Lijunyi at SemEval-2020 Task 4: An ALBERT Model based Maximum Ensemble with Different Training Sizes and Depths for Commonsense Validation and Explanation

Junyi Li, Bin Wang, Haiyan Ding*
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
*Corresponding author, hyding@ynu.edu.cn

Abstract

This article describes the system submitted to SemEval 2020 Task 4: Commonsense Validation and Explanation. We only participated in the subtask A, which is mainly to distinguish whether the sentence has meaning. To solve this task, we mainly used ALBERT model-based maximum ensemble with different training sizes and depths. To prove the validity of the model to the task, we also used some other neural network models for comparison. Our model achieved the accuracy score of 0.938(ranked 10/41) in subtask A.

1 Introduction

After thousands of years of accumulation, human civilization has undergone tremendous changes. People can accumulate a lot of experience every day in study, work and life. These large amounts of experience have become our common knowledge through summarization and verification. In our daily life, common sense can often tell us some other people's practical experience, so that we can avoid the repetition of some errors. After scientific verification, a lot of wrong common sense is slowly being corrected. However, it takes a lot of human resources to manually correct normal knowledge. At the same time, in recent years, due to the development of natural language processing and neural networks, common sense revision has entered the era of mechanization.

Semeval 2020 task 4 (Wang et al., 2020) (Wang et al., 2019) is designed for common sense verification and interpretation. This task is to directly test whether the system can distinguish meaningful natural language statements from unreasonable natural language statements. In this way, we can save a lot of time to distinguish between common sense and extraordinary knowledge. Subtask A is to choose from two natural language statements with similar wording, one of which is meaningful and the other is meaningless. This subtask is mainly to directly distinguish whether the discourse has common sense. Subtask B is to find out the key reason why this sentence violates common sense from the three options that can be selected. Subtask C is the reason why generating statements violates common sense. For this task, we mainly participated in subtask A. We want to be able to effectively distinguish meaningful sentences from meaningless sentences through subtask A.

In this task, we only participate in subtask A: Identify whether the language has meaning. For this task, we use a combination of research and deep learning methods to deal with related tasks. In the task, we mainly use the lightweight ALBERT model based on the Transformer mechanism. According to the progress of the latest related research, the ALBERT model has become our preferred model. Because it has fewer parameters and better performance than the bert model. After data processing, we input the generated word vector into the trained model. Besides, we also used pre-trained models of different training sizes and depths for comparative experiments to optimize the prediction probability of the results. Meanwhile, we also adopted the method of maximum voting ensemble to optimize the performance of our model. To prove the excellent performance of our method, we use some other neural networks for comparative experiments. In this task, our method is an effective way to get good performance.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 556–561
Barcelona, Spain (Online), December 12, 2020.

The rest of our paper is structured as follows. Section 2 introduces related work. Model and data preparation are described in Section 3. Experiments and evaluation are described in Section 4. The conclusions are drawn in Section 5.

2 Related Work

In natural language processing, the problem of common sense verification has always been an important one. The common sense verification problem has a large data set. Such as, Event2Mind (Rashkin et al., 2018) is a crowdsourced corpus of 25,000 event phrases covering a diverse range of everyday events and situations. Situations with Adversarial Generations (SWAG) (Zellers et al., 2018) is a dataset consisting of 113k multiple choice questions about a rich spectrum of grounded situations. The winograd schema challenge (Trinh and Le, 2018) is a dataset for common sense reasoning. It employs winograd Schema questions that require the resolution of anaphora: the system must identify the antecedent of an ambiguous pronoun in a statement. Reading Comprehension with Commonsense Reasoning Dataset (ReCoRD) (Zhang et al., 2018) is a large-scale reading comprehension dataset which requires commonsense reasoning.

For these data sets, there are some neural networks that can get good performance. For example, convolutional neural network (Kim, 2014), bidirectional recurrent neural network (Wang et al., 2018), BERT model based on transform mechanism (Devlin et al., 2018), Multi-Task Deep Neural Networks (Liu et al., 2019), XLNet (Yang et al., 2019), etc.

Based on the above research, we have a clear direction to deal with this task.

3 Model and Data Preparation

Our neural network model is shown in Figure 1. In this task, we mainly use a lightweight ALBERT model based on the Transformer mechanism.

