Exploring Pre-trained Models for Common Sense Validation: A Study on Sequence Classification and Multiple Choice Tasks

19307110429 Wang Siyu
20307110462 Wang Wanyi
19307110530 Yang Leiting

1 Introduction

Natural Language Processing (NLP) has gained attention in recent years and sparked discussions since the public release of ChatGPT. However, models trained on large corpora do not effectively grasp common sense or make judgments based on it. This is primarily due to the challenge of training the model to learn common sense knowledge, which is akin to an iceberg hidden beneath the corpus's surface(Wang et al., 2019).

SemEval-2020 Task 4 offers a well-structured dataset to evaluate the model's ability in terms of common sense verification and interpretation. Task A requires choosing between two natural language statements to determine which one contradicts common sense.

Question: Which statement of the two is against commonsense?
Sentence A: He put a turkey into the fridge. Sentence B: He put an elephant into the fridge (against commonsense).

Table 1: An example of an option in Task A

The difficulties of this task include:

1. Violations of common sense are often syntactically structured in statements, but their meaning does not align with common sense knowledge. 2. The challenge lies in identifying specific indicators to detect common sense. 3. Finding a clear and direct indicator of the model's ability to employ common sense is challenging.

To address these challenges, we consulted relevant literature. One proposed solution is to incorporate structured knowledge sources like ConceptNet (a knowledge base for common sense reasoning) or unstructured sources like Wikipedia. These sources provide extracted and contextualized knowledge to calculate sentence scores and offer answers(Wang et al., 2019; Lin et al., 2019).

Previous research indicates that pre-trained language models acquire some common sense knowledge from their training on large-scale corpora. Consequently, we can perceive pre-trained language models as not only context encoders but also repositories of implicit common sense knowledge that can be utilized during question and answer sessions(Zhao et al., 2020; Xing et al., 2020).

Building upon the initial ideas presented in this paper, we introduce pre-trained models like Bert, RoBerta, and ALBert, which already possess structured or unstructured knowledge sources. By leveraging the common sense knowledge embedded in these models, we can enhance the judgment process.

2 Approaches

In this paper, we employ two main approaches for NLP: sequence classification and multiple choice. We utilize Bert, RoBerta, and ALBert to implement these approaches. In this section, we provide an introduction to these approaches and models.

2.1 Sequence classification

Sequence classification is an NLP task that involves categorizing input text sequences into predefined categories. The input can be a sentence, a paragraph, or a document, and the model needs to comprehend the semantics and context of the text to classify it appropriately. Common sequence classification tasks include sentiment analysis, spam recognition, text classification, and intention recognition.

To tackle sequence classification tasks, a common approach is to use a pre-trained language model (such as BERT, RoBerta, or ALBERT) as the base model and fine-tune its parameters. Pretraining the language model allows it to learn a comprehensive text representation, which can then be adapted for specific classification targets through fine-tuning.

2.2 Multiple choice

Multiple choice is another NLP task where the model needs to select the most appropriate or correct answer from a given question or descriptive context along with several alternative answers. This task is commonly used in reading comprehension, question answering systems, and linguistic reasoning. The model must understand the context, reason, and analyze to choose the best-fitting answer.

The multiple choice task measures the model's ability to understand semantics, reason, and utilize contextual information. It requires the model to make judgments and select the most suitable option among a set of alternatives.

A multiple-choice based Q model $\mathcal{M}$ consists of a Pre-trained language model (PLM) encoder and a task-specific classification layer which includes a feed-forward neural network $f()$ and a softmax operation.

$$score_i = \frac{\exp\left(f(C^i)\right)}{\sum_{i'} \exp(f(C^{i'}))}, C^i = PLM(inp)$$

The candidate answer which owns a higher score will be identified as the final prediction.
Adjust this subtask into the multiple-choice style QA problem as

$$\hat{y} = argmax_{i\in\{1,2\}} P(S^i|Q)$$

where Q is the additional prompt question, two statements S^1, S^2 are the candidate answers and y stands for the index of the commonsensible statement.

2.3 Transformer

The transformer is a neural network model that learns context and meaning by capturing relationships in sequential data. It relies on attention mechanisms to process sequential data, unlike recurrent or convolutional neural networks. The models mentioned in this paper, Bert, RoBerta, and ALBert, are all based on the transformer architecture(Han et al., 2021).

2.4 BERT

BERT(Devlin et al., 2018) (Bidirectional Encoder Representations from Transformers) is a transformer-based machine learning technique used for various NLP tasks. Its key innovation lies in bidirectional training, which considers the context from both directions when processing each word in a text. Unlike previous models that trained text data in a unidirectional manner, BERT's bidirectional training enables a deeper understanding of the text by capturing the word's meaning within its sentence context(Tenney et al., 2019).

