Knowledge graph augmented advanced learning models for commonsense reasoning

Knowledge graph augmented advanced learning models for commonsense reasoning

Akhilesh Pothuri [1], Hari Sai Raghuram Veeramallu [2], Pooja Malik [3]

Department of Computer Science, Shiv Nadar University, Greater Noida, India
E-mail: ap354@snu.edu.in [1], hv829@snu.edu.in [2], pooja.malik@snu.edu.in [3]

Abstract. Machine learning is the key solution to many AI issues, but learning models rely heavily on specific training data. While a Bayesian setup can be used to incorporate some learning patterns with previous knowledge, those patterns can not access any organized world knowledge on requirements. The primary objective is to enable human-capable machines in ordinary everyday circumstances to estimate and make presumptions. In this paper we propose to respond to such common sense issues through a textual inference system with external, organized common sense graphs for explanatory inferences. The framework is based on a schematic map as a pair of questions and answers, a linked subgraph from the semantine to the symbolic space of knowledge-based external information. It displays a schematic map with a new network graphic module for information knowledge and performance with graph representations. LSTMs and graphical networks with a hierarchical attention-based direction are the basis of our model. It is flexible and understandable from the intermediate attention scores, leading to confident results. We also achieved state-of-the-art reliability on CommonsenseQA, a broad database of common sense reasoning utilizing ConceptNet as the only external tool for BERT-based models.

Keywords: Machine learning, Natural language processing, Knowledge graphs, Artificial Intelligence, LSTM, Commonsense QA, Neural networks, ConceptNet, Hierarchical attention mechanism.

1. Introduction

Human beings are logical, and the ability to reason is a significant component of rationality.Reasoning is the process of synthesis of facts and theories, of making new decisions [1] and of manipulating knowledge to extract conclusions [2]. Commonsense logic incorporates the basic knowledge which represents our natural world perception and individual actions that are familiar to all people. Machine learning today focuses on algorithms that can be conditioned on task-specific samples labelled and unlabelled. It has been seen as the bottleneck of artificial intelligence to empower machines with the ability to perform commonsense reasoning [3]. A few large-scale data sets have recently emerged to test machine commonsense with a different focus [4][5][6].

The reasoner for common sense is required to discern the right option among other "distractive" applicants in a typical dataset, CommonsenseQA [7], with the questions "*Where do kids play?*," with response options as a classroom (x), park(✓), office(x). False choices are typically strongly related to the meaning of the scenario, but less likely in real-world situations, making the task even more difficult.

The question we are addressing is: can we develop learning models that can be trained so that, apart from learning based on data from training, we can infuse a broad set of global knowledge into a prediction? With the use of world knowledge, we are thinking of structured knowledge for general purposes that do not need to be specific to a particular domain. There are also knowledge bases created by humans, such as Freebase [8] and WordNet [9]. In this knowledge base, the knowledge contains common knowledge and partially encompasses common sense and domain knowledge [10]. Knowledge Graphs [11] are a popular source of such structured world knowledge. It is proved to be a

very strong base method to simply fine-tune big, pre-entrained language models like GPT [12] and BERT [13]. However the performance of such baselines and human performance also vary greatly.

