YNU-oxz at SemEval-2020 Task 4: Commonsense Validation using BERT with Bidirectional GRU

Xiaozhi Ou, Hongling Li *

School of Information Science and Engineering

Yunnan University, Yunnan, P.R. China

Abstract

This paper describes the system and results of our team participated in SemEval-2020 Task4: Commonsense Validation and Explanation (ComVE), which aim to distinguish meaningful natural language statements from unreasonable natural language statements. This task contains three subtasks: Subtask A–Validation, Subtask B–Explanation (Multi-Choice), and Subtask C–Explanation (Generation). In these three subtasks, we only participated in Subtask A, which aims to distinguish whether a given two natural language statements with similar wording are meaningful. To solve this problem, we proposed a method using a combination of BERT with the Bidirectional Gated Recurrent Unit (Bi-GRU). Our model achieved an accuracy of 0.836 in Subtask A (ranked 27/45).

1 Introduction

In recent years, the introduction of commonsense into natural language understanding systems has received more and more research attention. Commonsense Validation and Explanation describes whether the natural language statement is consistent with the facts and explains the reasons for its against common sense. Specifically, the purpose of the commonsense verification is to determine that the natural language statement is reasonable and meaningful. This task involves commonsense, understanding, and reasoning, so it is very important for a system to be able to accurately distinguish between meaningful natural language statements and unreasonable natural language statements. In SemEval-2020 Task 4: Verification and Interpretation of Common Sense (Wang et al., 2020). The organizer has designed three subtasks. Subtask A selects from two natural language statements with similar grammatical structures (one of which is meaningful and one of which is not). It requires the model to determine which of the two statements violates commonsense; Subtask B is to find the key reason why a given statement violates commonsense from three options, this task is a multiple-choice; Subtask C requires the system to generate the reason why this statement violates commonsense. Among these three subtasks, Subtask A and Subtask B are evaluated by accuracy, and Subtask C is evaluated by BLEU. For Subtask A, we experimented with different neural network-based models, such as Bidirectional Long Short-Term Memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997), Bidirectional Gated Recurrent Unit (Bi-GRU) (Bahdanau et al., 2014), and bidirectional encoder representation from Transformers (BERT) (Devlin et al., 2018). We found that BERT performs best on the validation set. Therefore, we chose to use BERT to combine different models to optimize the system.

The rest of the paper is structured as follows. Section 2 introduce the related work of ComVE; section 3 introduce the data description and methods; in section 4, we analyze the experimental results; in section 5, we summarize and look forward to future work.

*Corresponding author: honglingli66@126.com

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 626–632
Barcelona, Spain (Online), December 12, 2020.

2 Related Work

Commonsense verification aims to require models to determine whether natural language statements are meaningful. One of the key issues in how to build an effective classifier is to find suitable features. Through the introduction of word embedding technology, better performance can be achieved. A common way of a neural network is to add more layers in order to learn high-level features, and the stacked residual LSTM model can solve the problem of gradient disappearance and deviation between layers (Wang et al., 2018a). Commonsense reasoning is a challenging task for modern machine learning methods (Zhong et al., 2018; Talmor et al., 2019). Many commonsense representations and reasoning processes have explored and developed large knowledge bases of commonsense. The JHU Ordinal Common-sense Inference (JOCI) (Zhang et al., 2017) aims to mark the plausibility of human response in certain circumstances from 5 (very likely) to 1 (impossible). The story cloze test (also known as ROC story) compares the true ending of the story to the incredible false ending (Mostafazadeh et al., 2016). Some researchers also investigate ways to use common sense to answer questions. CommonsenseQA (CQA) (Talmor et al., 2019) requires crowd workers to create questions from ConceptNet (Speer et al., 2017). Talmor et al. (2019) show that using Google search to extract context from the first 100 result snippets selected for each question and answer does not greatly improve. CQA is even trained using the most advanced reading comprehension model BiDAF ++ (Seo et al., 2017), which also adds a self-attention layer and Elmo representation (Peters et al., 2018). (Camburu et al., 2018) generates interpretation and prediction of natural language inference problems.

