DeBERTa Model Fine-tuned for Commonsense Validation and Explanation

Chenxi Lu
School of Information
University of California, Berkeley
luchenxi@berkeley.edu

Erkan Tas
School of Information
University of California, Berkeley
erkan@berkeley.edu

Abstract

In this paper, we explore how pre-trained DeBERTa models can be fine-tuned on SemEval-2020 Task 4, **Com**monsense **V**alidation and **E**xplanation (**ComVE**). ComVE includes three subtasks, and we introduce our systems for the first two subtasks: Subtask A is to distinguish a natural language statement that makes sense from one that does not, and Subtask B is to provide reasonable explanations for why the false statement does not make sense. We achieved an accuracy rate of 87.8% for Subtask A and 89.7% for Subtask B.

No team competing in the contest has ever used DeBERTa model for the tasks. More importantly, since DeBERTa model does not have a built-in capability for directly solving multiple choice questions, our paper describes our approach to develop the multiple choice capability on top of DeBERTa, and fine-tune it on ComVE dataset. Our model out-performed the fine-tuned BertForMultipleChoice model. Our accuracy rate for Subtask B would have ranked us at top 10 among all submissions that did not resort to external knowledge base to complement their pretrained language models.

1 Introduction

Human-analogous natural language understanding (NLU) is a grand challenge in artificial intelligence. Improving common sense for deep learning applications has become one of the primary research goals in NLP space as it's essential to achieve realistic and scalable implementation of NN-backed solutions, particularly for NLU purposes.

Humans have commonsense knowledge, which means they have basic level of knowledge and reasoning concerning everyday situation and events. However, machines lack this kind of background knowledge. For example, without knowledge from daily life, a machine would find it extremely difficult to understand that the sentence *'I like to ride my chocolate'* is against common sense because *'Chocolate is food, not a transportation unit'*. Commonsense knowledge is assumed to be commonly shared among most people, so people would naturally omit the explanation when communicating with others. This adds more challenges for NLU systems to acquire the knowledge and to reason beyond the limited data.

SemEval-2020 Task 4, Commonsense Validation and Explanation (**ComVE**) (Wang et al., 2020) includes three subtasks. Subtask A is a validation task that asks a system to choose the statement that is against common sense from a pair of given statements. Subtask B is a multiple choice explanation task that asks a system to choose the correct reason, out of three choices, that best explains the rationale behind the the false statement. Subtask C is an explanation generation task that asks a system to generate the reason why a false statement does not make sense. The Subtasks A and B are evaluated using accuracy, and Subtask C is evaluated with the BLEU score. Since this paper works on the first two subtasks, we illustrate some examples of the subtasks A and B in Table 1.

Our paper focus on how the Transformer-based DeBERTa model (He et al., 2021) can be fine-tuned on ComVE, without using further textual corpus. The Transformer is so far the most effective and trending architecture for neural language modeling. Transformers apply self-attention to draw global dependencies between input and output (Vaswani et al., 2017). It allows for significantly more parallelization than recurrent neural networks (RNNs).

The rise of large-scale Transformer-based Pretrained Language Models (PLMs) has motivated researchers to fine-tune these algorithms in many downstream NLP tasks. We choose to fine-tune DeBERTa on ComVE because it is a relatively new and powerful algorithm and no team competing in SemEval-2020 has ever used DeBERTa before. In

Task	Examples	
Subtask A	S0:	My cousin throws the snowball to my brother.
	S1:	My cousin throws the house to my brother.
Subtask B	False Sentence:	My cousin throws the house to my brother.
	Option A:	Houses are usually expensive.
	Option B:	A house is too big to throw.
	Option C:	There are several houses on this street.

Table 1: Examples from the ComVE dataset.

Section 2, we will briefly introduce the models.

The paper is organized as follows: Section 2 reviews some models and important works in the commonsense domain. Section 3 describes the task and datasets. Section 4 lays out our fine-tuned DeBERTa system for ComVE tasks. Section 5 discusses the experiments and results. Section 6 concludes our research project.

2 Background

We approach the commonsense tasks by applying pre-trained language models via transfer learning. Our goal is to fine-tune the DeBERTa algorithm on ComVE datasets. Our baseline model is BERT algorithm (Devlin et al., 2019). Therefore, we first briefly discuss the two pretrained language models used in our work. Next, we review some related work in SemEval-2020 Task 4 that inspired us.

