Building on BERT with Common Object Image Captions for Commonsense Validation

Jaclyn Andrews
University of California Berkeley MIDS
jandrews1@berkeley.edu

Abstract

Commonsense validation tasks evaluate the presence or absence of common sense within language. This paper proposes two approaches for classifying commonsense and nonsense sentence pairs from SemEval 2020 Task 4 subtask A. The first approach uses outputs from various layers of Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) to best capture each sentence's "sense". The second approach fine tunes BERT on the Common Objects in Context (COCO) image caption dataset (Lin et al., 2014) to infuse the model with common knowledge that may not be typically written explicitly. Neither approach produced competitive accuracy results compared to the task leaderboard, indicating the need for alternative problem configuration or additional outside knowledge injection to validate commonsense.

1 Introduction

Commonsense reasoning is a subset of natural language understanding that is ultimately aimed at creating models that capture the common sense that humans naturally have. Achieving that would allow for machine learning systems that can seamlessly interact with users in a human-like way and solve problems that require common sense knowledge.

A gap in the way that machines and human approach reasoning is that machines only look at the text they have in front of them or they were trained on, while humans can pull from a lifetime of knowledge and world observations that may or may not be explicitly written down. The creation of large pre-training models such as BERT (Devlin et al., 2018) that have captured massive amounts of text has recentered the conversation around commonsense reasoning.

The task is difficult because the best language models now are trained on large corpus of text, but common sense isn't often written down.

Several benchmarks for commonsense reasoning have been developed aimed at different aspects of natural language understanding. There are benchmarks for identifying the presence or lack of common sense in text, choosing an answer that makes the most common sense, evaluating the reason behind why text does not make common sense and others. In this paper, we focus on comparing two similar sentences and choosing which does not make sense through the ComVE benchmark in subtask A of SemEval 2020 Task 4 (Wang et al., 2020).

2 Background

With over 30 papers in the SemEval 2020 proceedings dedicated to this Commonsense Validation and Explanation (ComVE) task and several other common sense benchmarks like SWAG (Zellers et al., 2018) and Winograd Schema Challenge (Levesque et al., 2012), many researchers have already laid the groundwork for this task.

Most current solutions use bidirectional encoder representations from transformers, or BERT (Devlin et al., 2019). Fine tuning RoBERTa, a more robust version of BERT, is a common approach used among papers with high accuracy (Fadel et al., 2020). However, many researchers have found that the knowledge captured within the pretraining of BERT and even RoBERTa doesn't fully encompass common sense.

Zhang et al. (2020) reached the top of the leaderboard on subtask A using the ConceptNet (Speer et al., 2016) knowledge graph and version of K-BERT (Liu et al., 2019) to inject external knowledge into the language model. ConceptNet is a graph of triples that defines 34 different types of relationships between objects and properties. One example of a triple that contains common sense knowledge is $< whale, hasproperty, big >$. While a Wikipedia page for whales might *imply*

whales are big, the explicit nature of the ConceptNet triples is what makes them useful for commonsense validation model training. Based on the success of the knowledge graph, we will try a different approach to using explicit knowledge for the classification task.

3 Methods

3.1 Dataset

The SemEval 2020 Task 4 subtask A data set contains 10,000 training sentence pairs and 1,000 test sentence pairs. Each sentence pair contains two similar sentences, one of which makes sense and the other of which does not. The label indicates which of the two sentences does **not** make sense.

The task is simply to build a model that can select the sentence that does not make sense. Any outside knowledge can be used to complete the task. Table 1 shows examples of sentence pairs within the dataset, along with their label.

Sent 1	She put water in the bucket.✓
Sent 2	**She put a building in the bucket.✗**
Label	1
Sent 1	**A dog can live in an aquarium.✗**
Sent 2	A shark can live in an aquarium.✓
Label	0

Table 1: Sentence pair examples where ✗ indicates the sentence that does not make sense.

The data set was created specifically for this task, written by human annotators and designed to have equal classes - half the examples in each of the training and test sets start with the nonsense sentence, the other half end with the nonsense sentence. The measurement used in the task leaderboard to evaluate model success is accuracy. Human accuracy on the task is cited as 99.1% (Wang et al., 2020).