A survey of commonsense data sets and more common natural language data sets show that there are already some data that focus on non-linguistic world knowledge verification (Wang et al., 2018c) or limited attributes or actions of world knowledge (Forbes and Choi, 2017). The Winograd Schema Challenge (WSC) (hector j levesque et al., 2012; Morgenstern and Ortiz, 2015) is a widely used and important task that requires more commonsense knowledge. In recent years, some large-scale knowledge resources of commonsense reasoning have also been released, which may be helpful for the development of commonsense reasoning tasks. Common sense Question Answering (CQA) is a multiple-choice question and answer datasets proposed for the development of natural language processing (NLP) models with common sense reasoning capabilities (Talmor et al., 2019). Sap et al. (2019) proposes a huge knowledge map of commonsense reasoning, which has nine if-then relationship variables, including causes, effects, etc. Techniques based on Language Modeling, such as GPT and BERT models, can achieve human-level performance on these datasets (Radford et al., 2018; Devlin et al., 2019).

3 Methodology and Data

3.1 Data description

In this task, we first make a simple statistics on the dataset, as shown in Table 1.

Task	Training set	Validation set	Test set
Subtask A	10000	997	1000
Subtask B	10000	997	1000
Subtask C	1074	997	1000

Table 1: Data statistics table

id	sent0	sent1
1	He drinks apple.	He drinks milk.
4	A niece is a person.	A giraffe is a person.
10	Eggs eat kis on Easter.	Kids find eggs on Easter.

Table 2: Some instance of Subtask A in the training set

As can be seen from Table 1, for the training set, there are 10,000 against commonsense sentences and 10,000 correct sentences in Subtask A. Table 2 shows some instances in the dataset of Subtask A, we can see that each row in the dataset contains 3 fields: id, sent0, sent1, which are the id of the instance, and two input sentences.

3.2 Methodology

In this task, we have experimented with some methods. In this section, we introduce our methods in detail. We report the performance of these methods on the validation set. The experimental results show that BERT shows the power of the pre-trained model, and on the basis of which fine-tuning can improve the effect, so we try to combine other methods with the BERT model. Finally, the experimental results prove that the combination of BERT with Bi-GRU is the best effect on the validation set.

Bi-LSTM We use Bidirectional LSTM as a baseline model to compare results. Bi-LSTM (Hochreiter and Schmidhuber, 1997) is the abbreviation of Bidirectional Long Short-Term Memory, which is a combination of forward LSTM and backward LSTM. The LSTM model can better capture long-distance dependencies, but it can't encode the information from the back to the front. Bi-LSTM can better capture bidirectional semantic dependencies.

Bi-GRU Chung et al. (2014) proposed an LSTM variant called gate recursive unit (GRU), GRU is to combine the forget gate and input gate in LSTM into update gate. It makes GRU simpler and more efficient than traditional LSTM models (Wang et al., 2018b). We use the structure of a bidirectional GRU to encode vectorized text to establish this contextual connection. Bi-GRU is a neural network model consisting of unidirectional GRU with opposite directions and whose output is jointly determined by the states of these two GRUs. The input of the forward GRU is the forward sequence of the input of the previous layer, and the input of the backward GRU is the reverse sequence of the input of the previous layer. At each moment, the input provided two GRUs with directions opposite at the same time, and the output is decided by the two unidirectional GRUs jointly. The simple structure of Bi-GRU is shown in Figure 1. The current hidden layer state of Bi-GRU is jointly determined by three parts: current input x_t, output $\overrightarrow{h_{t-1}}$ of forward hidden layer state and output $\overleftarrow{h_{t-1}}$ of backward hidden layer state at *t-1* moment:

$$\overrightarrow{h_t} = GRU(x_t, \overrightarrow{h_{t-1}}) \tag{1}$$

$$\overleftarrow{h_t} = GRU(x_t, \overleftarrow{h_{t-1}}) \tag{2}$$

$$h_t = [\overrightarrow{h_t} : \overleftarrow{h_t}] \tag{3}$$

where the GRU() function represents a non-linear transformation of the input word vector, and encodes the word vector into the corresponding GRU hidden layer state. $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ respectively represent the forward hidden state and the backward hidden state corresponding to the bidirectional GRU at *t* moment; h_t express the vector that contact $\overrightarrow{h_t}$ with $\overleftarrow{h_t}$.