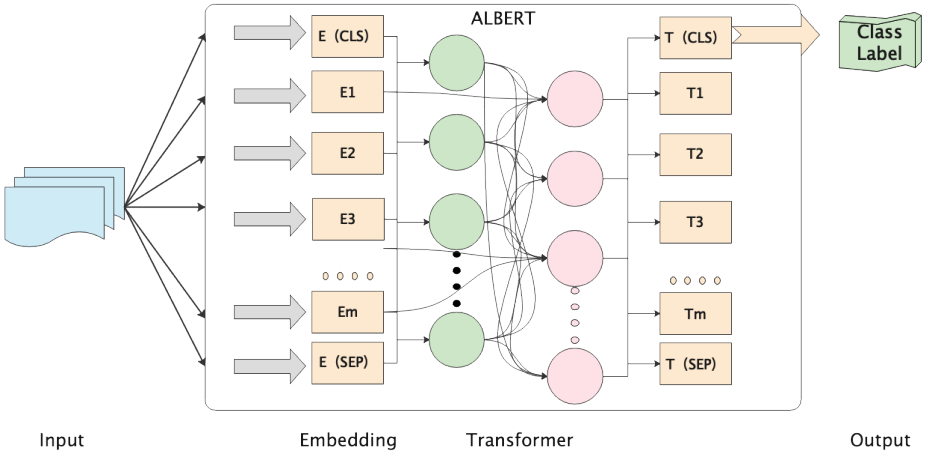

Figure 1: The architecture of the ALBERT model in our task

where the $E[CLS]$ and $E[SEP]$ are added at the beginning and end of each instance, respectively, which can separate different sentences.

3.1 Data Preparation

In this task, we only participated in subtask A. The data set of subtask A is composed of two files, one of which is called $data$ and the other is called $answers$. The $data$ file mainly includes two selected sentence pairs and the id of the sentence pairs. The $answers$ file includes the sequence number of the sentence that does not conform to common sense and the id of the sentence pair. In order to facilitate our data preprocessing, we merged the two files. This way we can include data and answers in one file. At the same time, we can display the data more intuitively. The format of our merger is shown in Figure 2.

Id	Sent0	Sent1	Label
0	He poured orange juice on his cereal.	He poured milk on his cereal.	0
1	He drinks apple.	He drinks milk.	0
...	...	...	...
...	...	...	...

Figure 2: The format of the dataset after merged in subtask A of task 4

The organizers provided training, dev and test sets, containing 10000, 997 and 1000 data respectively. Cleaning the text before further processing helps to generate better functionality and semantics. We perform the following preprocessing steps.

- Tokens are converted to lower case.
- We know that some repeated symbols have no meaning. As a result, repeated periods, question marks and exclamation marks are replaced with a single instance with the special mark ”repeat” added.
- All contractions were changed to complete parts. This helps the machine understand the meaning of words (for example:”there’re” changed to ”there” and ”are”.
- Generally, words have different forms according to the change of context. However, different forms of words will cause ambiguity in pre-training and affect the effect of pre-training. Lexicalization, through WordNetLemmatizer to restore language vocabulary to the general form (can express complete semantics).

3.2 ALBERT

The ALBERT model (Lan et al., 2019) is a lightweight improvement of the BERT model. The embedding layer of the ALBERT model and the bert model is similar. The ALBERT model performs 3 embeddings: the word embedding; the position embedding and segment embedding. The word embedding encodes the information of each word, and the position embedding is similar to the word, mapping a position into a low-dimensional dense vector. Segment embedding indicates whether the currently encoded word belongs to the same sentence. However, the embedding layer of the ALBERT model is much smaller than the embedding layer parameter of the BERT model. There are many redundant parameters that have not been entered. These parameters usually have no effect. The embedding layer of the ALBERT model is shown in Figure 3.

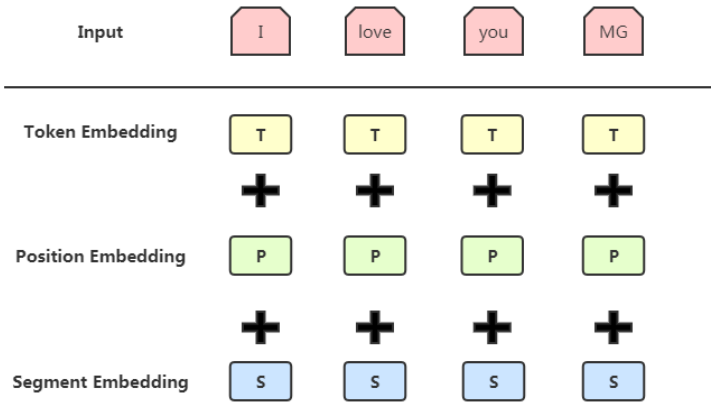

Figure 3: The detail of embedding layer in ALBERT model

3.3 Max Ensemble

We know that common ensemble methods are based on voting ensemble. In this paper, we first fine-tune the ALBERT model with different random seeds. For each input, we will output the best predictions and probabilities made by fine-tuned ALBERT, and summarize the prediction probabilities of each model. The output of the ensemble model is the prediction with the highest probability. We call this integration method voting ensemble, and the ALBERT model after voting ensemble (Xu et al., 2020) has been significantly improved. The formula for the ensemble ALBERT model we used is shown below

$$ALBERT_{vote}(x;s) = Max(\sum_{n=1}^{s} ALBERT(x_s)) \tag{1}$$

where $ALBERT_{vote}(x;s)$ is the max ensemble ALBERT model and $ALBERT(x_s)$ represents a fine-tuning of the ALBERT model.

