Towards Generalizable Neuro-Symbolic Systems for Commonsense Question Answering

Kaixin Ma[¶][*]   **Jonathan Francis**[¶][§]   **Quanyang Lu**[†]   **Eric Nyberg**[¶]   **Alessandro Oltramari**[§]

[¶]Language Technologies Institute, School of Computer Science, Carnegie Mellon University
[†]Department of Mechanical Engineering, College of Engineering, Carnegie Mellon University
[§]Intelligent IoT, Bosch Research and Technology Center (Pittsburgh, USA)
{kaixinm, jmf1, qlv, ehn}@cs.cmu.edu
alessandro.oltramari@us.bosch.com

Abstract

Non-extractive commonsense QA remains a challenging AI task, as it requires systems to reason about, synthesize, and gather disparate pieces of information, in order to generate responses to queries. Recent approaches on such tasks show increased performance, only when models are either pre-trained with additional information or when domain-specific heuristics are used, without any special consideration regarding the knowledge resource type. In this paper, we perform a survey of recent commonsense QA methods and we provide a systematic analysis of popular knowledge resources and knowledge-integration methods, across benchmarks from multiple commonsense datasets. Our results and analysis show that attention-based injection seems to be a preferable choice for knowledge integration and that the degree of domain overlap, between knowledge bases and datasets, plays a crucial role in determining model success.

1 Introduction

With the recent success of large pre-trained language models (Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019; Liu et al., 2019), model performance has reached or surpassed human-level capability on many previous question-answering (QA) benchmarks (Hermann et al., 2015; Rajpurkar et al., 2016; Lai et al., 2017). However, these benchmarks do not directly challenge model reasoning capability, as they require only marginal use of external knowledge to select the correct answer, i.e., all the evidence required to solve questions in these benchmarks is explicit in the context lexical space. Efforts have been made towards building more challenging datasets that, by design, require models to synthesize external commonsense

knowledge and leverage more sophisticated reasoning mechanisms (Zhang et al., 2018; Ostermann et al., 2018), showing that the previous state-of-the-art models often struggle to solve these newer tasks reliably. As a result, commonsense has received a lot of attention in other areas as well, such as natural language inference (Zellers et al., 2018b, 2019) and visual question answering (Zellers et al., 2018a). Despite the importance of commonsense knowledge, however, previous work on QA methods takes a coarse-grained view of commonsense, without considering the subtle differences across the various knowledge types and resources. Such differences have been discussed at length in AI by philosophers, computational linguists, cognitive psychologists (see for instance (Davis, 2014)): at the high level, we can identify *declarative commonsense*, whose scope encompasses factual knowledge, e.g., 'the sky is blue', 'Paris is in France'; *taxonomic knowledge*, e.g., 'football players are athletes', 'cats are mammals'; *relational knowledge*, e.g., 'the nose is part of the skull', 'handwriting requires a hand and a writing instrument'; *procedural commonsense*, which includes prescriptive knowledge, e.g., 'one needs an oven before baking cakes', 'the electricity should be off while the switch is being repaired' (Hobbs et al., 1987); *sentiment knowledge*, e.g., 'rushing to the hospital makes people worried', 'being in vacation makes people relaxed'; and *metaphorical knowledge* (e.g., 'time flies', 'raining cats and dogs'). We believe that it is important to identifiy the most appropriate commonsense knowledge type required for specific tasks, in order to get better downstream performance. Once the knowledge type is identified, we can then select the appropriate knowledge-base(s), and the suitable neural integration mechanisms (e.g., attention-based injection, pre-training, or auxiliary training objectives).

---

[*]Work was done during an internship at Bosch Research.

Accordingly, in this work we conduct a comparison study of different knowledge bases and knowledge integration methods, and we evaluate model performance on two multiple-choice QA datasets that explicitly require commonsense reasoning. In particular, we used ConceptNet (Speer et al., 2016) and the recently-introduced ATOMIC (Sap et al., 2019) knowledge resources, integrating them with the *Option Comparison Network* model (OCN; Ran et al. (2019)), a recent state-of-the-art model for multiple choice QA tasks. We evaluate our models on the DREAM (Sun et al., 2019) and CommonsenseQA (Talmor et al., 2019) datasets. An example from DREAM that requires commonsense is shown in Table 1, and an example from CommonsenseQA is shown in Table 2. Our experimental results and analysis suggest that attention-based injection is preferable for knowledge integration and that the degree of domain overlap, between knowledge-base and dataset, is vital to model success.[1]

Dialogue:
M: I hear you drive a long way to work every day.
W: Oh, yes. it's about sixty miles. but it doesn't seem that far, the road is not bad, and there's not much traffic.
Question:
How does the woman feel about driving to work?
Answer choices:
A. She doesn't mind it as the road conditions are good.*
B. She is unhappy to drive such a long way everyday.
C. She is tired of driving in heavy traffic.