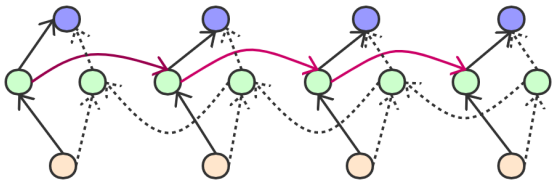

Figure 1: Bidirectional GRU (Bi-GRU) structural model diagram

BERT The full name of BERT is Bidirectional Encoder Representation from Transformers (Devlin et al., 2018). Its main model structure is composed of the encoder stack of Transformer. The main innovations of the model are in the pre-train method, that is, Masked LM and Next Sentence Prediction

are used to capture the representation of words and sentences respectively. Since the dataset in SemEval-2020 Task 4 is small, we input the dataset into a pre-trained BERT model. The overall architecture diagram of the model is shown in Figure 2. The fine-tuning BERT of text pair classification in Figure 2 is different from the single text classification in terms of input representation. The BERT input sequence explicitly indicates the input of a text pair, where the special classification mark "$\langle CLS \rangle$" is used for sequence classification, and the special classification mark "$\langle SEP \rangle$" marks the end of a single text or separates a pair of text. As shown in Figure 2, in the text classification application, the BERT of the special classification mark "$\langle CLS \rangle$" denotes to encode the information of the entire input text sequence, and as the representation of the input text, it will be input into a small Multi-Layer Perceptron(MLP) composed of fully connected (Dense) layers to output the distribution of all discrete label values. Finally, a softmax layer is used for the final output. However, in order to obtain more abundant feature vectors, we join up the Bi-GRU layer after BERT.

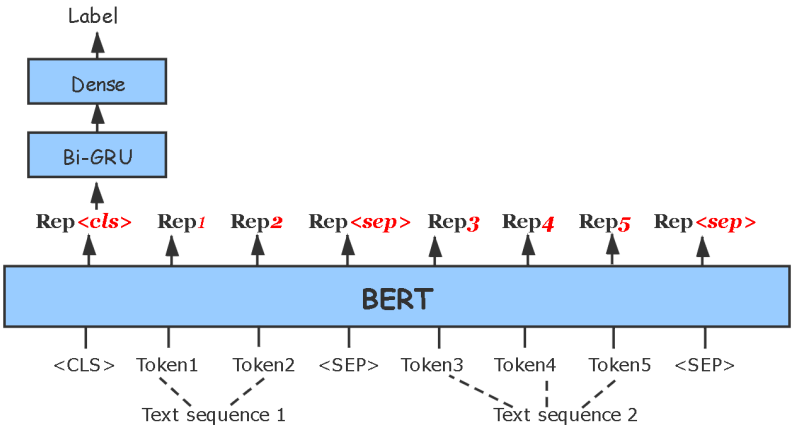

Figure 2: The overall architecture diagram of the model

4 Experiment results

4.1 Evaluation

In this task, Subtask A and Subtask B are evaluated by accuracy, while Subtask C is evaluated by BLEU. For Subtask A in which we are participant, we evaluate it by accuracy. We introduce a classic confusion matrix to help us understand accuracy, as shown in Figure 3. Accuracy considers all samples, which

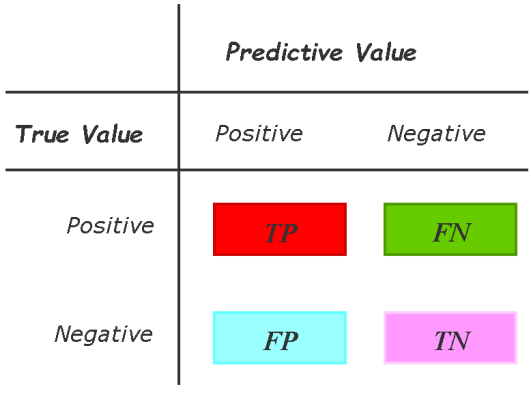

Figure 3: Confusion matrix

refers to the ratio of the number of correctly predicted samples to the total number of predicted samples. The accuracy rate reflects the correct predictions made by the model. The higher the accuracy, the better the classifier. The calculation formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