3.2 Baseline

The baseline method of classifying sentences pairs in this research was to convert each individual word into Word2Vec (Mikolov et al., 2013) embeddings and average the embeddings across words. This averaged embedding is used as a method to capture semantics of a sentence. Then I used the sentence label (0 for sentences that make sense, 1 for sentences that don't make sense) to run a logistic regression with the embedding vector values as the inputs and choose the sentence with the higher output probability as the positive class. The baseline model achieved 61.5% accuracy on the test set, an improvement over the 50% possible by just guessing the positive class always.

3.3 Design and Implementation

Given the wide variety of topics covered in the data set, BERT is a natural fit to work with since it has already been trained on all massive corpuses of text. The task was configured as a classification problem, with both sentences in a pair fed into BERT simultaneously. This paper proposes two methods to improve upon the Word2Vec baseline model – pulling embeddings from various outputs of BERT to best capture each sentence's "sense" and fine tuning BERT on an outside unlabeled set of image captions prior to classification.

Several hyperparameters were tested in the making of these models including learning rate, dropout rate, batch size, and epsilon. Learning rate and dropout rate had the largest effect on accuracy. Final model hyperparameter configurations can be found in the Appendix 7.

3.4 Experimentation with BERT Outputs

The motivation for reconfiguring BERT's outputs came from Tenney et al. (2019) which found that BERT's layers may actually be capturing different senses of sentence inputs. Tenney et al. (2019) found that lower layers of BERT may capture syntax while higher layers capture semantics.

There are three separate model architectures highlighted in Figure 1 and compared in the final results table.

BERT_{CLS}: BERT_{CLS} is the simplest of the three models (found in the middle of Figure 1) which performs the classification on the last hidden state [CLS] token embeddings. Within this method, the main variable that was varied in testing was the number of retrained layers, with more retrained layers contributing to higher accuracy in the final models. $\text{BERT}_{\text{CLS-Frozen}}$ represents a model architecture in which none of the twelve BERT layers were retrained.

$\text{BERT}_{\text{HiddenStates}}$: Building on Tenney et al. (2019) work, tests were performed to classify sentence pairs using each individual layer's [CLS] token embedding and then combinations including adding, averaging and concatenating different combinations of high performing layers. Combining layer outputs was a tactic that the authors of the