Table 1: An example from the DREAM dataset; the asterisk (*) denotes the correct answer.

Question:
A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?
Answer choices:
A. Bank*; B. Library; C. Department Store;
D. Mall; E. New York

Table 2: An example from the CommonsenseQA dataset; the asterisk (*) denotes the correct answer.

2 Related Work

It has been recognized that many recent QA tasks require external knowledge or commonsense to solve, and numerous efforts have been made in injecting commonsense in neural models. Bauer et al. (2018) introduced a pipeline for extracting grounded multi-hop commonsense relation paths from ConceptNet and proposed to inject commonsense knowledge into neural models' intermediate representations, using attention. Similarly, Mihaylov and Frank (2018) also proposed to extract relevant knowledge triples from ConceptNet and use Key-Value Retrieval (Miller et al., 2016) to gather information from knowledge to enhance the neural representation. Zhong et al. (2018) proposed to pre-train a scoring function using knowledge triples from ConceptNet, to model the direct and indirect relation between concepts. This scoring function was then fused with QA models to make the final prediction. Pan et al. (2019a) introduced an entity discovery and linking system to identify the most salient entities in the question and answer-options. Wikipedia abstracts of these entities are then extracted and appended to the reference documents to provide additional information. Weissenborn et al. (2018) proposed a strategy of dynamically refining word embeddings by reading input text as well as external knowledge, such as ConceptNet and Wikipedia abstracts. More recently, Lin et al. (2019) proposed to extract subgraphs from ConceptNet and embed the knowledge using Graph Convolutional Networks (Kipf and Welling, 2016). Then the knowledge representation is integrated with word representation through an LSTM layer and hierarchical attention mechnism. Lv et al. (2019) introduced graph-based reasoning modules that takes both ConceptNet knowledge triples and Wikipedia text as inputs to refine word representations from a pretrained language model and make predictions.

Commonsense knowledge integration has also received a lot of attention on many other tasks. Tandon et al. (2018) proposed to use commonsense knowledge as hard/soft constraints to bias the neural model's prediction on a procedural text comprehension task. Ma et al. (2018) proposed to used embedded affective commonsense knowledge inside LSTM cell to control the information flow in each gate for sentiment analysis task. Li and Srikumar (2019) presented a framework to convert declarative knowlegde into first-order logic that enhance neural networks' training and prediction. Peters et al. (2019) and Levine et al. (2019) both tried to injecting knowlegde into language models by pretraining on knowledge bases.

Previous works only focus on using external

---

[1] From a terminological standpoint, 'domain overlap' here must be interpreted as the overlap between question types in the targeted datasets, and types of commonsense represented in the knowledge bases under consideration.

knowledge sources to improve model performance on certain tasks, disregarding the type of commonsense knowledge and how the domain of the knowledge resource affects results on downstream tasks. In this paper, we examine the roles of knowledge-base domain and specific integration mechanisms on model performance.

3 Approach Overview

In this section, we describe the model architecture used in our experiments. Next, we introduce two popular knowledge resources, we define our knowledge-extraction method, then we illustrate various neural knowledge-integration mechanisms.

3.1 Model architecture

The BERT model (Devlin et al., 2019) has been applied to numerous QA tasks and has achieved very promising performance, particularly on the DREAM and CommonsenseQA datasets. When utilizing BERT on multiple-choice QA tasks, the standard approach is to concatenate the dialogue context and the question with each answer-option, in order to generate a list of tokens which is then fed into BERT encoder; a linear layer is added on top, in order to predict the answer. One aspect of this strategy is that each answer-option is encoded independently: from a cognitive perspective, this aspect contradicts how humans typically solve multiple-choice QA tasks, namely by *weighing* each option to find correlations within them, in addition to correlations with respect to the question. To address this issue, Ran et al. (2019) introduced the *Option Comparison Network* (OCN) that explicitly models pairwise answer-option interactions, making OCN better-suited for multiple-choice QA task structures. We re-implemented OCN while keeping BERT as its upstream encoder.[2] Specifically, given a dialogue D, a question Q, and an answer-option O_k, we concatenate them and encode with BERT to get hidden representation $T_{enc} \in \mathbb{R}^{n \times d}$:

$$T_{enc} = \text{BERT}(D; Q; O_k) \quad (1)$$

Where d is the size of BERT's hidden representation and n is the total number of words. Next,

the dialogue encoding $D_{enc} \in \mathbb{R}^{n_d \times d}$, question encoding $Q_{enc} \in \mathbb{R}^{n_q \times d}$, and answer-option encoding $O_{k,enc} \in \mathbb{R}^{n_o \times d}$ are separated from T_{enc}. Here, option-encoding consists both of question and option, i.e. $Q_{enc} \subseteq O_{k,enc}$ and $n_d + n_o = n$, as suggested by Ran et al. (2019). Given a set of options O_k ($k = 1, 2, ...$), these options are compared, pairwise, using standard tri-linear attention (Seo et al., 2016):

$$\text{Att}(u, v) = W_1 \cdot u + W_2 \cdot v + (W_3 \circ v) \cdot u \quad (2)$$

