An ALBERT-based approach for Commonsense Validation

Zhengxian Fan
University College London
zhengxian.fan.17@ucl.ac.uk

Abstract

BERT, RoBERTa, ALBERT have achieved remarkable results on a range of NLP tasks by fine-tuning the aggregate representation on task-specific datasets. However, they do not always lead to vast improvements, especially for small datasets. In this paper, we propose a novel and effective ALBERT-based approach for Commonsense Validation; a commonsense reasoning task tested by sentence-pair classification. Instead of fine-tuning the sequence representations, we utilize Mask Language Modelling (MLM) loss generated by ALBERT for the classification. We first fine-tune the encoder on commonsense knowledge bases to improve its commonsense reasoning ability and then train the encoder and a classifier together on the training set. Our approach achieves 92.2% on the official test set and significantly outperforms baselines.

1 Introduction

Commonsense Validation (Wang et al., 2019) is a recently introduced task that aims to evaluate a model's sense-making ability. Before this, there have been many open domain commonsense reasoning tasks such as Choice of Plausible Alternatives (COPA) (S.Gordon et al., 2012), Situations With Adversarial Generations (SWAG) (Zellers et al., 2018), CommonsenseQA (Talmor et al., 2018). COPA and SWAG emphasize commonsense causal reasoning. Questions of COPA ask for the most plausible cause of a given premise. For example, *"Premise: I tipped the bottle. What happened a RESULT?" "Alternative 1: The liquid in the bottle froze." "Alternative 2: The liquid in the bottle poured out."*. SWAG is inspired by COPA, choosing the correct one from four different endings of a given situation. CommonsenseQA tests commonsense knowledge through question answering; a system is required to select the correct answer of a question from four distractors. Unlike the tasks mentioned above, the Commonsense Validation task is to select the statement which makes more sense from two statements, as shown in Table 1.

One main challenge in this task is that all sentence pairs have similar wordings. In the development set, the average number of words that are present in one statement but not in another is 2.57, and the average length of true statements is the same as false statements. Additionally, negative words (e.g., 'not', 'no', 'can't') are evenly distributed between true and false statements, so a system can not make correct predictions according to some of the words only.

A: I have a lamp on my desk. (T)
B: I have a desk on my lamp. (F)
A: He put a turkey into the fridge. (T)
B: He put an elephant into the fridge. (F)

Table 1: Samples of the Commonsense Validation Task

For a sentence-pair classification problem like SWAG, the pre-trained language model BERT (Devlin et al., 2019) outperforms humans after fine-tuning on the SWAG dataset. The system utilizes the final hidden vectors of the classification tokens (the first token outputs by the encoder [CLS]) as aggregate sequence representations for the classification. However, the true statements in Commonsense Validation have similar wordings as false statements. This makes their sequence representations also similar and hard to differentiate. Moreover, the provided training set is small compared with SWAG, 8k, and 37k, respectively, so the same fine-tuning approach can not reproduce similar improvements.

In this paper, we use the new state-of-the-art language model ALBERT (A Lite BERT) (Lan et al., 2019) for feature extraction, which significantly re-

duces the number of parameters required for BERT to generate similar results. It breaks the embedding matrix to two smaller matrices $V \times E$ and $E \times H$ so that the dimension of E does not grow with the hidden size H. Another key improvement is to replace the NSP (Next Sentence Prediction) with SOP (Sentence Order Prediction) because pre-training on NSP is too weak compared with SOP. Our research focused on the following two aspects:

- Enriching the pre-trained ALBERT with commonsense inference ability
- Discovering discriminating features that represent sentence semantics

We first fine-tune the ALBERT encoder on a dataset constructed from two commonsense knowledge bases, improving the performance of pre-trained ALBERT by 9.2%. Secondly, we find that the pre-trained ALBERT always outputs low prediction loss for bigrams with high frequency without considering the sentence meaning. We alleviate this problem by adding frequencies of words as additional inputs to the classifier, which gains a further 8.9% improvement.

2 Methods

In this section, we describe our approach and intuitions in detail.