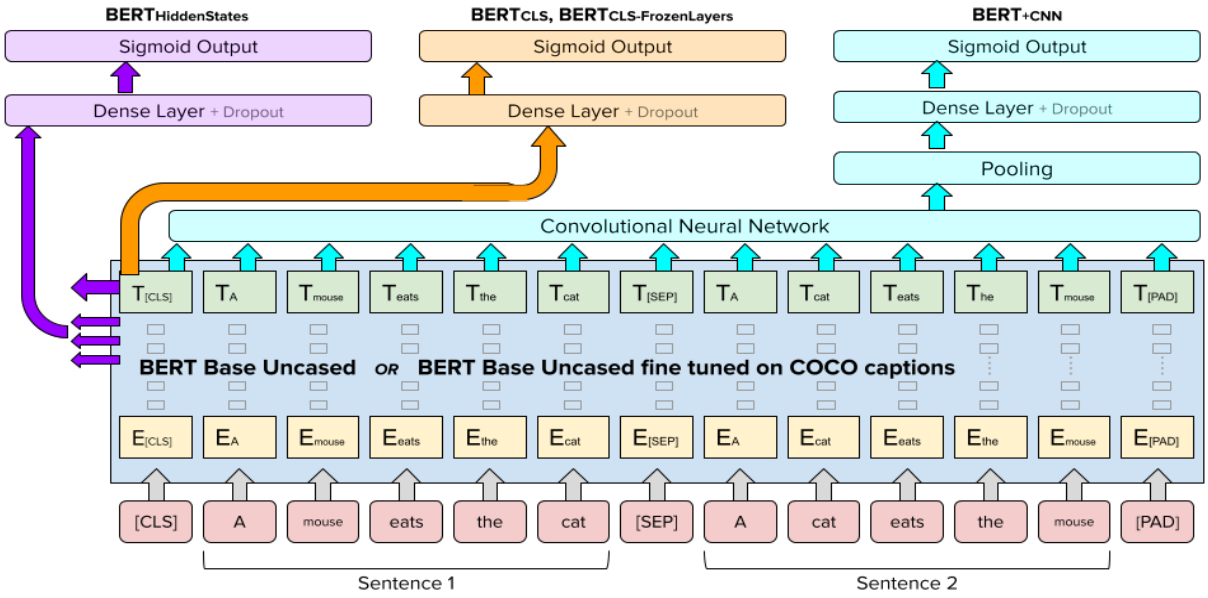

Figure 1: Configuration of top three separate model setups.

original BERT paper employed for named entity recognition (Devlin et al., 2018). Ultimately, the final model used concatenated BERT's top four layer [CLS] token embedding outputs for classification (found on the left side of Figure 1).

BERT$_{+CNN}$: The third model setup in this paper, BERT$_{+CNN}$, uses the sequence embeddings output from BERT and runs the whole sequence through a convolutional neural network (found on the right side of Figure 1). While typically used on images, convolutional neural networks have also been used successfully in sentence classification tasks (Kim, 2014).

All three model configurations were run on both the pretrained BERT base uncased model and also BERT base uncased fine tuned on COCO image captions (BERT$_{+COCO}$) (Lin et al., 2014).

3.5 Image Caption Fine Tuning

The motivation behind fine tuning BERT on image captions was to capture visual observation knowledge that humans naturally gather but may not typically be written down. This approach can be equated to "domain specific" fine tuning methods that other researchers have used (Beltagy et al., 2019). Instead of a specific domain, it is meant to capture a specific type of knowledge, or common sense. Nearly 600,000 common objects in context image dataset (Lin et al., 2014) captions were used for fine tuning. The *BERTforMaskedLM* model from Huggingface library was used to randomly mask 15% of words in the unlabeled captions and fine tune BERT. The fine tuned BERT model was then saved and loaded as the starting point for the sentence pair classification task.

4 Results

All model results on the test set are found in Table 2.

Baseline$_{(Word2Vec)}$	**61.5**	
Model	**BERT Accuracy**	**BERT$_{+COCO}$ Accuracy**
BERT$_{CLS\text{-}Frozen}$	62.8	**64.2**
BERT$_{CLS}$	**72.2**	70.4
BERT$_{HiddenStates}$	**71.1**	70.9
BERT$_{+CNN}$	**72.5**	70.2
Current Leader	**97.0**	
Human	**99.1**	

Table 2: Test set accuracy results.

BERT$_{CLS}$: Using just the embedding output from BERT while freezing all layers of BERT yielded a low accuracy rate at 62.8% just barely higher than the word2vec embedding baseline at 61.5%. The true power of BERT was discovered while retraining layers of BERT during the fine tuning process.

Retraining additional layers of BERT takes more compute power, so testing was performed to find the best balance between accuracy and resources needed. The results of these tests can be found in

Figure 2. The line chart shows accuracy increasing as more layers are trained, evening out without significant improvement after 8 layers. The interesting takeaway from this testing can be found in the bar charts showing the model's confidence in its predictions. Retraining less than 6 layers led to models that were not at all confident in their correct or incorrect predictions. From 6 to 10 layers, the models were noticeably more confident in their correct predictions, but not as confident in incorrect predictions, as expected. While retraining 11 or 12 layers, the models gained confidence in the incorrect predictions, signifying that the models were overfit to training data. All final models were run with 10 layers retrained during fine tuning. The simple model using just the top layer [CLS] token for sentence pair classification yielded 72.2% accuracy.

BERT$_{\text{HiddenStates}}$: The top performing model using more than just the last hidden state layer was concatenating the the top 4 layers [CLS] tokens. I tested out the predictive power of each layer's [CLS] token individually and found that the higher up layers were stronger, which is in line with Tenney et al. (2019) who found that higher layers capture semantics of a sentence vs lower layers capturing sentence syntax. Concatenating the top layers outperformed the last hidden layer output in the development set (results on development set can be found in the Appendix 7 but had similar accuracy on the test set, showing little improvement from adding additional [CLS] token embeddings to the dense layer input.

