as the sentence encoder.

paths. We did not conduct pruning on concept pairs with less than three paths. For very few pairs with none path, $\hat{\mathbf{R}}_{(i,j)}$ will be a randomly sampled vector. We learn our KAGNET models with Adam optimizers (Kingma and Ba, 2015). In our experiments, we found that the recall of ConceptNet on commonsense questions and answers is very high (over 98% of QA-pairs have more than one grounded concepts).

5.4 Performance Comparisons and Analysis

Comparison with standard baselines.
As shown in Table 2, we first use the official split to compare our model with the baseline methods reported on the paper and leaderboard. BERT and GPT-based pre-training methods are much higher than other baseline methods, demonstrating the ability of language models to store commonsense knowledge in an implicit way. This presumption is also investigated by Trinh and Le (2019) and Wang et al. (2019). Our proposed framework achieves an absolute increment of 2.2% in accuracy on the test data, a state-of-the-art performance.

We conduct the experiments with our in-house splits to investigate whether our KAGNET can also work well on other universal language encoders (GPT and BERT-BASE), particularly with different fractions of the dataset (say 10%, 50%, 100% of the training data). Table 1 shows that our KAGNET-based methods using fixed pre-trained language encoders outperform fine-tuning themselves in all settings. Furthermore, we find that the improvements in a small data situation (10%) is relatively limited, and we believe an important future research direction is thus few-shot learning for commonsense reasoning.

Comparison with knowledge-aware baselines.
To compare our model with other adopted baseline methods that also incorporate ConceptNet, we set up a bidirectional LSTM networks-based model for our in-house dataset. Then, we add baseline methods and KAGNET onto the BLSTMs to compare their abilities to utilize external knowledge[5]. Table 3 shows the comparisons under both easy mode and hard mode, and our methods outperform all knowledge-aware baseline methods by a large margin in terms of accuracy. Note that we compare our model and the CoS-E in Table 2. Although CoS-E also achieves better result than only fine-tuning BERT by training with human-generated explanations, we argue that our proposed KagNet does not utilize any additional human efforts to provide more supervision.

Ablation study on model components.
To better understand the effectiveness of each component of our method, we have done ablation study as shown in Table 4. We find that replacing our GCN-LSTM-HPA architecture with traditional relational GCNs, which uses separate weight matrices for different relation types, results in worse performance, due to its over-parameterization. The attention mechanisms matters almost equally in two levels, and pruning also effectively filters noisy paths.

Error analysis.
In the failed cases, there are three kinds of hard problems that KAGNET is still not good at.

- **negative reasoning:** the grounding stage is not sensitive to the negation words, and thus can choose exactly opposite answers.
- **comparative reasoning strategy:** For the

[5]We do LSTM-based setup because it is non-trivial to apply token-level knowledge-aware baseline methods for complicated pre-trained encoders like BERT.

Model	IHdev.(%)	IHtest.(%)
KAGNET (STANDARD)	62.35	57.16
: replace GCN-HPA-LSTM w/ R-GCN	60.01	55.08
: w/o GCN	61.84	56.11
: #GCN Layers = 1	62.05	57.03
: w/o Path-level Attention	60.12	56.05
: w/o QAPair-level Attention	60.39	56.13
: using all paths (w/o pruning)	59.96	55.27

Table 4: Ablation study on the KAGNET framework.

questions with more than one highly plausible answers, the commonsense reasoner should benefit from explicitly investigating the difference between different answer candidates, while KAGNET training method is not capable of doing so.

- **subjective reasoning:** Many answers actually depend on the "personality" of the reasoner. For instance, "*Traveling from new place to new place is likely to be what?*" The dataset gives the answer as "exhilarating" instead of "exhausting", which we think is more like a personalized subjective inference instead of common sense.

5.5 Case Study on Interpretibility

Our framework enjoys the merit of being more transparent, and thus provides more interpretable inference process. We can understand our model behaviors by analyzing the hierarchical attention scores on the question-answer concept pairs and path between them.

Figure 4 shows an example how we can analyze our KAGNET framework through both pair-level and path-level attention scores. We first select the concept-pairs with highest attention scores and then look at the (one or two) top-ranked paths for each selected pair. We find that paths located in this way are highly related to the inference process and also shows that noisy concepts like 'fountain' will be diminished while modeling.

