Cracking the Contextual Commonsense Code: Understanding Commonsense Reasoning Aptitude of Deep Contextual Representations

Jeff Da **Jungo Kasai**
Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA, USA
{jzda,jkasai}@cs.washington.edu

Abstract

Pretrained deep contextual representations have advanced the state-of-the-art on various commonsense NLP tasks, but we lack a concrete understanding of the capability of these models. Thus, we investigate and challenge several aspects of BERT's commonsense representation abilities. First, we probe BERT's ability to classify various object attributes, demonstrating that BERT shows a strong ability in encoding various commonsense features in its embedding space, but is still deficient in many areas. Next, we show that, by augmenting BERT's pretraining data with additional data related to the deficient attributes, we are able to improve performance on a downstream commonsense reasoning task while using a minimal amount of data. Finally, we develop a method of fine-tuning knowledge graphs embeddings alongside BERT and show the continued importance of explicit knowledge graphs.

1 Introduction

Should I put the toaster in the oven? Or does the cake go in the oven? Questions like these are trivial for humans to answer, but machines have a much more difficult time determining right from wrong. Researchers have chased mimicking human intelligence through linguistic commonsense as early as McCarthy (1960):

> ... [machines that] have much in common with what makes us human are described as having common sense. (McCarthy, 1960).

Such commonsense knowledge presents a severe challenge to modern NLP systems that are trained on a large amount of text data. Commonsense knowledge is often implicitly assumed, and a statistical model fails to learn it by this reporting bias (Gordon and van Durme, 2013). This critical difference of machine learning systems from human intelligence hurts performance when given examples outside the training data distribution (Gordon and van Durme, 2013; Schubert, 2015; Davis and Marcus, 2015; Sakaguchi et al., 2019).

On the other hand, NLP systems have recently improved dramatically with contextualized word representations in a wide range of tasks (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019). These representations have the benefit of encoding context-specific meanings of words that are learned from large corpora. In this work, we extensively assess the degree to which these representations encode grounded commonsense knowledge, and investigate whether contextual representations can ameliorate NLP systems in commonsense reasoning capability.

We present a method of analyzing commonsense knowledge in word representations through attribute classification on the semantic norm dataset (Devereux et al., 2014), and compare a contextual model to a traditional word type representation. Our analysis shows that while contextual representations significantly outperform word type embeddings, they still fail to encode some types of the commonsense attributes, such as visual and perceptual properties. In addition, we underscore the translation of these deficiencies to downstream commonsense reasoning tasks.

We then propose two methods to address these deficiencies: one implicit and one explicit. Implicitly, we train on additional data chosen via attribute selection. Explicitly, we add knowledge embeddings during the fine-tuning process of contextual representations. This work shows that knowledge graph embeddings improve the ability of contextual embeddings to fit commonsense attributes, as well as the accuracy on downstream reasoning tasks.

2 Attribute Classification

First, we preform an investigation to see if the output from BERT is able to encode the necessary features to determine if an object has a related attribute. We propose a method to evaluate BERT's representations and compare to previous non-contextual GloVe (Pennington et al., 2014) baselines, using simple logistic classifiers.

2.1 Commonsense Object Attribution

To get labels for attribute features of commonsense features of objects, we utilize CSLB, a semantic norm dataset collected by the Cambridge Centre for Speech, Language, and the Brain (Devereux et al., 2014). Semantic norm datasets are created through reports from human participants asked to label the semantic features of a given object. Thus, a proportion of these features are obvious to humans, but may be difficult to find written in text corpora. This is notably different from the collection methods of prominent commonsense databases, such as ConceptNet (Speer and Havasi, 2013).

CSLB gives 638 different attributes describing a variety of objects provided by 123 participants. To make results consistent between baselines (GloVe) and BERT, we first preprocess the attributes present in CSLB. We removed attributes with two-word names, ambiguous meanings (i.e. homographs), or missing GloVe representations. This gives a 597 attribute vocabulary. Examples of objects described are *zebra*, *wheel*, and *wine*. Example of attributes are *is upright*, *is a toy*, and *is an ingredient*.

2.2 Contextualization

Since BERT is commonly utilized at the sequence embedding level (Devlin et al., 2019), we develop a contextualization module to allow representations of (*object, attribute*) pairs, allowing us to acquire one sequence embedding from BERT for each pair. From a high level, we want to develop a method to transform (*object, attribute*) into simple grammatical sentences.