2.1 Overview of Models

BERT (Devlin et al., 2019): Bidirectional Encoder Representations from Transformers, is a deep bidirectional transformer model that produces context representations. It is trained on both Masked Language Modeling (MLM) and Next Sentence Prediction (NSP): (1) In MLM, 15% of tokens are replaced by `[MASK]` token in order to create a noised version of input; (2) In NSP, the model predicts whether a sentence from the training data follows from the other. BERT model can be fine-tuned for a wide range of tasks.

DeBERTa (He et al., 2021): Decoding-enhanced BERT with disentangled attention, is an improvement over the BERT and RoBERTa (Liu et al., 2019) models using two novel techniques: (1) The disentangled attention mechanism, where each word is represented using two vectors that encode its content and position, respectively, and the attention weights among words are computed using disentangled matrices on their contents and relative positions, respectively; (2) Enhanced mask decoder, which incorporates absolute positions in the decoding layer to predict the masked tokens when pre-training the model. DeBERTa has significantly improved the efficiency of model pre-training and has advanced state-of-the art performance on many NLU downstream tasks.

2.2 Relevant Work

Our paper benefited from participants in SemEval-2020 Task 4. There are a total of 39 valid submissions for Subtask A and 27 submissions for Subtask B. Most top performers adopted PLMs such as BERT and RoBERTa. Among all valid submissions, the top accuracy is 97.0% and 95.0% for Subtasks A and B, respectively.

Solomon team (Srivastava et al., 2020) mainly uses BERT, ALBERT, and RoBERTa as the encoder. For Subtask A, they prepare the dataset by splitting a given input into two separate sentences and each being passed through the model to generate probability score for unreasonability. They add a score comparison system on top of the model to predict the final label for the sentence pair. For Subtask B, they formulated the task as a three-way binary classification problem. Each input is a concatenation of the false sentence, a connecting phrase, and one of the three choices. The best results their system achieved were 96.0% and 94.0% for Subtask A and B, respectively. These are the highest scores among the participants who use no external knowledge data base to complement large-scale pre-trained language models.

IIE-NLP-NUT team (Xing et al., 2020) proposes input reconstruction strategy with prompt templates, which allow them to effectively formalize the the subtasks into multiple-choice question answering format. They mainly uses RoBERTa as the encoder, but also complement it with external textual corpus. Their approach achieved an accuracy of 96.4% for Subtask A and 94.3% for Subtask B.

Data	Label 0	Label 1	Total
Train	4,979	5,021	10,000
Dev	518	479	997
Test			1,000

Table 2: Label distribution of Subtask A: Validation.

Data	A	B	C	Total
Train	3,195	3,362	3,443	10,000
Dev	344	327	326	997
Test				1,000

Table 3: Label distribution of Subtask B: Explanation.

3 Task and Data

Subtask A is a binary classification problem. The system needs to choose from two natural language statements with similar wordings which one does not make sense. The training data has 10,000 sentence pairs, sentence 0 (S0) and sentence 1 (S1), in the training data. Each instance is labeled as either 0 or 1, representing that S0 or S1 is against common sense.

Subtask B is a multiple-choice problem. The system needs to choose the right explanation from three choices explaining why the given false sentence does not make sense. The training data has 10,000 false sentences, each with three choices. Three out of 10,000 instances have only two choices. There is only one correct explanation for each instance.

We present the data distribution for Subtasks A and B in Table 2 and Table 3. We will describe the data processing along with model descriptions in the Methods Section since the data format needs to fit the model.

4 Methods

4.1 Baseline Models

Since most top performers in SemEval-2020 Task 4 adopted Transformer-based models, we use BERT as baseline for both subtasks. The BERT$_{\texttt{base-uncased}}$ model contains 12 layers, 12 attention heads, 768 hidden state size, and 110 million model parameters.

For Subtask A, the binary classification problem, we use BertForSequenceClassification model (*i.e.* BertModel$_{\texttt{base-uncased}}$ with a linear layer on top of the pooled output.) fine-tuned on ComVE-SubA dataset as our baseline model. The model is the BertModel$_{\texttt{base-uncased}}$ with a linear layer on top of the pooled output.

For Subtask B, the multiple choice problem, we use BertForMultipleChoice model fine-tuned on ComVE-SubB dataset as our baseline model. It is the BertModel$_{\texttt{base-uncased}}$ with a linear layer on top of the pooled output, and finally adding a softmax layer. We choose the model because it has successfully fine-tuned on the Situations with Adversarial Generations (SWAG), a large-scale dataset with multiple choice questions about a grounded commonsense inference (Zellers et al., 2018). Devlin et al. (2019) showed that BERT$_{\texttt{large}}$ achieved a test accuracy of 86.3% on SWAG task.