5.6 Model Transferability.

We study the transferability of a model that is trained on CommonsenseQA (CSQA) by directly testing it with another task while fixing its parameters. Recall that we have obtained a BERT-LARGE model and a KAGNET model trained on CSQA. Now we denoted them as CSQA-BL and CSQA-KN to suggest that they are not trainable anymore.

In order to investigate their transferability, we separately test them on SWAG (Zellers et al., 2018)

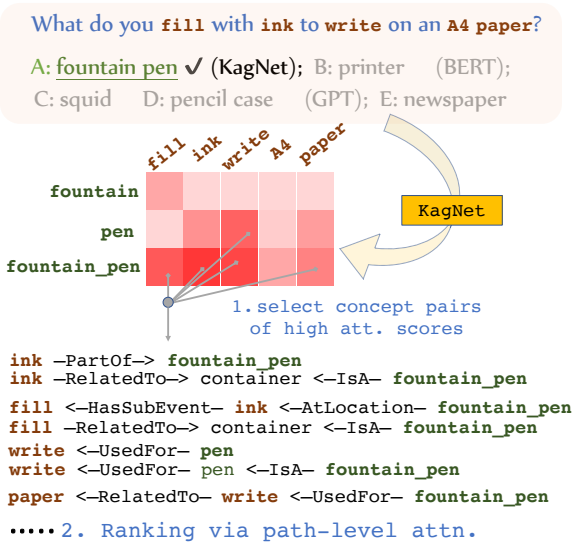

Figure 4: An example of interpreting model behaviors by hierarchical attention scores.

and WSC (Levesque, 2011) datasets. We first test them the 20k validation examples in SWAG. CSQA-BL has an accuracy of 56.53%, while our fixed CSQA-KN model achieves 59.01%. Similarly, we also test both models on the WSC-QA, which is converted from the WSC pronoun resolution to a multi-choice QA task.

The CSQA-BL achieves an accuracy of 51.23%, while our model CSQA-KN scores 53.51%. These two comparisons further support our assumption that KAGNET, as a knowledge-centric model, is more extensible in commonsense reasoning. As we expect for a good knowledge-aware frameworks to behave, our KAGNET indeed enjoys better transferablity than only fine-tuning large language encoders like BERT.

5.7 Recent methods on the leaderboard.

We argue that the KAGNET utilizes the ConceptNet as the only external resource and other methods are improving their performance in orthogonal directions: 1) we find that most of the other recent submissions (as of Aug. 2019) with public information on the leaderboard utilize larger additional textual corpora (e.g. top 10 matched sentences in full Wikipedia via information retrieval tools), and fine-tuning on larger pre-trained encoders, such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019). 2) there are also models using multi-task learning to transfer knowledge from other reading comprehension datasets, such as RACE (Lai et al., 2017) and

OpenBookQA (Mihaylov et al., 2018).

An interesting fact is that the best performance on the OFtest set is still achieved the original fine-tuned RoBERTa model, which is pre-trained with copora much larger than BERT. All other RoBERTa-extended methods have negative improvements. We also use statement vectors from RoBERTa as the input vectors for KagNet, and find that the performance on OFdev marginally improves from 77.47% to 77.56%. Based on our above-mentioned failed cases in error analysis, we believe fine-tuning RoBERTa has achieved the limit due to the annotator biases of the dataset and the lack of comparative reasoning strategies.

6 Related Work

Commonsense knowledge and reasoning. There is a recent surge of novel large-scale datasets for testing machine commonsense with various focuses, such as situation prediction (SWAG) (Zellers et al., 2018), social behavior understanding (Sap et al., 2019a,b), visual scene comprehension (Zellers et al., 2019), and general commonsense reasoning (Talmor et al., 2019), which encourages the study of supervised learning methods for commonsense reasoning. Trinh and Le (2018) find that large language models show promising results in WSC resolution task (Levesque, 2011), but this approach can hardly be applied in a more general question answering setting and also not provide explicit knowledge used in inference. A unique merit of our KagNet method is that it provides grounded explicit knowledge triples and paths with scores, such that users can better understand and put trust in the behaviors and inferences of the model.