For each *(object, attribute)* pair, we raise the pair to a sentence structure such that the attribute is describing the object. We would enforce the following representation, in line with the procedure of Devlin et al. (2019):

$[CLS]$ c_{prefix} noun c_{affix} adj. c_{postfix} $[SEP]$

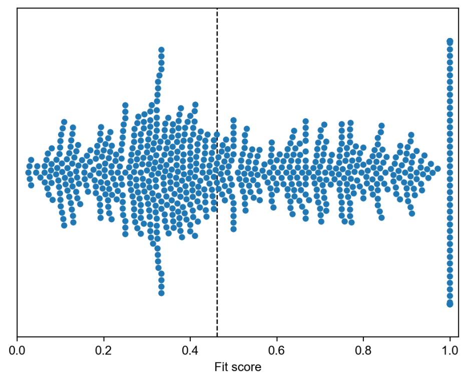
The goal is to create a simple formula that allows the model to isolate the differences between the object-attribute (noun-adjective) pairs, rather than variation in language. c_{prefix} represents previous context, i.e. context that appears before the word. c_{affix} is context that appears between the noun and the adjective. c_{postfix} is context that closes out the sentence.

We illustrate this algorithm for use with CSLB, but this methodology can be used for any dataset, such as other semantic norm datasets. We use this process for each *(object, attribute)* pair in CSLB. First, we check if any words in the attribute need to be changed. For example, in CSLB, instead of *does deflate*, we use *deflates* as the attribute text, since it simplifies the language. Then, for c_{prefix}, we use either *A* or *An*, and for c_{postfix}, and use a period. For c_{affix}, we use either *is* or nothing, depending on the attribute. Some example sentences would be: *(shirt, made of cotton)* would become "A shirt is made of cotton." and *(balloon, does deflate)* becomes "A balloon deflates." See the appendix for full pseudocode.

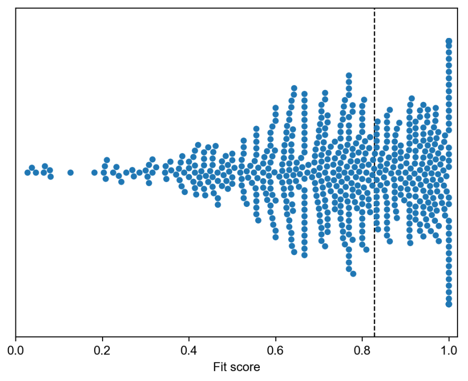
We find that this method is a better alternative to simply creating a sequence with the concatenation of the object and the attribute. Some attribute-object pairs translate better to English than others. For example, "wheel does deflate" might be a relatively uncommon and awkward English phrase when compared to more natural phrases such as "shirt made of cotton".

2.3 Determining Attribute Fit

We explore if word embeddings contain the necessary information within their embedding space to classify various semantic attributes. Our procedure involves use of a simple logistic classifier to classify if an *attribute* applies to a candidate *object*. We create a list of *(object, attribute)* pairs as training examples for the logistic classifier (thus, there are $n_{objects} \times n_{attributes}$ training examples in total). We then train logistic classifiers for each attribute, and use leave-one-out accuracy as accuracy – averaging the leave-out-one result across all $n_{objects}$ classifiers, since we leave out a different object each time. For example, to examine the attribute *made of cotton*, we train on all objects except one, using the label 1 if the object is made of cotton, and 0 otherwise. Then, we test to see if the left-out object is classified correctly. We repeat $n_{objects}$ times, removing a different object each

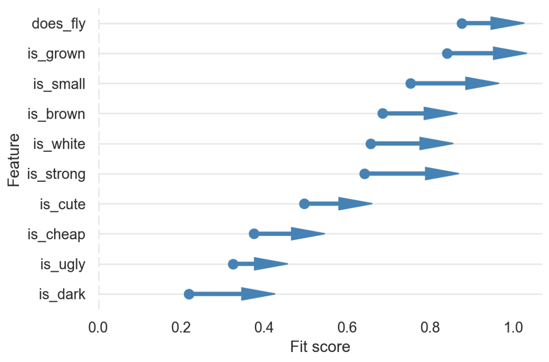


(a) GloVe

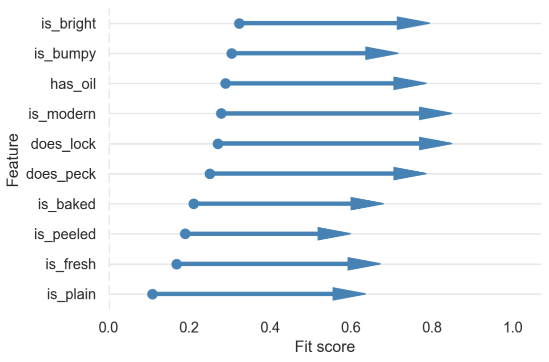


(b) BERT

Figure 1: Swarm plots showing attribute fit scores for GloVe (left) and BERT (right). Each dot represents a single attribute, displayed along the x-axis according to the classifier's ability to fit that feature with the given embeddings. The y-axis is not significant, and instead, dots are displaced along the y-axis instead of overlapping to show quantity. The median fit score per embedding type is displayed with a dotted line.