Where, $W_1, W_2, W_3 \in \mathbb{R}^d$ are trainable weights and $u \in \mathbb{R}^{x \times d}$, $v \in \mathbb{R}^{y \times d}$ are input matrices; x and y here are generic placeholder for input lengths; matrix multiplication and element-wise multiplication are denoted by ($\cdot$) and ($\circ$), respectively. Next, we gather information from all other options, to form a new option representation $O_{k,new} \in \mathbb{R}^{n_o \times d}$. Formally, given option $O_{k,enc}$ and another option $O_{l,enc} \in \mathbb{R}^{n_l \times d}$, $O_{k,new}$ is computed as follows:

$$O_k^l = O_{l,enc} \cdot \text{Att}(O_{l,enc}, O_{k,enc}) \quad (3)$$

$$\widetilde{O_k^l} = [O_{k,enc} - O_k^l; O_{k,enc} \circ O_k^l] \quad (4)$$

$$O_{k,new} = \tanh(W_c \cdot [O_{k,enc}; \{\widetilde{O_k^l}\}_{l \neq k}]) \quad (5)$$

Where, $W_c \in \mathbb{R}^{(d+2d(|O|-1)) \times d}$, $|O|$ denotes total number of options and n_l denotes the number of words in the compared option. Then, a gating mechanism is used to fuse the option-wise correlation information $O_{k,new}$ with the current option-encoding $O_{k,enc}$. Gating values are computed as:

$$G = \text{sigmoid}(W_g[O_{k,enc}; O_{k,new}; \widetilde{Q}]) \quad (6)$$

$$\widetilde{Q} = Q_{enc} \cdot \text{softmax}(Q_{enc} \cdot V_a)^T \quad (7)$$

$$O_{fuse} = G \circ O_{k,enc} + (1 - G) \circ O_{k,new} \quad (8)$$

Here, $W_g \in \mathbb{R}^{3d \times d}$ and $V_a \in \mathbb{R}^{d \times 1}$. Co-attention (Xiong et al., 2016) is applied to re-read the dialogue, given the fused option-correlation features:

$$A_{do} = \text{Att}(D_{enc}, O_{fuse}) \quad (9)$$

$$A_{od} = \text{Att}(O_{fuse}, D_{enc}) \quad (10)$$

$$O_d = A_{od} \cdot [D_{enc}; A_{do} \cdot O_{fuse}] \quad (11)$$

$$\widetilde{O_d} = \text{ReLU}(W_p([O_d; O_{fuse}])) \quad (12)$$

Here, $W_p \in \mathbb{R}^{3d \times d}$. Finally, self-attention (Wang et al., 2017) is used to compute final option repre-

---

[2]Because the newly-released XLNet has out-performed BERT on various tasks, we considered using XLNet as the OCN's encoder. However, from our initial experiments, XLNet is very unstable, in that it easily provides degenerate solutions—a problem noted by Devlin et al. (2019) for small datasets. We found BERT to be more stable in our study.
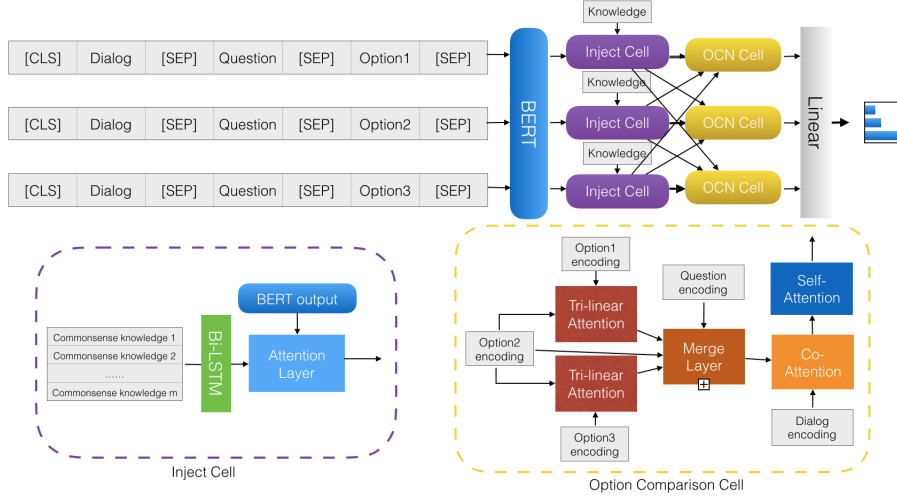
Figure 1: Option Comparison Network with Knowledge Injection

sentation $\widetilde{O_f} \in \mathbb{R}^{n_o \times d}$:

$$O_s = \widetilde{O_d} \cdot \text{Att}(\widetilde{O_d}, \widetilde{O_d}) \tag{13}$$

$$O_f = [\widetilde{O_d}; O_s, \widetilde{O_d} - O_s; \widetilde{O_d} \circ O_s] \tag{14}$$

$$\widetilde{O_f} = \text{ReLU}(W_f \cdot O_f) \tag{15}$$

Unlike the vanilla BERT model, which takes the first token to predict the answer, max-pooling is applied on the sequence dimension of $\widetilde{O_f} \in \mathbb{R}^{n_o \times d}$, in order to generate the final prediction.