2.1 Masked Language Modeling

ALBERT is pre-trained with Masked Language Modeling (MLM) objective, masking target words, and then predicting the missing words, on BooksCorpus and English Wikipedia. For a masked token at the i_{th} position of the input sequence, the final hidden vector of the encoder output W_i^o is multiplied by an embedding matrix W^v after a non-linear transformation and then mapped to a probability distribution over the vocabulary by applying the softmax function. The probability that the masked token is the j_{th} word of the vocabulary is represented as $P(X = j)_i$, and the prediction loss is the negative log of the probability.

$$P(X = j)_i = \text{softmax}_i(\text{norm}(W_i^o W^e)W^v + b_v)$$

$$loss_i = -\log(P(X = label)_i)$$

Where $W^e \in \mathbb{R}^{d_{output} \times d_{emb}}$, $W^v \in \mathbb{R}^{d_{emb} \times d_{voc}}$ are projection matrices, and the norm represents layer normalization.

In (Petroni et al., 2019), BERT performs well on recovering factual knowledge and commonsense knowledge by masking the objects of factual sentences and letting the model predict them (e.g., Washington is the capital of [MASK]). Because ALBERT can store knowledge into parameters, sentences with high MLM loss are likely to against commonsense knowledge. We mask every word in a sentence sequentially and use the average MLM loss of words as the loss of a sentence. For the commonsense validation task, the statement with lower loss is marked as the true statement. We evaluate four pre-trained ALBERT models with different configurations on the Commonsense Validation dataset containing 2021 instances. The statistics of them are shown in table 2. The pre-trained ALBERT-xxlarge produces the best result; thus we fine-tune it only in the following experiment.

Model	Parameters	Accuracy
ALBERT$_{base}$	12M	65%
ALBERT$_{large}$	18M	67%
ALBERT$_{xlarge}$	60M	71%
ALBERT$_{xxlarge}$	235M	73%

Table 2: Performance of pre-trained ALBERT models with different size

2.2 Bigram Frequency

In the previous subsection, we determine which statement makes more sense by comparing the average MLM loss. However, we find two special cases that the average loss incorrectly scores a statement. The first usually occurs when two statements have different lengths. Extra words in the longer statement will lower the average loss if they are uninformative. For example, *'the sky is blue', 'the sky is likely to be white'* are two input statements, the pre-trained model outputs a lower average loss for the second statement because the MLM loss of *'to', be'* is extremely low.

When two sentences are the same length, using the average MLM loss solely is still not a good method. We observe for a previously unseen input sequence, ALBERT unavoidably outputs a relatively low loss for common collocations in the sequence regardless of how logical the sentence is. For the two samples in Table 3, the loss of *'to fly'* is higher than *'to speak'* in both pairs of statements, although the two samples are mutually contradictory. We hypothesize that ALBERT is not sensitive

Sent1:	humans	can	-	learn	to	**fly**	-
Loss:	6.71	2.15	-	0.050	0.29	**7.90**	**Avg:3.42**
Sent2:	humans	can	-	learn	to	**speak**	-
Loss:	6.76	1.26	-	0.13	0.047	**6.82**	**Avg:3.00**
Sent1:	humans	can	not	learn	to	**fly**	-
Loss:	6.85	3.51	2.05	1.22	0.093	**8.00**	**Avg:3.62**
Sent2:	humans	can	not	learn	to	**speak**	-
Loss:	7.11	3.42	2.29	1.57	0.16	**7.09**	**Avg:3.61**

Table 3: An example of the MLM prediction affected by common collocations

to negations or small changes in sentences, and *'to speak'* has a much higher occurrence than *'to fly'* in the corpus, so it always outputs a lower loss for *'to speak'*. The bigram frequency of *'to speak'* in the BookCorpus is 951045 provided by (Michel et al., 2010), where *'to fly'* is only 129228. If we replace *'to speak'* in the first sample with significantly less frequent bigrams such as *'to swim'* (frequency: 48676), *'to analyze'* (frequency: 40536), the model also incorrectly output a higher loss than *'to fly'*. We test on 100 pairs of statements similar to the example, and the pre-trained ALBERT only achieves 59.5% accuracy by comparing the MLM loss of two statements only.