(a) Small increase in fit score (< 0.15)



(b) Large increase in fit score (> 0.3)

Figure 2: Differences between fit scores when using GloVe (start of arrow) or BERT (end of arrows) embeddings.

time. To judge fit, we use F1 score, as F1 score is not affected by dataset imbalance. We consider other classifiers, such as SVD classifiers, but we find that there is no significant empirical difference between the classifiers. For baseline tests, we use the pretrained 300 dimensional GloVe embeddings,[1] as they have shown to perform better than word2vec embeddings (Lucy and Gauthier, 2017). See appendix for specific logistic regression parameters, such as the number of update steps used.

2.4 Attribute Scores

We show our findings for feature fit for each attribute. Figure 1 highlights that BERT is much stronger on this benchmark – the median fit score is nearly double that of the previously reported GloVe baselines. This suggests that BERT encodes commonsense traits much better than previous baselines, which is suggestive of its strong scores on several commonsense reasoning tasks. Notably, we can see that much fewer features have a fit score less than 0.5. We observe that many more traits have a perfect fit score of 1.0. However, our results also show that BERT is still unable to fully fit many attributes. This underscores that BERT still lacks much attribution ability, perhaps in areas outside of its training scheme or pretraining data. Seen in Figure 2 is the change in fit scores between GloVe and BERT. We can see that some traits exhibit much larger increases – in particular, physical traits such as *made of wood*, *does lock*, and *has a top*. Traits that are more abstract tend to have a lesser increase. For example, *is creepy* and *is strong* still are not able to be fit by the contextualized BERT module.

[1] https://nlp.stanford.edu/projects/glove/

Metric	Visual	Encyclopedic	Functional	Perceptual	Taxonomic	Overall
Median_{GloVe}	46.2	38.9	44.4	49.0	89.1	46.1
Median_{BERT}	83.3	76.2	78.3	80.0	100	82.7
Δ	**+37.1**	**+37.3**	**+33.9**	**+31.0**	**+10.9**	**+36.6**

Table 1: Comparison of median logistic classifier fit scores (out of 100 percent fit) across categories defined in CSLB.

Category	Lower scoring attributes (fit score < 1.0)	Attributes perfectly fit (fit score = 1.0)
Visual	is triangular, is long and thin, is upright, has two feet, does swing, is rigid	does come in pairs, has a back, has a barrel, has a bushy tail, has a clasp
Encyclopedic	is hardy, has types, is found in bible, is American, does play, is necessary essential	does grow on plants, does grow on trees, does live in rivers, does live in trees, does photosynthesize, has a crew
Functional	does work, does spin, does support, does drink, does breathe, does hang	does DIY, does carry transport goods, does chop, does drive
Perceptual	is chewy, does rattle, is wet, does squeak, is rough, has a strong smell	does bend, has a sting, has pollen, has soft flesh, is citrus, is fermented
Taxonomic	is a home, is a dried fruit, is a garden tool, is a vessel, is a toy, is an ingredient	is a bird of prey, is a boat, is a body part, is a cat, is a citrus fruit, is a crustacean

Table 2: Fine-grained comparison across categories between attributes that lack some level of fit (left) and perfectly fit attributes (right) with classification using BERT representations.

Table 1 shows a comparison of fit scores across different types of attribute categories. These categories are defined per attribute in CSLB (Devereux et al., 2014). Visual attributes define features that can be perceived visually, such as *is curved*. Perceptual defines attributes that can be perceived in other non-visual ways, such as *does smell nice*. Functional describes the ability of an object, such as *is for weddings*. Taxonomic defines a biological or symbolic classification of an object like *is seafood*. Finally, encyclopedic are traits that may be the most difficult to classify, as they are attributes that most pertain to abstract commonsense, such as *is collectible*.

BERT has stronger scores in all categories, and just short of double the overall accuracy. Importantly, however, it struggles to classify many categories of objects. In taxonomic categories, it is able to perfectly fit more than half the objects. We suspect that this is intuitive, as BERT is trained on text corpora that allow for learning relationships between classes of objects and the object itself. GloVe notably also preforms strong in this category, for the same reasons. BERT scores the lowest on encyclopedic traits, which most closely resemble traits that would appear in commonsense tasks. This suggests that BERT maybe be relatively deficient in regards to reasoning about commonsense attributes.