3.2 Knowledge bases

The first knowledge-base we consider for our experiments is ConceptNet (Speer et al., 2016). ConceptNet contains over 21 million edges and 8 million nodes (1.5 million nodes in the partition for the English vocabulary), generating triples of the form $(C1, r, C2)$: the natural-language concepts $C1$ and $C2$ are associated by commonsense relation r, e.g., *(dinner, AtLocation, restaurant)*. Thanks to its coverage, ConceptNet is one of the most popular semantic networks for commonsense. ATOMIC (Sap et al., 2019) is a new knowledge-base that focuses on procedural knowledge. Triples are of the form *(Event, r, {Effect|Persona|Mental-state})*, where head and tail are short sentences or verb phrases and r represents an *if-then* relation type. An example would be: *(X compliments Y, xIntent, X wants to be nice)*. Since both DREAM and CommonsenseQA datasets are open-domain and require general commonsense, we think these knowledge-bases are most appropriate for our investigation.

3.3 Knowledge elicitation

ConceptNet. For the DREAM dataset, we find ConceptNet relations that connect dialogues and questions to the answer-options. The intuition is that these relation paths would provide explicit evidence that would help the model find the answer. Formally, given a dialogue D, a question Q, and an answer-option O, we find all ConceptNet relations *(C1, r, C2)*, such that $C1 \in (D + Q)$ and $C2 \in O$, or vice versa. This rule works well for single-word concepts. However, a large number of concepts in ConceptNet are actually phrases, and finding exactly matching phrases in $D/Q/O$ is much harder. To fully utilize phrase-based ConceptNet relations, we relaxed the exact-match constraint to the following:

$$\frac{\# \text{ words in C} \cap \text{S}}{\# \text{ words in C}} > 0.5 \tag{16}$$

Here, S represents $D/Q/O$, depending on which sequence we try to match the concept C to. Additionally, when the part-of-speech (POS) tag for a concept is available, we make sure it matches the POS tag of the corresponding word in $D/Q/O$. For CommonsenseQA, we use the same procedure to find ConceptNet relations for each answer-option, except that only Q is present and used. Table 3 shows the extracted ConceptNet triples for the CommonsenseQA example in Table 2. It is worth noting that we are able to extract the original ConceptNet sub-graph that was used to create the question, along with some extra triples. Although not perfect, the bold ConceptNet triple does provide some clue that could help the model resolve the correct answer.

25

Options	Extracted ConceptNet triples
Bank	(revolving door *AtLocation* bank) **(bank RelatedTo security)**
Library	(revolving door *AtLocation* library)
Department Store	(revolving door *AtLocation* store) (security IsA department)
Mall	(revolving door *AtLocation* mall)
New York	(revolving door *AtLocation* New York)

Table 3: Extracted ConceptNet relations for sample shown in Table 2.

Input sentence	Generated ATOMIC relations
Utterance 1	(xAttr dedicated) (xWant to get to work)
Utterance 2	(xAttr far) **(xReact happy)** (xWant to get to their destination)
Option A	**(xAttr calm)** (xWant to avoid the road)
Option B	**(xAttr careless)** (xReact annoyed) (xEffect get tired)
Option C	**(xAttr frustrated)** (xEffect get tired) (xWant to get out of car)

Table 4: Sample generated ATOMIC relations for sample shown in Table 1.

ATOMIC. We observe that many questions in `DREAM` inquire about agent's opinion and feeling. Superficially, this particular question type seems well-suited for ATOMIC, whose focus is on folk psychology and related general implications; we could frame our goal as evaluating whether ATOMIC can provide relevant knowledge to help answer these questions. However, one challenge to this strategy is that heads and tails of knowledge triples in ATOMIC are short sentences or verb phrases, while rare words and person-references are reduced to blanks and PersonX/PersonY, respectively. This calls for a new matching procedure, different from the ConceptNet extraction strategy, for eliciting ATOMIC-specific relations: we rely on the recently-published COMET model (Bosselut et al., 2019) to generate new ATOMIC relations, with intermediate phrasal resolutions. In particular, we first segmented all dialogues, questions, and answer-options into sentences. We further segment long sentences into sub-sentences, using commas as seperators. Because only verb-phrases satisfy the definition of an "event" in ATOMIC (i.e., relations are only invoked by verbs), we remove all sentences/sub-sentences that do not contain any verb. Next, we use a pre-trained COMET model (Bosselut et al., 2019) to generate all possible ATOMIC relations, for all candidate sentences/sub-sentences and we use greedy-decoding to take the 1-best sequences. Table 4 shows the sample ATOMIC relations, generated using the `DREAM` example in Table 1. It is interesting to note that the reaction for the woman agent (second utterance) is identified as *happy*, since she said that 'the road is not bad.' If we compare the identified attributes for answer-options,

the one from correct answer seems to be sematically closer than the other two.