In this task, our voting ensemble is chosen when outputting the maximum performance. Our voting ensemble is shown in Figure 4, where x represents our input and y represents the model ensemble output.

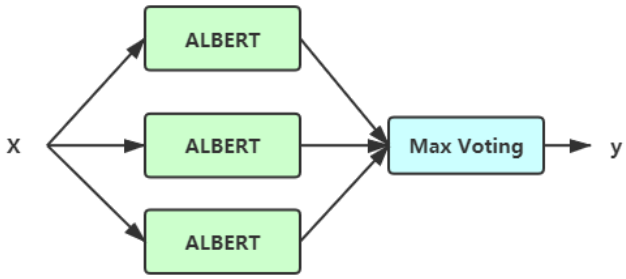

Figure 4: The architecture of max ensemble in our task

3.4 Attention-LSTM and Text-CNN

TextCNN is a model that uses multiple convolutional neural networks to output in tandem (Kim, 2014). In the model, the convolution window of each convolutional neural network is different in size. The convolution results obtained by convolution windows of different sizes are combined and output.

In the attention-based LSTM model, all sentences and labels are converted to word vectors by the word embedding layer. These word vectors will be fed to the LSTM layer. Subsequently, the word vector is represented as a hidden vector. Next, the attention(Vaswani et al., 2017) mechanism assigns weights to each hidden vector. Note that the mechanism produces attention weight vectors and weighted hidden representations. Note that the weight vector is mainly obtained by calculating the similarity. An attention weight vector is generated by calculating a sentence vector matrix and a label vector matrix.

For related work, we chose Attention-LSTM and Text-CNN, which can verify the effectiveness of the ALBERT model for the task.

4 Experiments and Evaluation

In this task, we obtained good performance using the ALBERT model. Meanwhile, we found that ALBERT models with different pre-training sizes and pre-training depths will get different results. We use dev data set to verify the performance of the model. The standard of our judgment is accuracy, and this standard is the judgment standard used for our task. In our task, the results of our comparison are shown in Table 1. Where, H is the number of hidden layers. L is the number of layers of the model and E is the number of layers of the model embedding.

By comparing the performance of the ALBERT model with different pre-training depths, we can get the following conclusions. In this task, we found that increasing the number of hidden layers by a certain number can improve the performance of the ALBERT model. At the same time, in the task, based on the

Model	status	L	H	E	Accuracy
ALBERT	base	12	768	128	0.83
	large	24	1024	128	0.86
	xlarge	24	2048	128	0.88
	xxlarge	**12**	**4096**	**128**	**0.92**

Table 1: The parameter configuration of the ALBERT model on dev data sets

dev data set, we see that the model achieves the best results in the $xxlarge$ state. Therefore, we will use the pre-trained model in the $xxlarge$ state to make predictions.

In this task, we use the ALBERT model to pre-train the task. For the ALBERT model, the main hyper-parameters we pay attention to are the training step size, batch size and learning rate. At the same time, according to relevant work research, we also use some other neural network models for experiments. For other models, the main parameters we pay attention to are batch size learning rate and epoch. The parameters of all models are shown in Table 2.

Models	train step	learning rate	batch size	epoch
BERT	20000	4e-6	32	30
Attention-LSTM	None	3e-4	32	20
Text-CNN	None	2e-4	64	30
ALBERT	20935	5e-6	32	None

Table 2: Details of the hyper-parameters.

This sub-task A is to evaluate the classification system by calculating the accuracy. In order to prove the effectiveness of our model to improve performance. We also used other neural network models for comparative experiments. We hope to verify that our model is more effective for our task through comparative experiments. The results of our comparative experiments are shown in Table 3.

Model	Accuracy
Attention-LSTM(avg)	0.74
Text-CNN(avg)	0.78
BERT(Base)	0.82
BERT(Large)	0.85
ALBERT(Max ensemble)	**0.938**

Table 3: Results of our comparative experiments on the test data set.

where avg represents the average result of the model after 5 experiments. Through 5 experimental trainings, accidental errors can be eliminated and $BERT(Large)$ and $BERT(Base)$ mean that the BERT model is large-scale and base-scale.

Through the table, we can clearly see that ALBERT models based maximum ensemble with different training sizes and depths has good performance in this task.

5 Conclusion

In this task, we mainly used ALBERT model-based maximum ensemble method with different training sizes and training depths for experiments. At the same time, we also used some other neural networks for comparative experiments, to prove that our model can obtain excellent performance. The results show that our model can achieve the best performance in this task.

In the future, we will continue to adjust the model, improve the hardware configuration of the computer, collect more external data, and conduct more experiments to obtain better results.

Acknowledgements

This work was supported by the Natural Science Foundations of China under Grant 61463050, the Science Foundation of Yunnan Education Department under Grant 2020Y0011.

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*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

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*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv:1904.09482*.

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

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual lstm model for sentiment intensity prediction. *Neurocomputing*, 322:93–101.

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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy, July. Association for Computational Linguistics.

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 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.