We also examine specific attributes where BERT is fully fit (with a perfect fit score), and compare those attributes to features where BERT is unable to fit. Table 2 shows examples of both levels of fit. BERT is able to fit many features that would be easily represented in text, such as $does\ bend$, $does\ grow\ on\ plants$, and $does\ drive$. It is unable to fit traits that may be less common in text and more susceptible to the reporting bias, such as $is\ American$, $is\ chewy$, and $has\ a\ strong\ smell$. Surprisingly, it is also unable to fit several features that would be likely common in text such as $is\ a\ toy$, suggesting that BERT's training procedure is lacking coverage of many everyday events perhaps due to the reporting bias.

2.5 Do Knowledge Graphs Help?

We extend our investigation with two inquiries. First, given the large gain in accuracy over GloVe, we wonder if BERT embeddings now encode the same information that external commonsense knowledge graphs (such as ConceptNet (Speer and Havasi, 2013)) provide. Second, we question if it is possible to increase the overall accuracy above the accuracy presented by using BERT embeddings (otherwise, it could mean that the deficit is simply because the logistic classifier does not have

System	Median
GloVe	46.1
BERT$_{LARGE}$	82.7
ConceptNet	23.2
BERT$_{LARGE}$ + ConceptNet	**90.7**

Table 3: Results for attribute classification with ConceptNet as a knowledge graph source.

needed capacity (Liu et al., 2019a)).

We use ConceptNet (Speer and Havasi, 2013) for our experiments. We label each relationship type with an index. ($antonym$ as 0, $related\_to$ as 1, etc.) During classification, we query the knowledge base with the object and the attribute and check if there are any relationships between the two. We embed the indexes of matched relationships to randomly initialized embeddings and concatenate them with the original BERT embeddings. If more than one relationship is found, we randomly choose a relationship to use.

Table 3 shows our results. By itself, the explicit commonsense embeddings do not have enough coverage to learn classifications of each attribute, since the knowledge graph does not contain information about every ($object$, $attribute$) pair. However, by combining the knowledge graph embeddings with the BERT embeddings, we illustrate that knowledge graphs cover information that BERT is unable to generate the proper features for. In addition, the results suggest that BERT is deficient over various attributes, and the traditional knowledge graphs are able to cover this feature space. These results support the hypothesis that BERT simply lacks the features rather than the problem of the logistic classifier.

3 Improving BERT's Representations

We have gained an understanding of the types of commonsense attributes BERT is able to classify and encode in its embeddings, and also have an understanding of the types of attributes that BERT's features are deficient in covering. In Section 2.5, we have shown that commonsense knowledge graphs may also help encode information that extends beyond BERT's embedding features. However, we have yet to know whether this BERT's deficiency will translate to any of BERT's downstream reasoning ability, which is ultimately more important.

We empirically address the gap between attribute classification and downstream ability in BERT. First, we demonstrate that there is a correlation between low-scoring attributes and low accuracy on reasoning questions that involve those attributes. Then, we leverage our investigation to build two baseline methods of improving BERT's commonsense reasoning abilities (Figure 4). Since BERT is trained on implicit data, we explore a method of using RACE (Lai et al., 2017) alongside a list of attributes that BERT is deficient in (such as the one in Section 2.4). We also extend our investigation in Section 2.5 on commonsense knowledge graphs by proposing a method to integrate BERT with external knowledge graphs. See appendix for hyperparameters.

Passage: For my anniversary with my husband, I decided to cook him a very fancy and nice breakfast. One thing I had always wanted to do but never got to try was making fresh squeezed orange juice. I got about ten oranges because I wasn't sure how much I was going to need to make enough juice for both me and my husband. I got home and pulled my juicer out from underneath my sink. I began using the juicer to squeeze the juice out of my orange juice. I brought my husband his breakfast with the orange juice, and he said that the juice was his favorite part!
How were the oranges sliced? **a) in half** b) in eighths
When did they plug the juicer in? a) after squeezing oranges **b) after removing it from the box**

Table 4: Example of a prompt from MCScript 2.0 (Ostermann et al., 2018), an everyday commonsense reasoning dataset. Questions often require script knowledge that extends beyond referencing the text.