3.4 Knowledge injection

Given previously extracted/generated knowledge triples, we need to integrate them with the OCN model. Inspired by Bauer et al. (2018), we propose to use attention-based injection. For Concept-Net knowledge triples, we first convert concept-relation tokens into regular tokens, in order to generate a pseudo-sentence. For example, "*(book, At-Location, library)*" would be converted to "book at location library." Next, we use the BERT embedding layer to generate an embedding of this pseudo-sentence, with C denoting a ConceptNet relation:

$$H_C = \text{BiLSTM}(C) \qquad (17)$$

If we let $H_C \in \mathbb{R}^{1 \times 2l}$ be the concatenation of the final hidden states and l be the number of hidden units in the LSTM layer, then m ConceptNet relations would yield the commonsense knowledge matrix $H_M \in \mathbb{R}^{m \times 2l}$. We adopt the attention mechanism used in QAnet (Yu et al., 2018) to model the interaction between H_M and the BERT encoding output T_{enc} (from Equation 1):

$$\widetilde{H}_M = H_M \cdot W_{proj} \qquad (18)$$
$$\mathcal{S} = \text{Att}(H_M, T_{enc}) \qquad (19)$$
$$A_m = \text{softmax}(\mathcal{S}) \cdot \widetilde{H}_M \qquad (20)$$
$$A_t = \text{softmax}(\mathcal{S}) \cdot \text{softmax}(\mathcal{S}^T) \cdot T_{enc} \qquad (21)$$
$$T_C = [T_{enc}; A_m; T_{enc} \circ A_m; T_{enc} \circ A_t] \qquad (22)$$
$$T_{out} = \text{ReLU}(T_C \cdot W_a) \qquad (23)$$

Specifically, H_M is first projected into the same dimension as T_{enc}, using $W_{proj} \in \mathbb{R}^{2l \times d}$. Then,

the similarty matrix $\mathcal{S} \in \mathbb{R}^{n \times m}$ is computed using tri-linear attention, as in Equation 2. We then use $\mathcal{S}$ to compute text-to-knowledge attention $A_m \in \mathbb{R}^{n \times d}$ and knowledge-to-text attention $A_t \in \mathbb{R}^{n \times d}$. Finally, the knowledge-aware textual representation $T_{out} \in \mathbb{R}^{n \times d}$ is computed, where $W_a \in \mathbb{R}^{4d \times d}$. T_{out} is fed to subsequent layers (in place of T_{enc}), in order to generate the prediction. The model structure with knowledge-injection is summarized in Figure 1.

For ATOMIC knowledge triples, the injection method is slightly different. Because heads of these knowledge triples are sentences/utterances and the tails contain attributes of the persons (i.e., subject and object of the sentence), it is not possible to directly inject the knowledge triples, as-is. We replace the heads of the ATOMIC knowledge triples with the corresponding speaker for dialogues and leave as blank for the answer-options. Next, we convert the special relation tokens into regular tokens, e.g., "xIntent"⇒"intent" and "oEffect"⇒ "others effect", to make pseudo-sentences. As a result, an ATOMIC relation "*(the road is not bad, xReact, happy)*" would be converted to "*(W, react, happy)*." Moreover, as the ATOMIC knowledge triples are associated with dialogues and answer-options, independently, we inject option relations into $O_{enc} \in \mathbb{R}^{n_o \times d}$ and dialogue relations into D_{enc}, respectively, using the injection method described above.

3.5 Knowledge pre-training

Pre-training large-capacity models (e.g., BERT, GPT (Radford et al., 2019), XLNet (Yang et al., 2019)) on large corpora, then fine-tuning on more domain-specific information, has led to performance improvements on various tasks. Inspired by this, our goal in this section is to observe the effect of pre-training BERT on commonsense knowledge and refining the model on task-specific content from our DREAM and CommonsenseQA corpora. Essentially, we would like to test if pre-training on our external knowledge resources can help the model acquire commonsense. For the ConceptNet pre-training procedure, pre-training BERT on pseudo-sentences formulated from ConceptNet knowledge triples does not provide much gain on performance. Instead, we trained BERT on the *Open Mind Common Sense* (OMCS) corpus (Singh et al., 2002), the original corpus that was used to create ConceptNet. We extracted about 930K English sentences from OMCS and

randomly masked out 15% of the tokens; we then fine-tuned BERT, using a masked language model objective. Then we load this fine-tuned model into OCN and trained on DREAM and CommonsenseQA tasks. As for pre-training on ATOMIC, we again use COMET to convert ATOMIC knowledge triples into sentences; we created special tokens for 9 types of relations as well as blanks. Next, we randomly masked out 15% of the tokens, only masking out tail-tokens. We use the same OMCS pre-training procedure.