During pre-training, BERT performs two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, a percentage of input tokens are randomly masked, and the model predicts them based on the non-masked tokens' context. NSP involves predicting whether one sentence naturally follows another. After pre-training, BERT can be fine-tuned with an additional output layer to create high-performance models for various tasks(Tenney et al., 2019).

However, BERT's computational cost and extensive training time necessitate more efficient models like RoBerta and ALBert. These models offer similar performance while using fewer parameters.

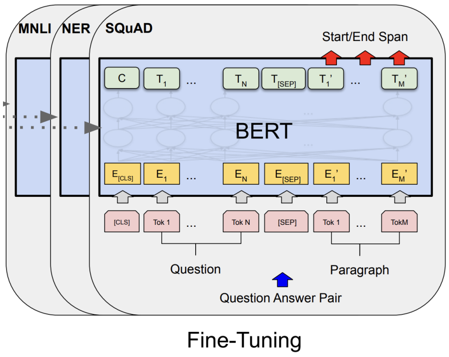

Figure 1: Overall fine-tuning procedures for BERT.(Devlin et al., 2018)

2.5 RoBERTa

RoBERTa is a BERT variant that optimizes performance through several changes made during training. It uses more training data with a larger batch size, employs dynamic masking by re-masking input sequences at the start of each epoch, removes the NSP task, has longer training time and larger sequence length, and utilizes an optimized Byte Pair Encoding (BPE). These improvements enable RoBERTa to achieve better performance compared to BERT across different NLP tasks(Liu et al., 2019).

We also used MNLI, which stands for Multi-Genre Natural Language Inference corpus, a large-scale, crowd-sourced dataset for English natural language inference (NLI). NLI is a task in natural language processing where a system analyzes a pair of sentences (a premise and a hypothesis) to determine their relationship. The relationship can be one of three types: entailment, contradiction, or neutral. The MNLI dataset provides sentence pairs annotated with these categories and is used for benchmarking and training NLI models, including transformer-based models like BERT and RoBERTa.

2.6 ALBERT

ALBERT (A Lite BERT) is a pre-trained language model that enhances and optimizes BERT. It reduces the number of parameters and shares them to enable more efficient training and inference. ALBERT achieves parameter efficiency by sharing layer parameters and introduces the SOP task as a secondary goal, enhancing sentence-level relations and reasoning. These improvements make ALBERT more efficient in training and inference, allowing it to achieve better results with the same computational resources compared to BERT and RoBerta(Lan et al., 2019).

3 Experiments and results

3.1 Data preprocessing

In this paper, we preprocess the dataset using the tokenizer() function to tokenize the sentences. The tokenized sentences are in the form below with [CLS] and [SEP]:

```
[CLS] + sentA + [SEP] + sentB
```

We define the class ComVEDataset to store the encodings and labels in the dataset. After preprocessing, the various models used in this paper are fine-tuned for sequence classification.

For multiple choice, we also tokenize the original dataset and collate the encodings and labels(Zhao et al., 2020). The sentences are in the format:

```
[CLS] + Q + [SEP] + sentA + [SEP]
[CLS] + Q + [SEP] + sentB + [SEP]
```

The models are then fine-tuned for the multiple choice task.

3.2 Hyperparameters

Through experiments with others, our own observations, and consideration of the device, the following model hyperparameters were finally determined.

Model type	Batch size	Lr	Epoch	Optimizer
BERT	16	1.5e-5	4	Adam
RoBERTa	24	1e-5	8	AdamW
ALBERT(MC)	8	1e-5	4	AdamW
ALBERT(SC)	16	1e-5	4	AdamW

Table 2: Some fixed hyperparameters for each model class

3.3 BERT

In the sequence classification and multiple choice approaches using the BERT-base-uncased model, we adjust the parameters and obtain the best results as shown in the table2.

3.4 RoBERTa

RoBERTa-Large has more hidden layers and approximately double the parameters of RoBERTa-Base. With better training results, we choose RoBERTa-Large for our purposes. When utilizing the RoBERTa model for sequence classification and multiple choice tasks, we adjust the parameters and consider previous research. We conduct tests with and without an additional prompt question for the multiple choice task.

The results of the tests without a prompt question are shown in Table 4. We also conducted tests with a constructed prompt question and achieved the best results, which are presented in Table 4. Additionally, we utilized the RoBERTa+MNLI model for training and testing, and the results are also displayed in Table 4.