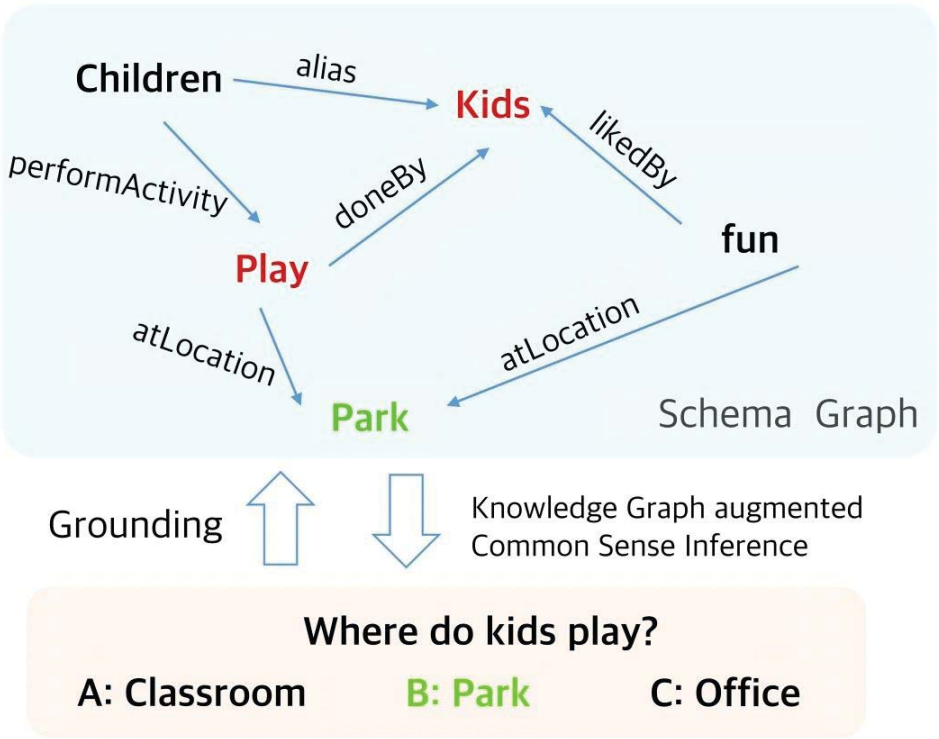

Figure 1. This is an example of using outside common sense knowledge to infer common sense questions in natural languages.

Simply relying on corporate pre-training large-scale language models can not provide an explanatory common sense framework. We suggest that introducing reasoners that can use common knowledge foundations would be more effective [14][5]. In order to find answers to common questions, we propose a knowledge-conscious reasoning framework that contains two key milestones: schematics and graph modelling. The diagrams are known as the "schema graphs" that are inspired by the schematic theory proposed by psychologists from Gestalt [15]. The derived schematic diagrams are usually much more complex and noisy. Our model is composed of LSTMs and graph-based networks [16]with a hierarchical path-based attention care mechanism that is an architecture of the GCN-LSTM-HPA for a path-based linked graphic representation. The experiments performed indicate that our architecture has reached an optimal performance. This system works better than other models of minimal control and produces human-readable outcomes by moderate attention levels.

2. Overview

In this section, we will introduce the problem to be addressed and then briefly go through the process and flow of the framework.

2.1 Answering Commonsense Questions

Given a generic commonsense based question using simple natural language and say some potential answers to that question, the problem is to find out a single answer among all those potential answers (Assuming N potential options). From the perspective of Knowledge Graphs, we need the question and answers to be grounded into a schema graph *sg*. The schema graph is very helpful in measuring the

plausibility of answer choices and thus is extracted from the external worldly Knowledge Base. The Knowledge Graph $G = (Vg, Eg)$ can be understood as a set of edges Eg and concepts Vg defining the relation between said concepts. Hence, to obtain the desired results, we have to ground and model the schema graphs to augment the reasoning for such questions.