Models	Dev Acc	Test Acc
BERT Large(*)	66.0	66.8
XLNet(*)	-	**72.0**
OCN	70.0	69.8
OCN + CN injection	**70.5**	69.6
OCN + AT injection	69.6	**70.1**
OCN + OMCS pre-train	64.0	62.6
OCN + ATOMIC pre-train	60.3	58.8

Table 5: Results on DREAM; the asterisk (*) denotes results taken from leaderboard.

Models	Dev Acc
BERT + OMCS pre-train(*)	68.8
RoBERTa + CSPT(*)	**76.2**
OCN	64.1
OCN + CN injection	67.3
OCN + OMCS pre-train	65.2
OCN + ATOMIC pre-train	61.2
OCN + OMCS pre-train + CN inject	**69.0**

Table 6: Results on CommonsenseQA; the asterisk (*) denotes results taken from leaderboard.

4 Experiments

4.1 Datasets

We choose to evaluate our hypotheses using the DREAM and CommonsenseQA datasets, because some / all questions require commonsense reasoning and because there remains a large gap between state-of-the-art models and human performance.

DREAM is a dialogue-based multiple-choice QA dataset, introduced by Sun et al. (2019). It was collected from English-as-a-foreign-language examinations, designed by human experts. The dataset contains 10,197 questions for 6,444 dialogues in total, and each question is associated with 3 answer-options. The authors point out that 34% of questions require commonsense knowledge to answer, which includes social implication, speaker's intention, or general world knowledge.

CommonsenseQA is a multiple-choice QA dataset that specifically measure commonsense reasoning (Talmor et al., 2019). This dataset is constructed based on ConceptNet (Speer et al.,

2016). Specifically, a source concept is first extracted from ConceptNet, along with 3 target concepts that are connected to the source concept, i.e., a sub-graph. Crowd-workers are then asked to generate questions, using the source concept, such that only one of the target concepts can correctly answer the question. Additionally, 2 more distractor concepts are selected by crowd-workers so that each question is associated with 5 answer-options. In total, the dataset contains 12,247 questions. For `CommonsenseQA`, we evaluate models on the development-set only, since test-set answers are not publicly available.

4.2 Training details

For ease of comparison, we borrow hyperparameter settings from Pan et al. (2019b); we used the BERT Whole-Word Masking Uncased model (Devlin et al., 2018) for all experiments. For `DREAM` experiments, we used a max sequence-length of 512, batch-size of 24, learning rate of $1e^{-5}$, and we trained the model for 16 epochs. For `CommonsenseQA`, we used a max sequence length of 60, batch-size of 32, learning rate of $1e^{-5}$, and trained for 8 epochs. For pre-training on OMCS, we used max sequence length of 35, batch-size of 32, learning rate of $3e^{-5}$, and trained for 3 epochs. For pre-training on ATOMIC, the max sequence length is changed to 45, other hyperparameters remain the same, and we only use the ATOMIC training set. When using OCN on `CommonsenseQA`, since there is no dialogue, we compute co-attention with Q_{enc}, in place of D_{enc}, in order to keep the model structure consistent.

4.3 Results

`DREAM` results are shown in Table 5, and `CommonsenseQA` results are shown in Table 6. For all of our experiments, we run 3 trials with different random seeds and we report average scores in the tables. Evaluated on `DREAM`, our OCN model got a significant performance boost (+3.0%), compared to BERT-large from previous work. We think the reasons are that OCN is better-suited for the task and that we used BERT Whole-Word Masking Uncased model. OCN with ConceptNet knowledge-injection achieves slightly better results on the development-set, while ATOMIC knowledge-injection helps achieve a small improvement on the test-set. However, we recognize that these improvements are very limited; to our

surprise, OCN pre-trained on OMCS or ATOMIC got significantly lower performance.

As for results on `CommonsenseQA`, ConceptNet knowledge-injection provides a significant performance boost (+2.8%), compared to the OCN baseline, suggesting that explicit links from question to answer-options help the model find the correct answer. Pre-training on OMCS also provides a small performance boost to the OCN baseline. Since both ConceptNet knowledge-injection and OMCS pre-training are helpful, we combine both approaches with OCN and we are able to achieve further improvement (+4.9%). Finally, similar to the results on `DREAM`, OCN pre-trained on ATOMIC yields a siginificant performance drop.