4.2 Experiment setting

In our experiment, we used BERT-base-uncased as our pre-trained models, and the batch size is set to 8, the epoch is set to 3, and the learning rate is set to *e-5*. For the Bi-GRU layer, we set the hidden unit to 512, added Dropout layers and BatchNormalization layers, and the rate of dropout layer is 0.35. The activation function of the final output layer is sigmoid, which is used for binary classification. The loss function of this model is binary cross-entropy, and the optimizer is adam. In our model, the input form of our model is $\langle CLS \rangle$ + sentence 1+ $\langle SEP \rangle$ + sentence 2+ $\langle SEP \rangle$, where $\langle CLS \rangle$ represents the special token of the classification task, its output is the pooler output of the model, and $\langle SEP \rangle$ represents the separator, sentence 1 and sentence 2 are the input text of the model. The experimental results show that the model converges fast after 3rd epoch and the loss on the validation set is small.

4.3 Result

In this section, we report the results on the validation set and the official test data set. In Table 3, we list the performance of the three basic models participating in the selection of Subtask A on the left, in which BERT achieved the best performers, and the accuracy reached 88.7%. In order to obtain richer feature vectors, we consider joining up RNN structure. we list the performance of the two combined models on the right, in which BERT+Bi-GRU achieved the best performers, and the accuracy reached 92.3%. It can be seen from Table 3 that the performance of the combined model is significantly higher than the basic model. Based on the verified results, we chose the combination of BERT with Bi-GRU as the final submitted model. In Table 4 shows the top three in the official ranking and the result we finally submitted.

Basic model	**Acc(Validation set)**	**Combination model**	**Acc(Validation set)**
Bi-LSTM	82.5	BERT+Bi-LSTM	90.8
Bi-GRU	84.3	BERT+Bi-GRU	92.3
BERT	88.7		

Table 3: Model results on validation set

Team Name	**Subtask A Accuracy**
hit_itnlp	97.0(1)
ECNU_ICA	96.7(2)
iie-nlp-NUT	96.4(3)
YNU_OXZ	83.6(27)

Table 4: Official leaderboard results

5 Conclusion

This year we participated in the sub-task A for Commonsense Validation and Explanation (ComVE), which is to verify which of the two sentences is against commonsense. This paper proposes a model combining BERT with Bi-GRU. Using the BERT model to extract more suitable features, and a combination model with a layer of Bi-GRU behind BERT is used for classification. Although the overall performance of our model is not ideal, the preliminary results show what we should do next. In future research, we will use multiple methods to improve our model and will consider introducing k-fold cross-validation to integrate the results to improve model performance.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. abs/1409.0473.

Oana-Maria Camburu, Tim Rocktschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli - natural language inference with natural language explanations. *Annual Conference on Neural Information Processing Systems*, abs/1812.01193:9560–9572.

Junyoung Chung, aglar Glehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. abs/1412.3555.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

Maxwell Forbes and Yejin Choi. 2017. Verb physics: Relative physical knowledge of actions and objects. In *Annual Meeting of the Association for Computational Linguistics*, volume abs/1706.03799, pages 266–276.

hector j levesque, ernest davis, and leora morgenstern. 2012. The winograd schema challenge.

S Hochreiter and J Schmidhuber. 1997. Long short-term memory. 9:1735–1780.

Leora Morgenstern and Charles Ortiz. 2015. The winograd schema challenge: Evaluating progress in commonsense reasoning. *AAAI Conference on Artificial Intelligence*, pages 4024–4026.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. pages 839–849.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, volume abs/1802.05365.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. abs/1611.01603.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, volume abs/1612.03975.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. pages 4149–4158.

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

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

Su Wang, Greg Durrett, and Katrin Erk. 2018c. Modeling semantic plausibility by injecting world knowledge. In *Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, volume abs/1804.00619.

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.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. abs/1611.00601.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2018. Improving question answering by commonsense-based pre-training. abs/1809.03568.