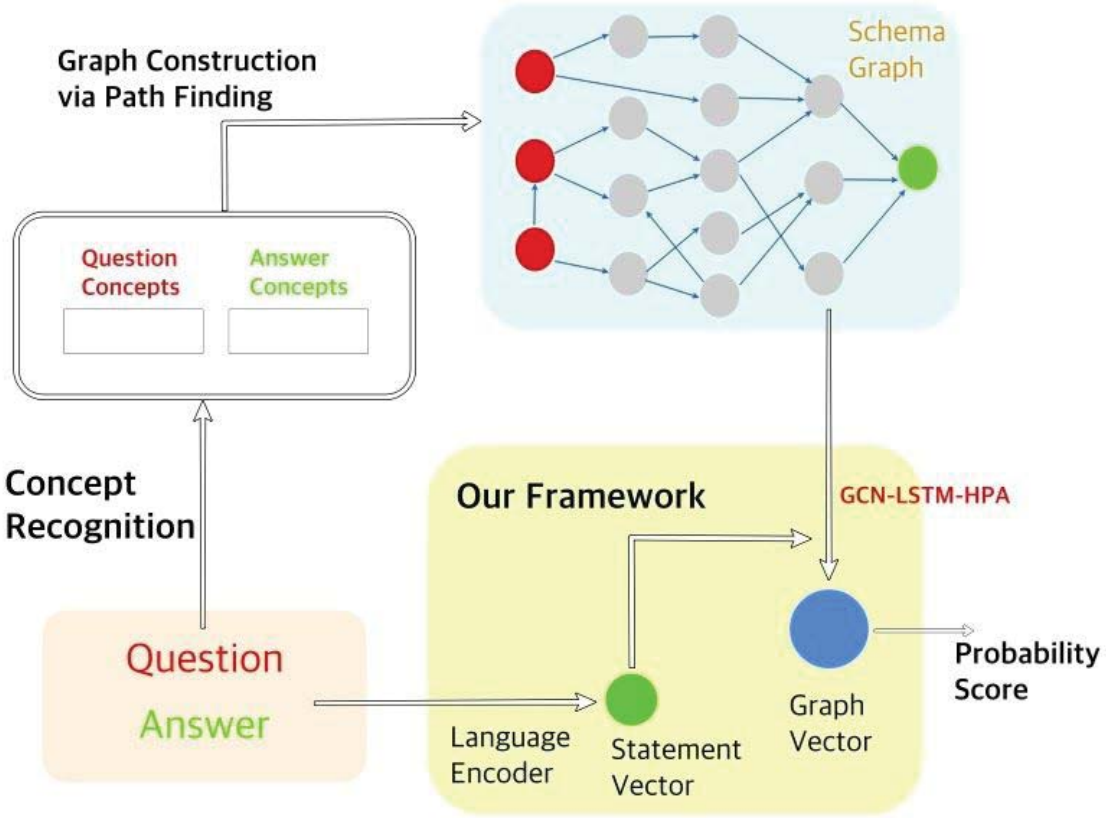

Figure 2. The Workflow of our proposed framework with schema graph module

2.2. Framework Workflow

Our framework takes a Question-Answer pair (qp, ap) and recognizes the concepts within the pair from an already existing concept set Vg (*from the Knowledge Graph G*). The schema graph is then constructed by finding paths between the concepts present [17].

Further encoding of the grounded schema graph is done with the proposed Knowledge Graph model. The Question-Answer pair is represented as a vector using a model-agnostic language encoder. The vector is an additional input to the GCN-LSTM-HPA architecture to obtain a graph vector using path-based attention modelling. A simple multi-layer perceptron takes the graph vector generated to score this (qp, ap) pair between 0 and 1 (t*his represents a probable inference*). The answer choice having the maximum probability score becomes our choice.

3. Schema Graph Grounding

Grounding is three-fold: All the concepts mentioned in the text should be identified, building diagrams by finding paths in the knowledge diagram and sniffing off noisy paths.

3.1. Concept Identification

In questions and answers, we match the tokens to the sets of the mentioned concepts (CN_{qp} and CN_{ap}) in the graph *G* (*for this paper, we have chosen ConceptNet because of their generality*). Matching n-grams precisely in phrases keeping concepts in Vg on the surface is a naive approach to such concept identification problems. For example, in the question "*Improper body posture while studying causes what sort of pain?*", the question results in CN_{qp} would be {improper, body_posture, body, posture, studying, pain, etc.}.We are mindful that these recovered ideas are not always perfect. The reality that qualitative information from brilliant knowledge tools is successfully obtained remains an open question for study [18][19] and most previous works thus decide to stop here [20][21]. We strengthen this realistic approach by incorporating such rules, such as filtering the stopwords, soft matching, and noise reduction, by reducing its value by the use of attention mechanisms.

3.2. Schema Graph Construction

3.2.1. ConceptNet. We want to present briefly our goal information map ConceptNet before plunging into schema graph development. ConceptNet is designed to offer realistic context- based assumptions regarding texts from the real world. It is perhaps due to the fact that its representation of the language in itself is semi-structured English [22]. ConceptNet can be viewed as a broad set of three forms (*h*, *t*, *r*), such as (*pen, write, hasproperties*), in which *h* and *t* represent head and tail concepts respectively belonging to the concept set *Vg* whereas *r* represents a certain type of relationship belonging to the predefined set E. To enhance the properties of the Knowledge Graph for grounding and simulation, we removed and merged the initial 42 relational types into 17 types.

3.2.2. Sub-graph Matching via Path Finding. A scheme diagram is defined as a sub-graph *sg* of the complete knowledge graph *G*, which portrays the knowledge for the questionnaire pair at task keeping additional concepts and edges at minimum. We ultimately want to find a minimum subgraph covering the complete "*Steiner Tree Problem*" in the graph for all questions and answers [23]. We consider that it is difficult to obtain a detailed yet useful collection of information facts because of the incompleteness and enormous scale of ConceptNet. Therefore, by finding the way among listed concepts, we offer a simple but effective graphics construction algorithm ($CN_{qp} \cup CN_{ap}$).