BERT$_{\text{+CNN}}$: The top performing model overall edged out the last hidden state [CLS] token model was taking the entire sequence output and running that through a CNN. Since this achieved the highest accuracy, it is worth exploring further in future work.

Image Caption Fine Tuning: The concept of fine tuning BERT on image captions was not successful, yielding slightly worse accuracy than just regular BERT on most models. These unexpected results may be due to how wide of a spectrum common sense covers. Our labeled data set contained 10,000 examples of a wide variety of topics from animals, to occupations to traveling. Even though there were nearly 600,000 image captions in the COCO data set, they likely didn't cover nearly all the common sense needed to correctly categorize the sentence pairs.

4.1 Error Analysis

Error analysis was performed on the top model to try and diagnose what the model was correctly and incorrectly predicting. Since the sentence pairs are crafted to be similar, with just a small difference that causes one to not make sense, one theory was that model was having trouble parsing out the sentences that were most similar. To test that theory, each sentence with the Universal Sentence Encoder (USE) which is a tool used to capture semantic similarity (Yang et al., 2019) and measure how close each pair of sentences was to each other. Figure 3 shows the distribution, with the vast majority of sentences being categorized as very similar.

There was a small but noticeable difference between the sentences the model categorized correctly and incorrectly - the distribution of the correct sentences skewed a little further left with a median semantic similarity of 0.74 while the incorrectly categorized sentence pairs had a median semantic similarity score of 0.78. This disparity aligns with the theory that the model is having a harder time picking out a sentence that doesn't make sense from sentences that are semantically similar. Table 3 displays two sentences pairs with very high semantic similarity that the best model was unable to correctly classify.

Sent 1	I took three Tylenol tablets to cure my headache.✓
Sent 2	**I took 300 Tylenol tablets to cure my headache.✗**

True Label	Pred. Label	Sem. Sim.
1	0	0.94

Sent 1	To get to the top of the building you may use an elevator.✓
Sent 2	**To get to the top of the building you must use an elevator.✗**

True Label	Pred. Label	Sem. Sim.
1	0	0.97

Table 3: Incorrectly classified sentence pairs with high semantic similarity where ✗ indicates the sentence that does not make sense.

5 Next Steps

Neither approach in this paper achieved competitive accuracy with the subtask A leaderboard. A high volume of image captions did not contain the

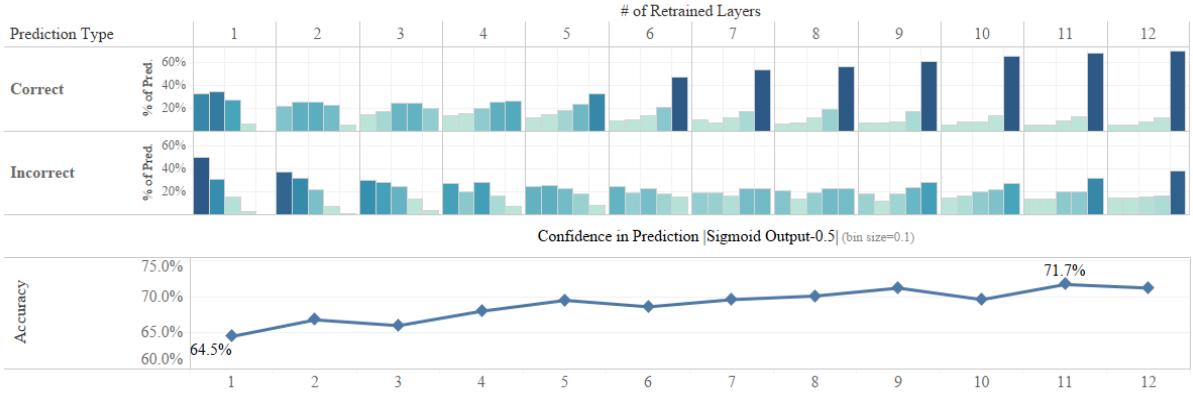

Figure 2: Model confidence and accuracy in correct and incorrect classifications by number of retrained layers.