4.2 Approach for Sense-making Validation

For Subtask A, we send the sentence pairs {S$_0$, S$_1$} as a list into the tokenizer, resulting in the following format: `[CLS]` S$_0$ `[SEP]` S$_1$ `[SEP]`. Using the example in Table 1, the format would be:

> `[CLS]` *my cousin throws the snowball to my brother.* `[SEP]` *my cousin throws the house to my brother.* `[SEP]`

We apply DeBERTa model for Subtask A, a binary classification task. The DeBERTa$_{\texttt{base}}$ model contains 12 layers, 12 attention heads, 768 hidden state size, and 140 million model parameters. The DebertaForSequenceClassification$_{\texttt{base}}$ model is DeBERTa with a linear layer on top of the pooled output, so it can be used for a sequence classification task. We fine-tune the model on ComVE Sub-A dataset.

4.3 Approach for Explanation Selection

Subtask B is much more challenging to apply DeBERTa. Because DeBERTa model does not have a built-in capability for directly solving multiple choice questions. Unlike BERT has the BertForMultipleChoice model, DeBERTa does not have a similar model. Therefore, we need to develop the multiple choice capability on top of DeBERTa, and fine-tune it on the ComVE Sub-B dataset.

For Subtask B, following the *Solomon* team's approach (Srivastava et al., 2020), we first formulate the task as a three-way binary classification dataset. The four key elements in the original task dataset are originally {S, O$_A$, O$_B$, O$_C$}, where *S* stands for the false sentence, and *O* stands for the option. We first convert the dataset to be sentence-option pairs, *i.e.* {S, O$_A$}, {S, O$_B$}, and {S, O$_C$}. This essentially triples the number of rows in the dataset by

Subtask	Input	Output
Subtask A: Validation	[CLS] S0 [SEP] S1 [SEP]	binary [0,1]
Subtask B: Explanation (paired)	[CLS] S [SEP] OA [SEP]	integer in range [0,2],
	[CLS] S [SEP] OB [SEP]	(where 0,1,2 represent
	[CLS] S [SEP] OC [SEP]	A, B, C, respectively.)
Subtask B: Explanation (concatenated)	[CLS] S *phrase* OA [SEP]	integer in range [0,2],
	[CLS] S *phrase* OB [SEP]	(where 0,1,2 represent
	[CLS] S *phrase* OC [SEP]	A, B, C, respectively.)

Table 4: Format of one slice of input and label. (phrase = *'No, this does not make sense because'*)

breaking each row into three rows. We then group every three rows together before sending into the tokenizer. The format of the example shown in Table 1 would be as follows:

> [CLS] *my cousin throws the house to my brother.* [SEP] *houses are usually expensive.* [SEP]
>
> [CLS] *my cousin throws the house to my brother.* [SEP] *a house is too big to throw.* [SEP]
>
> [CLS] *my cousin throws the house to my brother.* [SEP] *there are several houses on this street.* [SEP]

After we group each input as three sequences {S, OA}, {S, OB}, and {S, OC}, we then convert labels to an integer in the range of [0,2], indicating whether the gold answer should be *A*, *B*, or *C*.

In addition, we also try the *IIE-NLP-NUT* team's prompt template approach because they argue that the direct concatenation of the false statement and each candidate explanation could distract the model (Xing et al., 2020). We add a connecting phrase *'No, this does not make sense because'* between each pair of false sentence and option so that the model would possibly be aware of the causal relationship between the first and second sentences (*i.e.* the false sentence and the option of explanation). Thus, the format of the above {S, OB} would look like below, which is now a single sentence instead of a pair of sentences:

> [CLS] *my cousin throws the house to my brother.* ***no, this does not make sense because*** *a house is too big to throw.* [SEP]

The format of {S, OA} and {S, OC} would be the same.

Because Subtask B is a multiple choice problem, similar to SWAG task (Zellers et al., 2018), we need a model that is suitable for picking a correct answer from multiple choices. Borrowing a similar architecture from the building of BertForMultipleChoice from BertModel, we create our version of DebertaForMultipleChoice; we build the model by first adding a linear layer on top of the DeBERTa model's pooled output, and finally adding a softmax layer.

Here is the step-by-step description of our approach:

- Assuming a batch size of b and the sequence length of len, each ComVE-SubB example has 3 sequences, one correct and two incorrect.

- For each example, the the input id has a size of [3,len], same for the attention mask id and token type ids. And the label for each example is an integer within [0,2].

- After batching, the model would get an input of shape [b,3,len]. We need to first reshape them to [3b, len] before passing DeBERTa because the model is only capable of considering all of them independently.