3.1 Background: MCScript 2.0

We leverage MCScript 2.0 (Ostermann et al., 2019) for several investigations in this paper. MCScript 2.0 is a downstream commonsense reasoning dataset. Each datum involves one passage, question, and two answers, and the goal is to pick the correct answer out of the two choices. Many questions involve everyday scenarios and objects, which helps us link our semantic norm results to more downstream reasoning capability. Table 4

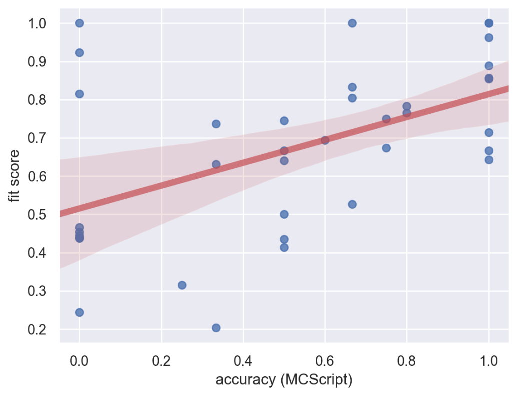


Figure 3: Linear regression fit of accuracy on MCScript 2.0, per attribute, versus fit score, with the inner 90 percent bootstrap confidence intervals highlighted (n = 1000). Each dot represents the accuracy of questions related to one attribute.

shows an example.

3.2 Do Low Classification Scores Result in Low Performance?

We examine if low-scoring attributes result in low downstream performance, and high-scoring attributes also result in high downstream performance. For each question in MCScript, we relate that question to 1 or more of the attributes in the previous experiment. For example, a question might be talking about whether to use a camera flash, and would be thus related to the traits *does have flash*, *is dark*, and *is light*. Here we aim to empirically assess deficiencies in BERT's ability and their downstream implications. For instance, if it is unable to fit *does have flash*, will it have a gap in knowledge in areas regarding camera flash? If a given feature does not have a related question, we do not include it in our experiments. In total, $n_{\text{questions}} = 193$, and $n_{\text{attributes}} = 92$.

For the MCScript model, we simply classify based on the $[CLS]$ token, as suggested in Devlin et al. (2019). We softmax over the logits between the two answers when producing our final answers, and split the passage-question pair and answer by a $[SEP]$ token. The attribute-related questions here are from the development set only.

Seen in Figure 3 are the results. We do not see a clear pattern, but we can still make several observations. First, we notice that there are simply a lot of items with a high fit score. Next, there are a lot of attributes that BERT simply gets correct. However, notably, BERT is less consistent with getting items that have a low fit score (< 0.5). We can also notice that all attributes that have high accuracy on MCScript also have a high fit score.

3.3 Implicit Fine-Tune Method

We develop a method of fine-tuning with additional data based on the deficiencies found in the previous section. We fine-tune on additional data, but we select only data related to attributes that BERT is deficient in.

3.3.1 Data Selection

In our experiments, we use RACE (Lai et al., 2017) as our supplementary dataset. While we can fine-tune on the entire dataset, we can also select a subset that directly targets the deficient attributes in semantic norm. To select such a subset, we define a datum as related if any words match between the datum in the supplementary dataset and the deficient feature in semantic norm, stemming all words beforehand. For some attributes, we remove frequent words ("is, "does", and "has") to avoid matching too many sentences within RACE.

Since each datum in RACE involves a question, answer, and passage, we allow matches between either of the three texts, and do not differentiate between matches in the question, answer, and passage. We find that this keeps around a third of the data in RACE (around 44K, out of the 97K data present in RACE). It is also key that this data selection process does not require access to the downstream task dataset. Thus, this procedure has the ability to generalize to other tasks beyond MCScript 2.0.

3.3.2 Fine-Tuning Procedure

We fine-tune BERT's language objectives on RACE. We do not change the properties of either objective, to keep comparability between our analysis and BERT. This mimics Devlin et al. (2019), and thus, we fine-tune the token masking objective and the next sentence prediction objective. Several works have improved on BERT's language objectives (Yang et al., 2019; Liu et al., 2019b), but we keep the language objectives in BERT intact for comparison.