Injecting external knowledge for NLU. Our work also lies in the general context of using external knowledge to encode sentences or answer questions. Yang and Mitchell (2017) are the among first ones to propose to encode sentences by keeping retrieving related entities from knowledge bases and then merging their embeddings into LSTM networks computations, to achieve a better performance on entity/event extraction tasks. Weissenborn et al. (2017), Mihaylov and Frank (2018), and Annervaz et al. (2018) follow this line of works to incorporate the embeddings of related knowledge triples at the word-level and improve the performance of natural language understanding tasks. In contrast to our work, they do not explicitly impose graph-structured knowledge into models , but limit its potential within transforming word embeddings to concept embeddings.

Some other recent attempts (Zhong et al., 2018; Wang et al., 2019c) to use ConceptNet graph embeddings are adopted and compared in our experiments (§5). Rajani et al. (2019) propose to manually collect more human explanations for correct answers as additional supervision for auxiliary training. KagNet-based framework focuses on injecting external knowledge as an explicit graph structure, and enjoys the relational reasoning capacity over the graphs.

Relational reasoning. KagNet can be seen as a knowledge-augmented Relation Network module (RN) (Santoro et al., 2017), which is proposed for the visual question answering task requiring relational reasoning (i.e. questions about the relations between multiple 3D-objects in an image). We view the concepts in the questions and answers as objects and effectively utilize external knowledge graphs to model their relations from both semantic and symbolic spaces (§4.2), while prior methods mainly work on the semantic one.

7 Conclusion

We propose a knowledge-aware framework for learning to answer commonsense questions. The framework first constructs schema graphs to represent relevant commonsense knowledge, and then model the graphs with our KagNet module. The module is based on a GCN-LSTM-HPA architecture, which effectively represent graphs for relational reasoning purpose in a transparent, interpretable way, yielding a new state-of-the-art results on a large-scale general dataset for testing machine commonsense. Future directions include better question parsing methods to deal with negation and comparative question answering, as well as incorporating knowledge to visual reasoning.

Acknowledgments

This work has been supported in part by National Science Foundation SMA 18-29268, DARPA MCS and GAILA, IARPA BETTER, Schmidt Family Foundation, Amazon Faculty Award, Google Research Award, Snapchat Gift and JP Morgan AI Research Award. We would like to thank all the collaborators in the INK research lab for their constructive feedback on the work.

References

K. M. Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In *Proc. of NAACL-HLT*.

Robert Axelrod. 1973. Schema theory: An information processing model of perception and cognition. *American political science review*, 67(4):1248–1266.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58(9):92–103.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*.

Michael R Garey and David S. Johnson. 1977. The rectilinear steiner tree problem is np-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. In *Proc. of ICLR*.

Philip N Johnson-Laird. 1980. Mental models in cognitive science. *Cognitive science*, 4(1):71–115.

Daniel Khashabi, Tushar Khot, Ashutosh Sabharwal, and Dan Roth. 2017. Learning what is essential in questions. In *Proc. of CoNLL*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proc. of EMNLP*.

Hector J. Levesque. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proc. of ACL*.

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

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proc. of EMNLP*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proc. of ACL*.

Tzvetan Mihaylov, Peter F. Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proc. of EMNLP*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proc. of ACL*.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. 2017. A simple neural network module for relational reasoning. In *Proc. of NIPS*.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proc. of AAAI*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proc. of European Semantic Web Conference*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proc. of AAAI*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proc. of NAACL-HLT*.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2017. Webchild 2.0 : Fine-grained commonsense knowledge distillation. In *Proc. of ACL*.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Trieu H. Trinh and Quoc V. Le. 2019. Do language models have common sense? *OpenReview*, ICLR submissions.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019a. Does it make sense? and why? a pilot study for sense making and explanation. In *Proc. of ACL*.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019b. Improving natural language inference using external knowledge in the science questions domain. In *Proc. of AAAI*.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019c. Improving natural language inference using external knowledge in the science questions domain.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proc. of AAAI*.

Dirk Weissenborn, Tomáš Kočiskỳ, and Chris Dyer. 2017. Dynamic integration of background knowledge in neural nlu systems. *arXiv preprint arXiv:1706.02596*.

Frank Xu, Bill Yuchen Lin, and Kenny Q. Zhu. 2018. Automatic extraction of commonsense locatednear knowledge. In *Proc. of ACL*.

Bishan Yang and Tom Michael Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proc. of ACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ArXiv*, abs/1906.08237.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *Proc. of NAACL-HLT*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proc. of CVPR*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proc. of EMNLP*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proc. of EMNLP*.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2018. Improving question answering by commonsense-based pre-training. *ArXiv*, abs/1809.03568.