5 Error Analysis

To better understand when a model performs better or worse with knowledge-integration, we analyzed model predictions. `DREAM` dataset provides annotations for about 1000 questions: 500 questions in the development-set and 500 in the test-set. Specifically, questions are manually classified into 5 categories: Matching, Summary, Logic inference, Commonsense inference, and Arithmetic inference; and each question can be classified under multiple categories. We refer readers to Sun et al. (2019) for additional category information. We extracted model predictions for these annotated questions in test-set and grouped them by types. The accuracies for each question-group are shown in Table 7. Note that we omitted 2 categories that have less than 10 questions. For the ConceptNet and the ATOMIC knowledge-injection models, we can see that they did better on questions that involve commonsense (last 3 columns in the table), and the performance on other types are about the same or slightly worse, compared to baseline OCN. As for models pre-trained on OMCS corpus or ATOMIC knowledge-base, we already saw that these model performances drop, compared to the baseline. When we look at the performance difference in each question type, it is clear that some categories account for the performance drop more than others. For example, for both the OMCS pre-trained model and the ATOMIC pre-trained model, performance drops significantly for Matching questions, in particular. On the other hand, for questions that require both commonsense inference and summarization, both models' performances only dropped

Models	M(54)	S(15)	A+L(11)	L(228)	C+L(122)	C(14)	C+S(60)
OCN	88.9	86.7	27.3	75.9	60.7	71.4	70.0
OCN + CN injection	83.3(-5.6)	86.7(+0.0)	18.2(-9.2)	76.8(+0.9)	59.8(-0.9)	64.3(-7.1)	78.3(**+8.3**)
OCN + AT injection	88.9(+0.0)	80.0(-6.7)	27.3(+0.0)	75.9(+0.0)	66.4(**+5.7**)	71.4(+0.0)	75(**+5.0**)
OCN + OMCS pre-train	70.4(-18.5)	73.3(-13.4)	45.4(+18.1)	69.7(-6.2)	48.4(-12.3)	57.1(-14.3)	68.3(**-1.7**)
OCN + ATOMIC pre-train	66.6(-22.3)	86.7(+0.0)	18.2(-9.2)	64.0(-11.9)	51.6(-9.1)	42.9(-28.5)	70.0(**+0.0**)

Table 7: Accuracies for each DREAM question type: **M** means *Matching*, **S** means *Summary*, **L** means *Logic inference*, **C** means *Commonsense inference*, and **A** means *Arithmatic inference*. Numbers beside types denote the number of questions of that type.

Models	AtLoc.(596)	Cau.(194)	Cap.(109)	Ant.(92)	H.Pre.(46)	H.Sub.(39)	C.Des.(28)	Des.(27)
OCN	64.9	66.5	65.1	55.4	69.6	64.1	57.1	66.7
+CN inj,	67.4(+2.5)	70.6(+4.1)	66.1(+1.0)	60.9(+5.5)	73.9(+4.3)	66.7(+2.6)	64.3(+7.2)	77.8(+11.1)
+OMCS	68.8(+3.9)	63.9(-2.6)	62.4(-2.7)	60.9(+5.5)	71.7(+2.1)	59.0(-5.1)	64.3(+7.2)	74.1(+7.4)
+ATOMIC	62.8(-2.1)	66.0(**-0.5**)	60.6(-4.5)	52.2(-3.2)	63.0(-6.6)	56.4(-7.7)	60.7(**+3.6**)	74.1(**+7.4**)
+OMCS+CN	71.6(+6.7)	71.6(+5.1)	64.2(+0.9)	59.8(+4.4)	69.6(+0.0)	69.2(+5.1)	75.0(+17.9)	70.4(+3.7)

Table 8: Accuracies for each CommonsenseQA question type: **AtLoc.** means *AtLocation*, **Cau.** means Causes, **Cap.** means *CapableOf*, **Ant.** means *Antonym*, **H.Pre.** means *HasPrerequiste*, **H.Sub** means *HasSubevent*, **C.Des.** means *CausesDesire*, and **Des.** means *Desires*. Numbers beside types denote the number of questions of that type.

slightly or did not change. Based on these results, we infer that commonsense knowledge-injection with attention is making an impact on models' weight distributions. The model is able to do better on questions that require commonsense but is losing performance on other types, suggesting a direction for future research in developing more robust (e.g., conditional) injection methods. Moreover, pre-training on knowledge-bases seems to have a larger impact on models' weight distributions, resulting in inferior performance. This weight distribution shift also favors of commonsense, as we see that commonsense types are not affected as much as other types. We also conducted similar analysis for CommonsenseQA. Since all questions in CommonsenseQA require commonsense reasoning, we classify questions based on the ConceptNet relation between the question concept and correct answer concept. The intuition is that the model needs to capture this relation in order to answer the question. The accuracies for each question type are shown in Table 8. Note that we have omitted question types that have less than 25 questions. We can see that with ConceptNet relation-injection, all question types got performance boosts, for both OCN model and OCN pre-trained on OMCS, suggesting that knowledge is indeed helpful for the task. In the case of OCN pre-trained on ATOMIC, although the overall performance is much lower than OCN baseline, it is interesting to see that performance for the "Causes" type is not significantly affected. Moreover, performance for "CausesDesire" and "Desires" types actually got much better. As noted by (Sap et al., 2019), "Causes" in ConceptNet is similar to "Effects" and "Reac-