Basically, we will identify routes between them which are shorter than k concepts for each query concept cn_x and response concept cn_y where $cn_x \in CN_{qp}$ and $c_y \in CN_{ap}$. Then, between the concept pairs in CN_{qp} or CN_{ap} we add edges if any.

3.3. Pruning paths using Knowledge Graph Embeddings

We first use Knowledge Graph Embedding (KGE) techniques, like TransE [24] to prune non- relevant paths from noisy schema graphs which may be potentially important. Using TransE we pre-train concept embeddings Vg and return relation embeddings R, which will be used to initialize our model. We decompose the path into a set of triples and measure them using a scoring function (confidence) of the KGE method in order to analyze the quality of them. We ultimately use the multiplication of these scores of each triple to calculate the final scores and set a cutoff threshold for pruning.

4. Knowledge Aware Graph Network

This is the main component which is required for the reasoning is our framework. Firstly, the schema graphs are encoded with graph convolutional networks to incorporate the concept embedding obtained previously, particularly the context related to the schema graph. Then the LSTMs are used to encode

the path between CN_{qp} and CN_{ap} (the question concept and answer concept respectively). Lastly, a Hierarchical path-based attention mechanism is used to complete the GCN-LSTM-HPA architecture. This models the relational schema graph according to the path between the CN_{qp} and CN_{ap}.

4.1. Graph Convolutional Network

Graph Convolutional Networks (GCN) [25] are a means of a scalable approach for problems involving data structured as a graph in a semi-supervised learning environment. The GCNs involve an efficient variant of convolutional neural networks which can be operated directly on the graphs.

The concepts representations obtained previously must now be transformed into their specific graph context. The concept vector is updated according to their neighbors using GCNs to prevent ambiguity and make the embedding context-specific. For instance, the word "*book*" can be inferred either as a noun in the context "*She reads a book*", or a verb in the context "*Book the tickets*". Additionally, the schema graph provides valuable information for reasoning.