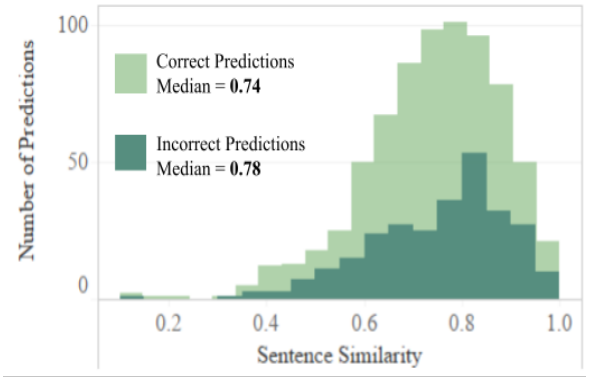

Figure 3: Sentence similarity distributions across correct and incorrect predictions of top performing model.

expansive knowledge needed for a machine to have "common sense", so I believe future work should focus on knowledge graphs and injecting that knowledge into the model in some other way. The winning SemEval paper used a knowledge graph and achieved 97.0% (Zhang et al., 2020) accuracy on the task. Figuring out how to get that amount of information to be learned with less compute power would be a big next step. In addition, I found that the models were having a harder time classifying sentence pairs that were more semantically similar. Since the two sentences in each pair tend to differ by only a word or phrase, a next step could be to isolate the differences in the sentences and weigh the most important words more within the model.

6 References

References

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/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.

Ali Fadel, Mahmoud Al-Ayyoub, and Erik Cambria. 2020. JUSTers at SemEval-2020 task 4: Evaluating transformer models against commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 535–542, Barcelona (online). International Committee for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019. K-BERT: enabling language representation with knowledge graph. *CoRR*, abs/1909.07606.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321, Barcelona (online). International Committee for Computational Linguistics.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual universal sentence encoder for semantic retrieval.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. *CoRR*, abs/1808.05326.

Yice Zhang, Jiaxuan Lin, Yang Fan, Peng Jin, Yuanchao Liu, and Bingquan Liu. 2020. CN-HIT-IT.NLP at SemEval-2020 task 4: Enhanced language representation with multiple knowledge triples. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 494–500, Barcelona (online). International Committee for Computational Linguistics.

7 Appendix

Baseline$_{\text{(Word2Vec)}}$	**61.5**	
Model	**BERT Accuracy**	**BERT$_{+\text{COCO}}$ Accuracy**
BERT$_{\text{CLS-Frozen}}$	62.2	**64.1**
BERT$_{\text{CLS}}$	72.8	**73.8**
BERT$_{\text{HiddenStates}}$	**74.1**	73.8
BERT$_{+\text{CNN}}$	**75.3**	71.4
Current Leader	**97.0**	
Human	**99.1**	

Table 4: Development set accuracy results.

Parameter	**Value**
Learning Rate	$5e-6$
Batch Size	16
Dropout Rate	0.1
Epsilon	$1e-6$
Epochs	5
Retrained Layers	10
Max Token Length	80

Table 5: Parameters used in final models.

Number of Retrained Layers	**Accuracy**
1	64.5
2	66.8
3	66.0
4	68.0
5	69.5
6	68.6
7	69.9
8	70.1
9	71.2
10	69.6
11	71.7
12	71.2

Table 6: Accuracy based on number of retrained layers.

Output Embedding	**Accuracy**
Layer 1 *[CLS]*	49.2
Layer 2 *[CLS]*	53.1
Layer 3 *[CLS]*	53.2
Layer 4 *[CLS]*	60.3
Layer 5 *[CLS]*	64.7
Layer 6 *[CLS]*	67.8
Layer 7 *[CLS]*	68.4
Layer 8 *[CLS]*	66.6
Layer 9 *[CLS]*	68.9
Layer 10 *[CLS]*	69.8
Layer 11 *[CLS]*	67.9
Layer 12 *[CLS]*	71.5

Table 7: Accuracy when using each of BERT's layers' [CLS] output embedding for classification.