tions" in ATOMIC; and "CausesDesire" in ConceptNet is similar to "Wants" in ATOMIC. This result also correlates with our findings from our analysis on DREAM, wherein we found that models with knowledge pre-training perform better on questions that fit knowledge domain but perform worse on others. In this case, pre-training on ATOMIC helps the model do better on questions that are similar to ATOMIC relations, even though overall performance is inferior. Finally, we noticed that questions of type "Antonym" appear to be the hardest ones. Many questions that fall into this category contain negations, and we hypothesize that the models still lack the ability to reason over negation sentences, suggesting another direction for future improvement.

6 Discussion

Based on our experimental results and error analysis, we see that external knowledge is only helpful when there is alignment between questions and knowledge-base types. Thus, it is crucial to identify the question type and apply the best-suited knowledge. In terms of knowledge-integration methods, attention-based injection seems to be the better choice for pre-trained language models such as BERT. Even when alignment between knowledge-base and dataset is sub-optimal, the performance would not degrade. On the other hand, pre-training on knowledge-bases would shift the language model's weight distribution toward its own domain, greatly. If the task domain does not fit knowledge-base well, model performance is likely to drop. When the domain of the knowledge-base aligns with that of the dataset perfectly, both knowledge-integration methods bring

performance boosts and a combination of them could bring further gain.

7 Future Work

We have presented a survey on two popular knowledge bases (ConceptNet and ATOMIC) and recent knowledge-integration methods (attention and pre-training), on commonsense QA tasks. Evaluation on two QA datasets suggests that alignment between knowledge-bases and datasets plays a crucial role in knowledge-integration. We believe it is worth conducting a more comprehensive study of datasets and knowledge-bases and putting more effort towards defining an auxiliary learning objective, in a constrained-optimization (i.e., multi-task learning) framework, that identifies the type of knowledge required, based on data characteristics. In parallel, we are also interested in building a *global commonsense knowledge base* by aggregating ConceptNet, ATOMIC, and potentially other resources like FrameNet (Baker et al., 1998) and MetaNet (Dodge et al., 2015), on the basis of a shared-reference ontology (following the approaches described in (Gangemi et al., 2010) and (Scheffczyk et al., 2010)): the goal would be to assess whether injecting knowledge structures from a semantically-cohesive lexical knowledge base of commonsense guarantees stable model accuracy across datasets.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Ernest Davis. 2014. *Representations of commonsense knowledge*. Morgan Kaufmann.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. Metanet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari. 2010. Interfacing wordnet with dolce: towards ontowordnet. *Ontology and the Lexicon: A Natural Language Processing Perspective*, pages 36–52.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Jerry R Hobbs, William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws. 1987. Commonsense metaphysics and lexical semantics. *Computational linguistics*, 13(3-4):241–250.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *ArXiv*, abs/1908.05646.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302, Florence, Italy. Association for Computational Linguistics.

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

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

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2019. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *ArXiv*, abs/1909.05311.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *AAAI*.

Todor Mihaylov and Anette Frank. 2018. Knowledge-able reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaoman Pan, Kai Sun, Dian Yu, Heng Ji, and Dong Yu. 2019a. Improving question answering with external knowledge. *CoRR*, abs/1902.00993.

Xiaoman Pan, Kai Sun, Dian Yu, Heng Ji, and Dong Yu. 2019b. Improving question answering with external knowledge. *CoRR*, cs.CL/1902.00993v1.

Matthew E. Peters, Mark Neumann, IV RobertL.Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. *ArXiv*, abs/1909.04164.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Qiu Ran, Peng Li, Weiwei Hu, and Jie Zhou. 2019. Option comparison network for multiple-choice reading comprehension. *CoRR*, abs/1903.03033.

Maarten Sap, Ronan LeBras, 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 *AAAI*.

Jan Scheffczyk, Collin F Baker, and Srini Narayanan. 2010. Reasoning over natural language text by means of framenet and ontologies. *Ontology and the lexicon: A natural language processing perspective*, pages 53–71.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1223–1237, Berlin, Heidelberg. Springer-Verlag.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. Dream: A challenge dataset and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66, Brussels, Belgium. Association for Computational Linguistics.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada. Association for Computational Linguistics.

Dirk Weissenborn, Tom'avs Kovcisk'y, and Chris Dyer. 2018. Dynamic integration of background knowledge in neural nlu systems.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *ArXiv*, abs/1611.01604.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Cite arxiv:1906.08237Comment: Pre-trained models and code are available at https://github.com/zihangdai/xlnet.

Adams Wei Yu, David Dohan, Thang Luong, Rui Zhao, Kai Chen, and Quoc Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2018a. From recognition to cognition: Visual commonsense reasoning. *CoRR*, abs/1811.10830.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

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