P1 and P2 represent two different prompt questions, with reference to previous research(Zhao et al., 2020), the specific input we used in multiple choice was as follows:

Type	Input
Orig.	[CLS] Si [SEP]
Multiple choice P1	[CLS] If the following statement is in common sense? [SEP] Si [SEP]
Multiple choice P2	[CLS] If Si is in common sense? [SEP]

Table 3: The specific input we used in multiple choice

3.5 ALBERT

In the completion of sequence classification and multiple choice using the ALBert-xxlarge-v2 model, we adjust the parameters and obtain the best results as shown in the table2.

It can be observed that the ALBert model achieves the highest accuracy rates for both approaches.

	MC		SC	
Epoch	**Train acc.**	**Valid acc.**	**Train acc.**	**Valid acc.**
1	0.925	0.958	0.925	0.958
2	0.985	0.949	0.985	0.949
3	0.996	0.948	0.996	0.948
4	0.995	0.945	0.995	0.945

Table 4: The training and validation accuracy for ALBert MC and SC

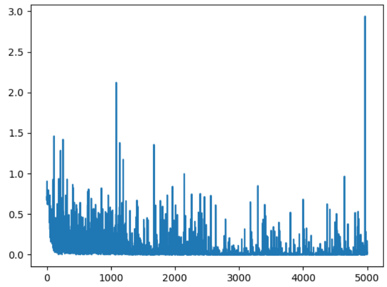

Figure 2: Loss curve of multiple choices on ALBERT

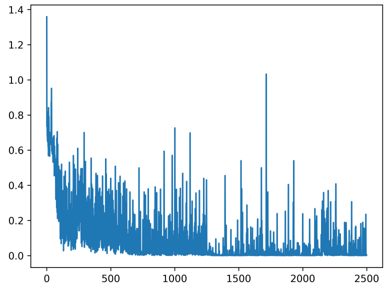

Figure 3: Loss curve of sequence classification on ALBERT

It can be seen that the fine-tuning convergence speed of the ALBERT model on this problem is quite fast, and its accuracy on the verification set will remain stable after about 1 epoch. Compared to the previous model, this model is more efficient for fine-tuning. However, due to the large number of model parameters, the training speed is not fast.

3.6 Results

The model with the highest accuracy in the test set is ALBert(MC), due to graphics memory limitations, we can only set the batch size to 8, we believe that if the batch size is 16, it will further improve the model performance.

Model	**Test acc.**	**Train acc.**	**Valid acc.**
BERT (SC)	84.6%	99.7%	85.7%
BERT (MC)	87.5%	99.7%	88.8%
RoBERTa (SC)	91.9%	99.8%	94.0%
RoBERTa +Orig.	94.7%	100%	95.2%
RoBERTa +P1	95.3%	99.6%	95.6%
RoBERTa +P2	93.0%	98.9%	93.5%
RoBERTa +MNLI	95.0%	98.2%	95.6%
ALBERT (SC)	94.3%	97.8%	95.8%
ALBERT (MC)	95.8%	92.5%	95.8%

Table 5: The best results within each model

In this table, "(SC)" and "(MC)" represent the use of sequence classification and multiple choice approaches, respectively. The "+Orig.", "+P1", "+P2" denote different prompt questions used in the multiple choice task. "+MNLI" indicates the usage of MNLI.

4 Discussion

In our results, the highest correct rates were obtained using the Albert model. Additionally, the accuracy of multiple choice questions was consistently higher than that of sequence classification. This disparity may be attributed to the differing approaches in information processing and prediction-making. Sequence classification relies on the content of the input sequence to predict outcomes, necessitating a model with strong understanding and reasoning abilities. Conversely, multiple choice questions enable a direct comparison of alternatives, allowing for the selection of the most likely option. Furthermore, the inclusion of contextual information within the multiple choice questions assists the model in making predictions. Our dataset directly provides two options, facilitating a straightforward comparison and prediction through the multiple choice format. Consequently, multiple choice outperforms sequence classification in this particular task.

Besides, in the multiple choice downstream task, we found that using question "If the following statement is in common sense ?" and "If the following statement is against common sense ?" has similar performance. We think that the performance on multiple-choice questions heavily relies on the patterns and information present in the training data. Some language models may learn statistical patterns from the data, including common phrasings or recurring patterns in questions and answers. If the opposite question shares similar linguistic patterns with the original question, the model might generate responses based on those patterns rather than genuine comprehension of the question's content.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. 2021. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919.

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. *arXiv preprint arXiv:1909.11942*.

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

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

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

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. *arXiv preprint arXiv:1906.00363*.

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. *arXiv preprint arXiv:2007.00924*.

Qian Zhao, Siyu Tao, Jie Zhou, Linlin Wang, Xin Lin, and Liang He. 2020. Ecnu-sensemaker at semeval-2020 task 4: Leveraging heterogeneous knowledge resources for commonsense validation and explanation. *arXiv preprint arXiv:2007.14200*.