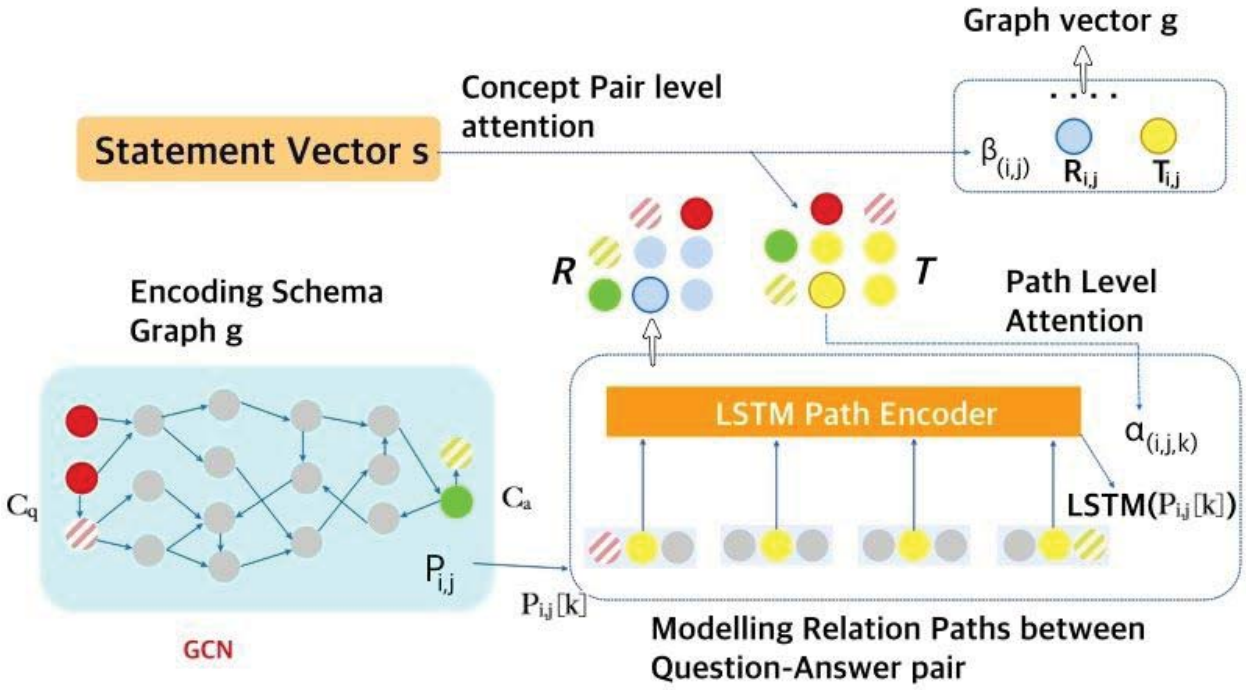

Figure 3. An overview of the GCN-LSTM-HPA used as our framework

Relational Graph Convolutional Networks (R-GCN) [26] is a recent work describing a class of neural networks operating on graphs, specifically dealing with multi-relational data. Although R-GCNs might seem promising, recent work by Marcheggiani [27] shows that these networks usually over parameterize the model. So, we cannot utilize this model effectively in case of multi-hop

relational data. Ultimately, we have used the vanilla version of the schema graphs, not considering edge relations. We will follow the layer-wise propagation rule for the multi-layer graph GCN [28] [25]. The concept vector $cn_x \in V g_{sg}$ is configured by their first ($h_x^{(0)} = V g_x$) pre-trained embeddings in the schema graph *sg* in particular. The $(l+1)$-th layer can then be updated by pooling features of their own nodes at l-th layer and neighboring nodes (N_x) with a non-linear activation function σ, equation(1):

$$h^{(l+1)} = \sigma(W_{self}^{(l)} h_x^{(l)} + \sum_{y \in N_y} \frac{1}{|N_x|} W^{(l)} h_y^{(l)}) \quad (1)$$

$$h_x^{(0)} = V_x \quad (2)$$

where $W_{(l)}$ is a layer-specific weight matrix.

4.2. Relation Path Encoder

The relational information of the schema graph has to be recorded. We have used an LSTM-based path encoder over the GCN outputs in our framework. As our goal is to find the probability of an option to the given question, we represent the graphs in reference to CN_{qp}, CN_{ap} ,and the relation path between them.

The relation path is represented in the form of entity-relation triplets $< concept1, concept2, relation >$ [29]. So each path can be collectively represented as:

$$P_{x,y}[k] = \left[(cn_x^{qp}, n_0, r_0), (n_0, n_1, r_1), \ldots, \left(n_{n-1}, cn_y^{ap}, r_n\right) \right] \quad (3)$$

$$cn_x^{qp} \in CN_{qp} \qquad cn_y^{ap} \in CN_{ap} \quad 1 \le y \le N_x \quad (4)$$

Here x represents the xth question and y represents the yth answer option of the N_x ,available options for the particular question. Here the concept vectors are the previously obtained GCN layer outputs ($h_{(l)}$). Now, for our model, we have used LSTM networks [30] for encoding the paths into our required sequence of the vector triples [29]. We now calculate the latent relation $R_{x,y}$, between $cn_{(qp)}$, and $cn_{(ap)}$ by using the concatenated vector of first and last hidden states and aggregating the representations of all paths between them in the schema graph [28]:

$$R_{x,y} = \frac{1}{|P_{x,y}|} \sum_k LSTM(P_{x,y}[k]) \quad (5)$$

We now use mean pooling on all vectors of R to produce a final vector representation of the schema graph:

$$T_{x,y} = MLP([s; cn_{qp}^{(x)}; cn_{ap}^{(x)}]) \quad (6)$$

$$sg = \frac{\sum_{xy} [R_{x,y}; T_{x,y}]}{|CN_{qp}| \times |CN_{ap}|} \quad (7)$$

where $[.;.]$ means concatenation of two vectors [28] and s is the statement vector obtained from language encoder. We now create a special token ("*question + [sep] + answer*") to combine the question and answer and then use a vector '[*cls*]' [7]. The $T_{x,y}$ inspired by the Relation Vector [31] is concatenated with $R_{x,y}$ for executing the average pooling. Ultimately, the probability score of the solution choice of a specific question *qp* can be determined as [28] :

$$probability\_score(qp,\ ap)\ =\ sigmoid(MLP(sg)) \tag{8}$$

4.3. Hierarchical Attention Mechanism

Now, there arises an argument that means pooling over path vectors does not always satisfy our requirement as some path may be more important than the rest. Also, we may come across some question-answer pairs which are not relevant or important for our purpose. In order to selectively choose important path vectors and useful question-answer pairs, we have proposed a hierarchical path-based attention mechanism [32] [33]. We have used path-level and concept-pair-level attention for contextual modelling of the graph [28]. We will calculate the importance of $\widehat{\alpha}_{(x,y,.)}$ for the path using a parametric matrix W_1 for path-level attention scores.