After fine-tuning on RACE, we fine-tune on MCScript with the classification objective only. We do this since we need to build a classification layer for the specific task, as noted in Devlin et al. (2019). We do not freeze the weights in this process, as to keep comparability with the fine-tuning

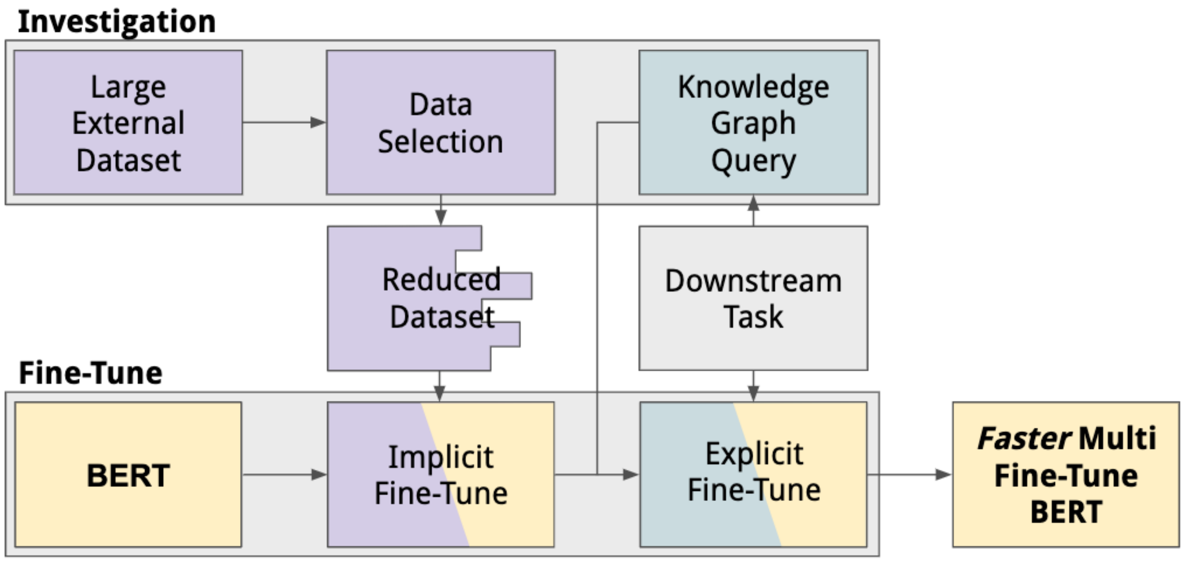


Figure 4: Outline of our baseline method of improving BERT for commonsense reasoning. Our method fine-tunes BERT through multiple facets while optimizing for accuracy and reduced train steps. We use RACE (Lai et al., 2017) as an external dataset, and MCScript 2.0 (Ostermann et al., 2019) as our downstream task.

procedure in Devlin et al. (2019).

3.4 Explicit Fine-Tune Method

Motivated by our results in 2.5, we develop a method of integrating knowledge graph embeddings with the BERT embeddings. First, we query knowledge graphs based on the given text to find relationships between objects in the text. Then, we generate an embedding for each relationship found (similar to Section 2.5). Finally, we fine-tune these embeddings alongside the BERT embeddings.

3.4.1 Knowledge Graph Query

We query a suite of knowledge bases (ConceptNet (Speer and Havasi, 2013), WebChild (Tandon et al., 2017), ATOMIC (Sap et al., 2019)) to create knowledge graph embeddings. First, we examine all relationships, indexing each unique relationship sequentially. Then, during fine-tuning, for each prompt in MCScript 2.0, we query the knowledge bases to find any *(start_node, end_node, edge)* matches between the knowledge base and the current prompt. For example, if eat and $dinner$ are both present in the text, the relationship $at\_location$ in ConceptNet would match (Figure 5). We record the index of the matched relationship, keeping a list of matched relationships per word in the prompt. If a *start_node* spans more than one word, we record the match as occurring for the first word in the phrase.

System	Acc.	Data
BERT$_{LARGE}$ + RACE	84.3	98 K
BERT$_{LARGE}$ + RACE (random)	84.0	44 K
BERT$_{LARGE}$ + RACE (selected)	**84.5**	44 K

Table 5: Test set results from the implicit method on MCScript 2.0. "selected" indicates a subset of RACE that consists of misclassified attributes in semantic norm. "random" is a randomly chosen subset.

3.4.2 Fine-Tuning Procedure

We fine-tune our knowledge graph embeddings alongside the BERT fine-tuning procedure. We randomly initialize an embedding for each relationship and each knowledge graph. We choose an embedding for each word in the prompt (randomly, if there is more than one relationship associated), creating a sequence of knowledge graph embeddings. We create a sequence embedding for the 30-dimensional graph embeddings by feeding the sequence through an bidirectional LSTM. Then, during fine-tuning, we classify each datum in MCScript based on the concatenation of the explicit graph sequence representation and the BERT sequence embedding (i.e. $[CLS]$), as per Devlin et al. (2019).