Both cases mentioned above are related to the frequency of words. For the first case, uninformative words with low MLM loss are usually highly frequent in corpora. In the second case, more frequent bigrams are more likely to have a low loss. We use the average of the frequency of every bigram in a sentence as the frequency of the sentence f_s. Table 4 shows the accuracy of the ALBERT model drops when the f_s difference increases. To find a balance between MLM loss and f_s, we propose to apply a classifier taking loss difference and f_s difference as inputs. We fine-tune the encoder and train the classifier together to utilize the training set effectively. Training details are provided in the next section.

Accuracy:(%)	73.9	71.8	70.8	70.1	69.0	68.7
$\lvert f_{s0} - f_{s1} \rvert$:($\geq$)	0	20k	40k	60k	80k	100k

Table 4: Accuracy against the absolute value of f_s difference in the development set

3 Experiments

3.1 Encoder Fine-tuning

The ALBERT-xxlarge encoder is a bi-directional transformer with 12 blocks, responsible for generating contextualized word embeddings. To store more commonsense knowledge into the parameters, we further fine-tune the pre-trained encoder on MLM and SOP over ConceptNet (Speer et al., 2017) and Event2Mind (Rashkin et al., 2018). ConceptNet is a multilingual knowledge graph containing over 21 million edges and 8 million nodes. It represents assertions as triples of the form (subject, relation, object). For example, the assertion *"birds can fly"* is represented as (bird, CapableOf, fly). Event2Mind contains 25000 event phrases with commonsense inferences on event participants' mental states such as intentions and emotions. They both contain knowledge beyond BooksCorpus and English Wikipedia.

We filter all triples containing non-English vocabularies and uninformative relations such as *'IsA', 'RelatedTo', 'FormOf' (Inflected Forms)*. After that, we remove triples with vocabularies that cannot be tokenized to 3 or fewer sub-tokens by SentencePiece tokenizer because they are generally knowledge about technical terms like chemical compounds which are beyond commonsense. During fine-tuning, we only expect the encoder to store commonsense knowledge rather than linguistic knowledge, so we use synthetic sentences as training samples instead of real sentences collected by distant supervision or other approaches. We convert camel-cased relations to phrases (e.g., 'UsedFor' to 'is used for', 'HasProperty' to 'is') and concatenate them with subjects and objects to form sentences and finally get 247,326 sentences from ConceptNet. For Event2Mind, we get 180,542 positive samples of NSO by concatenating the events with the intentions and using the emotions as the next sentences of them. The average length of constructed sentences is 7.01, so we set the maximum sequence length to 16 and maximum masked words per sequence to 3. Finally, we obtain 626,510 training samples to train on MLM and NSO jointly.

We fine-tune the ALBERT-xxlarge encoder on a single cloud TPUv3 with a batch size of 3840. We store and then evaluate the training checkpoints on the development set every 300 steps after 700

steps to avoid over-fitting on the training data because the constructed dataset is relatively small compared with the pre-training corpus. Figure 2 demonstrates that fine-tuning on the dataset improves the task accuracy. We stop at 3000 steps because experimental results show the encoder has already over-fitted after 1000 steps. The accuracy at 1000 training steps is 82.2%, 9.2% higher than the pre-trained ALBERT-xxlarge.