- Then we compute a raw unnormalized score for each independent sequence by adding a pooling layer. The scores would have a size of [3b] as well.

- Finally, we reshape the scores of size [3b] back to [b, 3] and then apply the cross entropy loss to train the model. We compute the softmax for each sequence and predict the answer.

The input and output for both subtasks are summarized in Table 4.

Model	Accuracy
BERT-SC	82.3
DeBERTa-SC	**87.8**

Table 5: Results for Subtask A on Test.

Model	Accuracy
BERT-MC-pair	84.0
DeBERTa-MC-connect	89.4
DeBERTa-MC-pair	**89.7**

Table 6: Results for Subtask B on Test.

5 Results and Discussions

5.1 Subtask A

Table 5 shows the results of the systems experimented for Subtask A. Our baseline BertForSequenceClassification `base-uncased` model (referred as BERT-SC in Table 5) has an accuracy of 83.0%. For DeBERTA, we fine-tuned DebertaForSequenceClassification`base` model (*i.e.* DeBERTa-SC) with learning rate of 2e-5 and a batch size of 32 for 4 epochs. Our fine-tuned DeBERTa model achieved an accuracy of 87.8%.

5.2 Subtask B

Table 6 shows the results of the systems experimented for Subtask B. Our baseline BertForMultipleChoice `base-uncased` model (referred as BERT-MC) has an accuracy of 84.0%. For DeBERTA, we try both the paired version and the template prompt version of data pre-processing, and we fine-tune our enhanced DeBERTa `base` model (referred as DeBERTa-MC) with learning rate of 1e-5, dropout probability of 0.1, and a batch size of 24 for 5 epochs. Our paired model achieved an accuracy of 89.7%, which out-performed the BERT baseline model by 5.7%.

Our accuracy rate for Subtask B would have ranked us at top 10 among all submissions that did not resort to external knowledge base to complement their pre-trained language models. As Wang et al. (2020) pointed out in the task summary paper, both ConceptNet and OMCS corpora are used as references for the annotator to write the data instances in the data creation stage, many of the highest-accuracy submissions may indicate data leaking to some extent due to the use of relevant external knowledge bases.

5.3 Discussion

In Subtask B, we compared two versions of data preprocessing: one is in the form of paired sequence as `[CLS]` S `[SEP]` O `[SEP]`, and the other utilizing the prompt template so that it becomes a single sentence `[CLS]` S + *connecting phrase* + O `[SEP]`. Our experiments show that adding prompt template did not help the model to predict better. This is probably because DeBERTa model is already quite good at recognizing the goal of the task through fine-tuning. The disentangled attention mechanism may help with this.

6 Future Work

First, due to the time and computing resource limit, we are using the base versions of both BERT and DeBERTa in this paper. The base versions have significantly fewer parameters than the largest versions of both models. Although the base versions have already produced satisfying accuracy rates, we do hope to apply the same approach on the larger models and see how much better the results could be.

Second, since PLMs have learned common sense through pre-training, we can have more variants of models and probe the commonsense knowledge within PLMs.

7 Conclusion

In this paper, we explore how pre-trained DeBERTa models can be fine-tuned on SemEval-2020 Task 4 on Commonsense Validation and Explanation. We achieved an accuracy rate of 87.8% for Subtask A and 89.7% for Subtask B. Our main contributions are:

- It is the first time that the newly-developed DeBERTa model is applied to ComVE datasets. And we found that it out-performed the BERT model by more than 5% in both subtasks.

- We have developed, described, and tested an approach that empowers the DeBERTa model to easily solve multiple choice problems with high accuracy.

Acknowledgments

This paper benefited from many inspiring discussions. We gratefully acknowledge Mark Butler, Joachim Rahmfeld, Yifang Wei, Jiyun Luo, and Jie Zhou for their guidance and feedback. We also thank the W266 faculty team and participants of the NLP study group at UC Berkeley Master of Information and Data Science Program. The authors are responsible for any errors that remain.

References

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

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.

Vertika Srivastava, Sudeep Kumar Sahoo, Yeon Hyang Kim, Rohit R.r, Mayank Raj, and Ajay Jaiswal. 2020. Team Solomon at SemEval-2020 task 4: Be reasonable: Exploiting large-scale language models for commonsense reasoning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 585–593, Barcelona (online). International Committee for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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.

Luxi Xing, Yuqiang Xie, Yue Hu, and Wei Peng. 2020. IIE-NLP-NUT at SemEval-2020 task 4: Guiding PLM with prompt template reconstruction strategy for ComVE. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 346–353, Barcelona (online). International Committee for Computational Linguistics.

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