3.5 Results and Analysis

Table 5 shows the results from the implicit method. Accuracy is consistent across the board, with all models giving about a 2% downstream ac-

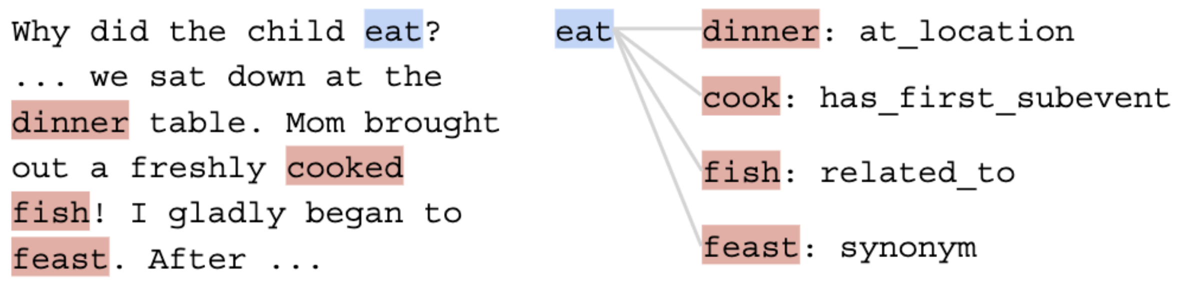

Why did the child eat?
... we sat down at the dinner table. Mom brought out a freshly cooked fish! I gladly began to feast. After ...
eat
dinner: at_location
cook: has_first_subevent
fish: related_to
feast: synonym

Figure 5: Visualization of ConceptNet knowledge base queries. The word *eat* is being queried with the other words in the text, with the valid edges discovered displayed against the left.

System	Accuracy
Human (Ostermann et al., 2019)	97.4
Random Baseline	48.9
BERT$_{LARGE}$	82.3
with ConceptNet	83.1
with WebChild	82.7
with ATOMIC	82.5
with all KB	83.3
with all KB + RACE (selected)	**85.5**

Table 6: Test set results for knowledge base embeddings on MCScript 2.0.

curacy boost. However, the model with the less amount of data (RACE, selected from deficiencies only) achieves equivalent accuracy to the entire RACE dataset, while using only half the amount of data. This underscores the importance of the abstract semantic norm task, as the related data selection process was effective in choosing examples that are directly related to deficiencies.

Table 6 shows our results with explicit knowledge embeddings. Each knowledge base improves accuracy, with ConceptNet giving the largest performance boost. ATOMIC gives the smallest boost, likely because the ATOMIC edges involve longer phrases, which means less matches, and the overlap between ATOMIC text and the text present in the task is not as large as either ConceptNet or WebChild.

We can also combine the explicit knowledge base embeddings and the implicit RACE fine-tuning, yielding the highest accuracy (with all KB + RACE (subset) in Table 6). The knowledge embeddings provide a similar +1% absolute improvement (85.5 vs. 84.5), suggesting that the knowledge embeddings cover different aspects and relationships in the text than learned during fine-tuning on RACE.

4 Related Work

Similar to our attribute classification investigation, several other works have used applied semantic norm datasets to computational linguistics (Agirre et al., 2009; Bruni et al., 2012; Kiela et al., 2016). Methodologically, our work is most similar to Lucy and Gauthier (2017), who use a logistic regression classifier to determine fit score of word type embeddings based on leave-one-out verification. Forbes et al. (2019) investigates the commonsense aptitude of contextual representations. However, our work differs in several important ways: 1) we connect our analysis to downstream reasoning aptitude, underscoring the importance of the semantic norm analysis, and 2) we introduce various ways of improving BERT, motivated by our analysis.

In contemporaneous work, various research has been done in improving upon BERT's performance through knowledge augmentation. Implicitly, Sun et al. (2019) explores fine-tuning on in-domain data, similarly to our fine-tuning on the RACE dataset (Lai et al., 2017). They discover an increase in accuracy that is especially prevalent over smaller datasets. Our work differs in that we do not fine-tune on the entire domain data, but rather select a smaller subset of data to fine-tune on. Other work extends BERT to domains where its original training data does not suffice (Beltagy et al., 2019; Lee et al., 2019). RoBERTa (Liu et al., 2019b) also pretrains on RACE, and finds increased results through altering several of BERT's pretraining tasks, claiming that BERT was extensively undertrained. Explicitly, ERNIE, Zhang et al. (2019) introduces information to contextual representations during pretraining. ERNIE uses word-level fusion between the contextual representation and explicit information.