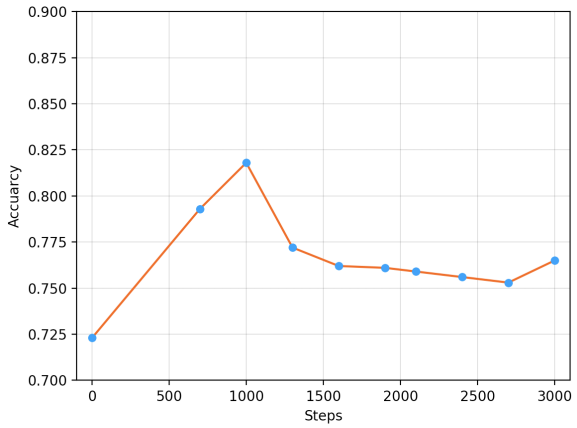

Figure 1: Performance Curve

3.2 Classifier

In the previous section, we propose to use MLM loss and f_s difference together for the classification. We use a MLP with one hidden layer as the classifier (Figure 3); it is first trained individually until convergence with MLM loss data recorded. We then fine-tune the encoder and the classifier on the training dataset because the encoder occasionally generates bad representations which bias the downstream classification task. During the fine-tuning, if the sigmoid cross-entropy loss for a sample is larger than 1, we update the encoder weights by minimizing the average MLM loss for the true statement and maximizing the MLM loss for the false one. The learning rate for the encoder and classifier is 1e-6 and 1e-3, respectively. After training for 3 epochs, the model reaches 99.8% accuracy on the training set and 91.1% accuracy on the development set.

3.3 Evaluation on the Test Set

We evaluate our proposed model and its ablated variants on the official test set with 1000 instances. Results of the evaluation are shown in Table 5. BERT, ELMO, fine-tuned ELMO are baselines provided in (Wang et al., 2019). By comparing with the baselines, our fine-tuning approach significantly

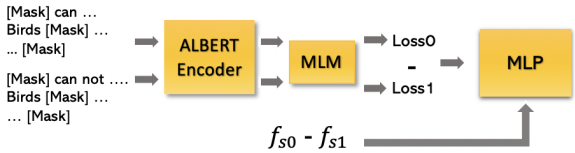

Figure 2: Model architecture

improves the commonsense inference ability of the model. After applying the classifier, the fine-tune ALBERT-xxlarge model improves by 9.9% on the test set, indicating that f_s is a discriminating feature for the classification. Moreover, the ablation study shows that both encoder fine-tuning and the additional classifier contribute to accuracy.

Model	Dev	Test
ELMO	69.4%	-
BERT	70.1%	-
fine-tuned ELMO	74.1%	-
ALBERT$_{xxlarge}$	73.0%	73.5%
$+encoder\ fine\text{-}tuning$	82.2%	82.3%
$+cls$	80.8%	83.9%
$+fine\text{-}tuning + cls$	**91.1%**	**92.2%**
Human	99.1%	-

Table 5: Comparison with baselines, the first group are experimental results in (Wang et al., 2019), the second group are results of ALBERT-based approaches.

4 Conclusion

In this paper, we propose an ALBERT-based approach for Commonsense Validation, which is efficient and effective compared with fine-tuning the classification token. Experiment results show that fine-tuning over ConceptNet and Event2Mind significantly improves the commonsense reasoning ability of the model. We opensource the training checkpoint for the fine-tuned ALBERT-xxlarge; it might benefit other commonsense inference tasks like CommonsenseQA. We also find the weakness of MLM loss predictions; highly frequent combinations get low prediction loss even if they are unreasonable in the context. Using f_s as an input feature significantly improves the result of Commonsense Validation.

Acknowledgments

We thank all reviewers for their valuable comments.

References

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

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *arXiv preprint arXiv:1909.11942*.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, William Brockman, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, , and Erez Lieberman Aiden. 2010. Quantitative analysis of culture using millions of digitized books. *Science*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP-IJCNLP 2019*.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. In *arXiv preprint arXiv:1805.06939*.

Andrew S.Gordon, Zornitsa Kozareva, , and Melissa Roemmele. 2012. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *SemEval 2012*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *AAAI 2017*.

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

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *ACL 2019*.

Chinmay Singh Yash Jain. 2019. Karna at coin shared task 1: Bidirectional encoder representations from transformers with relational knowledge for machine comprehension with common sense. In *EMNLP 2019*.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. In *arXiv preprint arXiv:1908.06725*.

Naman Goyal Jingfei Du Mandar Joshi Danqi Chen Omer Levy Mike Lewis Luke Zettlemoyer Veselin Stoyanov Yinhan Liu, Myle Ott. 2019. Roberta: A robustly optimized bert pretraining approach. In *NAACL 2019*.

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