$$\alpha_{(x,y,k)} = T_{x,y} W_1 LSTM(P_{x,y}[k]), \tag{9}$$

$$\widehat{\alpha}_{(x,y,.)} = SoftMax(\alpha_{(x,y,.)}), \tag{10}$$

$$\widehat{R}_{x,y} = \sum_k \widehat{\alpha}_{(xyj,k)} LSTM(P_{x,y}[k]) \tag{11}$$

In the same way, we see the attention scores over the concept-pairs

$$\beta_{x,y)} = sW_2 T_{x,y} \tag{12}$$

$$\widehat{\beta}_{(.,.)} = SoftMax(\beta_{(.,.)}) \tag{13}$$

$$g = \sum_{x,y} \widehat{\beta}_{(x,y)} [\widehat{R}_{x,y}; T_{x,y}] \tag{14}$$

Now we coin the architecture we have used as GCN-LSTM-HPA architecture. This architecture models relational reasoning graphs under the influence of both the symbolic space of knowledge and the semantic space of language.

5. Experiments

We will now walk through the setup using the CommonsenseQA dataset [7] and present vanilla methods along with analyzing the results.

Table 1. Accuracy Comparison Table

Model	10(%) of Mtrain		50(%) of Mtrain		100(%) of Mtrain	
	Mdev-Auc.(%)	Mtest-Auc.(%)	Mdev-Auc.(%)	Mtest-Auc.(%)	Mdev-Auc.(%)	Mtest-Auc.(%)
Arbitrary Guess	20.0	20.0	20.0	20.0	20.0	20.0
Bert-Base-Fine Tuning	31.02	29.01	38.39	37.41	51.25	51.02
Bert-Base-**Our Model**	31.02	**30.54**	39.83	**39.05**	54.75	**56.03**
GPT Fine-Tuning	21.08	27.03	32.31	30.87	47.05	45.72
GPT - **Our Model**	27.65	**26.37**	33.84	**31.98**	49.12	**46.17**
Human Performance	-	87.5	-	87.5	-	87.5

5.1. CommonsenseQA dataset and setup

A total of 12,102 (v1.11) natural-language questions that require an ability to answer human common sense reasoning when every question has 5 (hard mode) answers. The researchers also publish a simple version of the dataset by choosing two random words for the health check. CommonsenseQA is specifically obtained from real human annotators and covers a wide variety of common sense forms, including geographical, personal, contextual, physiological, temporal, etc. CommonsenseQA can be the best way for us to test guided models of training and answer questions in line with our best knowledge.

We will use the official split (9,700/1,242/1,160) called (OStrain/OSdev/OStest) to compare the results posted on CommonsenseQA paper and leaderboard. Please note that only predictions to organizers can be tested for OStest performance. We have chosen the randomly chosen 1 221 examples from the training data for our in-house information to test other baseline methods and ablation studies randomly, and have formed the (Mtrain/Mdev/Mtest) dividing (8500/1,241/1,221). As the authors suggest, both tests use random groups, so three or more random states are checked to find the best design sets.

5.2. Comparative Analysis

Two reference approaches are known as follows :

5.2.1. Knowledge-agnostic Approach. Such approaches use either no external resources or are only used as additional information for unstructured text bodies such as the compilation of document samples by search engines or major language models as BERT. QABILINEAR, QACOMPARE, ESIM are three natural language inference controlled models, which can be fitted with different word embedding frameworks like GloVe and ELMO. BIDAF++ uses Google's web snippets as context while using ELMO as input function, with an extra layer of self-attention. GPT / BERT has finely tuned approaches with a further linear layer to define as indicated by the researchers. They also apply a '[*sep*]' special token to the data and use the '[*cls*]' secret state as static layer information. More details can be found on them in the database report [7].

5.2.2. Knowledge-aware Approach. We also used a few methods recently suggested to include knowledge graphs for answering questions. [34] proposes to collect human explanations from annotators for common reasoning as additional knowledge and then to train a language model for improvement of model performance based on such human annotations.