Prior work has developed several benchmark datasets to assess commonsense knowledge of NLP models (Roemmele et al., 2011; Mostafazadeh et al., 2016; Zhang et al., 2017; Zellers et al., 2018, 2019; Ostermann et al., 2018, 2019; Sakaguchi et al., 2019). These benchmarks are typically posed as question answering, but we use semantic norm datasets to specifically assess BERT's ability to represent grounded attributes. Further, we demonstrate that these abstract attributes can be used to enhance BERT's representations and improve the downstream performance.

5 Conclusion

We found that BERT outperforms previous distributional methods on an attribute classification task, highlighting possible reasons why BERT improves the state-of-the-art on various commonsense reasoning tasks. However, we show that BERT still lacks proper attribute representations in many areas. We developed implicit and explicit methods of remedying this deficit on the downstream task. We demonstrated that, individually and combined, both methods can improve scores on the downstream reasoning task. We motivate future work in probing and improving the ability of neural language models to reason about everyday commonsense.

Acknowledgments

The authors thank Maxwell Forbes, Keisuke Sakaguchi, and Noah A. Smith as well as the anonymous reviewers for their helpful feedback. JD and JK are supported by NSF Multimodal and the Funai Overseas Scholarship respectively.

References

Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL-HLT*.

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *ArXiv*, abs/1903.10676.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proc. of ACL*.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58.

Barry J. Devereux, Lorraine K. Tyler, Jeroen Geertzen, and Billi Randall. 2014. The centre for speech, language and the brain (CSLB) concept property norms. *Behavior Research Methods*, 46(4).

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*.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? *Proc. of the 41st Annual Conference of the Cognitive Science Society*.

Jonathan Gordon and Benjamin van Durme. 2013. Reporting bias and knowledge acquisition. In *Proc. of AKBC*.

Douwe Kiela, Luana Bulat, Anita L. Vero, and Stephen Clark. 2016. Virtual embodiment: A scalable long-term strategy for artificial intelligence research. *ArXiv*, abs/1610.07432.

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*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *ArXiv*, abs/1901.08746.

Nelson F. Liu, Matthew Ph Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proc. of NAACL-HLT*.

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

Li Lucy and Jon Gauthier. 2017. Are distributional representations ready for the real world? Evaluating word vectors for grounded perceptual meaning. In *Proc. of RoboNLP*.

Jeanette McCarthy. 1960. Programs with common sense.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proc. of NAACL-HLT*.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In *Proc. of LREC*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. MCScript2.0: A machine comprehension corpus focused on script events and participants. In *Proc. of *SEM*, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proc. of EMNLP*, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL-HLT*.

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI 2011 Spring Symposium*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WINOGRANDE: An adversarial winograd schema challenge at scale. *ArXiv*, abs/1907.10641.

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 *Proc. of AAAI*.

Lenhart Schubert. 2015. What kinds of knowledge are needed for genuine understanding? In *Proc of Cognitum*.

R. Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? *ArXiv*, abs/1905.05583.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2017. WebChild 2.0 : Fine-grained commonsense knowledge distillation. In *Proc. of ACL 2017, System Demonstrations*, Vancouver, Canada. Association for Computational Linguistics.

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.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL*.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *TACL*, 5.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proc. of ACL*, Florence, Italy. Association for Computational Linguistics.

A Appendices

A.1 Hyperparameters

Seen in Table 7 is a list of hyperparameters for our experiments. We use the same parameters for both uses of explicit knowledge embeddings.

Regression Classifier	
Penalty	L2
# Penalty Coefficient	1.0
Iteration count	200
Optimizer	lbfgs
Patience	1e-4
Explicit Knowledge Embeddings	
Embedding size	10
Knowledge bases used	3
BERT Fine-Tuning	
Maximum sequence length	450
Train batch size	32
Learning rate	1e-5
Epochs	4
Warmup	20%
LSTM	
Hidden size	32
Dropout	0.0
Bidirectional	Yes

Table 7: Hyperparameters used throughout experiments.

A.2 Contextualization Module Pseudocode

Psuedocode can be found by referencing Algorithm 1.

Algorithm 1: Contextualization Module for CSLB Attributes

```
contextualize (object, attribute):
to_remove = [does]
if attribute[first word] in to_remove then
    attribute[second word] = make_plural(attribute[second word])
    attribute.remove(attribute[first word])
end if
if starts_with_vowel(attribute[first word]) then
    $c_{\text{prefix}}$ = An
else
    $c_{\text{prefix}}$ = A
end if
needs_affix = [made]
if attribute[first word] in needs_affix then
    $c_{\text{affix}}$ = is
else
    $c_{\text{affix}}$ = None
end if
$c_{\text{postfix}}$ = .
return $c_{\text{prefix}}$ + object + $c_{\text{affix}}$ + attribute + $c_{\text{postfix}}$
```