Table 2. Benchmark Comparison

Model	OSdev-Auc.(%)	OStest-Auc.(%)
Arbitrary Guess	20.0	20.0
BIDAF++	-	31.6
Esim + Elmo	-	32.0
Esim + Glove	-	30.9
QACompare + Glove	-	25.1
QABLinear + Glove	-	30.8
Cos-E	-	57.79
Bert-Base-FineTuning	51.42	51.13
GPT - Fine Tuning	46.79	45.74
Our Model	63.16	57.2
Human Performance	-	87.5

5.3. Implementation of our Model

We have two GCN layers (100 dim, 50 dim), and one (128dimm) Bidirectional LSTMs in our latest (OSdev) configurations. We pre-train KGE with GloVe embedding using TransE (100 dimensions). The statement encoder is BERT that works for each pair of questions and answers as a pre-trained sentence encoder. The paths have a threshold of 0.15, which holds 67.21% of the initial paths. We have not taken fewer than three directions in reproduction pairs. It is a randomly sampled vector for very few pairs without any path. Our Adam Optimizers show us our model designs. We also observed that ConceptNet's confirmation of common-sense questions and responses is very strong in our studies (over 98% of QA pairs have several established concepts).

5.4. Comparisons and Analysis of Performance

5.4.1. Standard Baseline Comparison We have used the official split to compare our model with the existing baseline methods reported till date. The results are shown in Table 2. Pre-training approaches based on BERT and GPT are far higher, showing the capacity of language models to store common sense information in an implicit manner than other baseline approaches. Trinh and Le [35] and Wang et. al. [21] also investigated and showed that these assumptions work. We have obtained a state-of-art-performance with a 2.2% increment in accuracy.

We have executed our experiments on different fractions of the dataset (10%, 50%,and 100% of the training data). The results obtained are described in Table 1. We further observe that improvements in small data (10%) scenarios are very limited as compared to the rest.

5.4.2. Error Analysis There are some failed cases where our model is not good at:

• Negative Reasoning: Sensitivity to negative words is not observed in the grounding state and resulting in choosing opposite answers to the correct ones.

• Comparative Reasoning: For answers having more than one probable answer. Albeit, the commonsense reasoning model should discern different answer possibilities, our model cannot do this.

5.5. Interpretability Case Study

Their system has the advantage of being straightforward, thus rendering the deduction method more interpretable. Through studying the centralized emphasis on question-answer pairs and the interaction between the two, we will explain their prototype behaviors. We select the key pairs which have the highest concentrations, then scan the highest pair paths. We can observe that the paths in this way are very closely related to the inferencing approach, which minimizes cracking concepts like “waterfall” by modeling.

5.6. Model Transferability

By explicitly checking it by setting its parameters for another task, we analyze the transferability of a model trained on CommonsenseQA (CSQA). We test it on SWAG [4] and WSC [36] data sets to determine their transferability. The 20k testing samples in SWAG are first tested.

5.7. Recent leaderboard methods

We argue that our model uses ConceptNet as its only external resource. Other methods are used to improve orthogonal performance; the most recent submissions with public information on the leaderboard (as of Novemeber 2019) are using larger additional text data and fine-tuning on larger pre-training encoders like XLNet [37], RoBERTa[38]. Even algorithms are used to pass information from other read-understanding datasets, such as RACE [39] and OpenBookQA [40].

The interesting fact is that the initial RoBERTa fine-tuned system, pre-trained with corpora much greater than BERT, is still the best performance on the OStest Collection. We also use statements vectors of Roberta for the input of our model and note that OSdevs performance marginally improves from 77.47% to 77.56%. All other Roberta extended methods have negative improvements. We think that RoBERTas fine-tuning has reached the limit because of our aforementioned failed case analyses in the data set and the absence of comparative reasoning strategies.

6. Related Works

6.1. Commonsense Knowledge Reasoning

Machine commonsense learning has gained wide attention and there has been a recent demand for large scale novel datasets for testing them on various focuses and disciplines like social behavior understanding [41], situation prediction (SWAG) [4] and commonsense reasoning in general [7]. These works encourage the study of learning technique methods for commonsense reasoning. Works like that of Trinh and Le [35] shows that WSC resolution [36] shows promising results for broad language models, but these are specific domain-driven and hardly can be applied for a generic scenario. An advantage of our framework is that it uses grounded triples and paths, enabling better behaviors and inferences from the model.

6.2. External knowledge for Understanding

A major component of our work depends on using external knowledge to answer questions and encode sentences. The first ones to suggest encrypting sentences include Yang and Mitchell [42], who keep related entities from knowledge bases and fusion in the LSTM calculations to boost their efficiency. Annervaz [29], Weissenborn et. al. [18], Mihaylov, and Frank [43] follow this sequence of work to use the related knowledge embeddings triples to improve the performance of understanding and reasoning tasks.

Some recent methods using ConceptNets by Zhong et. al. [20] and Wang et. al. [21] have been adopted in our experiments. As an explicit graph structure, our framework relies on the use of external information and provides contextual interpretation over the graphs. Rajani et. al. [34] propose to collect human data to provide explanations for correct answers as an additional feature for auxiliary learning.

6.3. Relational Reasoning

Relation Network Module (RN) [31] is proposed for visual question answering tasks. In order to view the concepts as objects and answers in questions and responses, and use external knowledge graphs to form connections between semantic and symbolic spaces, our framework can be seen as a kind of knowledge supplemented RN.

7. Conclusion

In this work, we have demonstrated the need to embody worldly knowledge in training models. In order to address commonsense questions, we propose a knowledge-compatible framework. The framework builds schema graphs for the appropriate broad consensus first and then shapes the graphs using our model. The system utilizes a design of GCN-LSTM-HPA, which conveniently shows objects to be interpreted and converted for functional purposes and provides a new state-of-the-art data package for the evaluation of machine commonsense. This is an effort to instill general world knowledge in order to study fundamental thinking styles.

There is plenty of room for expansion as the first work of its kind. A more advanced model can be developed that can better obtain facts from millions of entries. In future directions, the parsing of questions will be better for dealing with negation and comparative questions, as well as for integrating knowledge in visual reasoning.

8. References

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

[2] Drew A Hudson and Christopher D Manning. 2018 Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067.*

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

[4] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018 Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

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

[6] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019 From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6720–6731.

[7] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018 Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008 Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data,* pages 1247–1250. AcM.

[9] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990 Introduction to wordnet: An on-line lexical database. *International journal of lexicography,* **3(4)**:235–244.

[10] Yangqiu Song and Dan Roth. 2017 Machine learning with world knowledge: the position and survey. *arXiv preprint arXiv:1705.02908*.

[11] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015 A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, **104(1)**:11–33.

[12] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018 Improving language understanding by generative pre-training. URL *https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.*

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

[14] Robert Speer, Joshua Chin, and Catherine Havasi. 2017 Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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

[16] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017 Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

[17] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018 Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

[18] Dirk Weissenborn, Tomáš Kočisky`, and Chris Dyer. 2017 Dynamic integration of background knowledge in neural nlu systems. *arXiv preprint arXiv:1706.02596*.

[19] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2017 Learning what is essential in questions. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017),* pages 80–89.

[20] Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019 Improving question answering by commonsense-based pre-training. In *CCF International Conference on Natural Language Processing and Chinese Computing,* pages 16–28. Springer.

[21] Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. 2019 Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI Conference on Artificial Intelligence,* volume **33**, pages 7208–7215.

[22] Hugo Liu and Push Singh. 2004 Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal,* **22(4)**:211–226.

[23] 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.

[24] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014 Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

[25] Thomas N Kipf and Max Welling. 2016 Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
[26] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018 Modeling relational data with graph convolutional networks. In *European Semantic Web Conference,* pages 593–607. Springer.
[27] Diego Marcheggiani and Ivan Titov. 2017 Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.
[28] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019 Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.
[29] KM Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018 Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930*.
[30] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012 Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.
[31] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017 A simple neural network module for relational reasoning. In *Advances in neural information processing systems, pages 4967–4976*.
[32] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014 Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*
[33] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016 Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies,* pages 1480–1489.
[34] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019 Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.
[35] Trieu H Trinh and Quoc V Le. 2018 A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847.*
[36] Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012 The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.*
[37] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019 Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237.*
[38] 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. *arXiv preprint arXiv:1907.11692.*
[39] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017 Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683.*
[40] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018 Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789.*
[41] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019 Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728.*
[42] Bishan Yang and Tom Mitchell. 2019 Leveraging knowledge bases in lstms for improving machine reading. *arXiv preprint arXiv:1902.09091.*
[43] Todor Mihaylov and Anette Frank. 2018 Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. *arXiv preprint arXiv:1805.07858.*