TCU at SemEval-2022 Task 8: A Stacking Ensemble Transformer Model for Multilingual News Article Similarity

Xiang Luo[1] ,Yanqing Niu[1] and Boer Zhu[2]
[1]School of Mathematics and Statistics, South Central University for Nationalities, Wuhan, China
[2]School of Computer Science, South Central University for Nationalities, Wuhan, China
Contact: niuyanqing@mail.scuec.edu.cn

Abstract

Previous studies focus on measuring the degree of similarity of texts by using traditional machine learning methods, such as Support Vector Regression (SVR). Based on Transformers, this paper describes our contribution to SemEval-2022 Task 8 Multilingual News Article Similarity. The similarity of multilingual news articles requires a regression prediction on the similarity of multilingual articles, rather than a classification for judging text similarity. This paper mainly describes the architecture of the model and how to adjust the parameters in the experiment and strengthen the generalization ability. In this paper, we implement and construct different models through transformer-based models. We applied different transformer-based models, as well as ensemble them together by using ensemble learning. To avoid the overfit, we focus on the adjustment of parameters and the increase of generalization ability in our experiments. In the last submitted contest, we achieve a score of 0.715 and rank the 21st place.

1 Introduction

Providing computer the ability to understand the abstract meaning of real world is a fundamental tasks. Given a pair of news articles, this task seek to evaluate the semantic similarity between them, which focuses on the real world-happenings covered in the news articles. It's a regression problem for measuring similarity of multilingual texts.

Previous studies measured the similarity between texts by using traditional machine learning methods, such as using Support Vector Regression (SVR) (Šarić et al., 2012). Recently many deep learning methods came out, such as pre-trained model. It had attracted the interest of researchers and had shown good result. For example, Exploring Bidirectional Encoder Representations from Transformers (BERT), XLNet and Robustly optimized BERT approach (RoBERTa) and finally got a good ranking (Yang et al., 2020). And a new hybridized approach using Weighted Fine-Tuned BERT Feature extraction with Siamese Bi-LSTM model has been implemented. It is employed for determining question pair sets using Semantic-text-similarity from Quora dataset (Viji and Revathy, 2022). These novel deep learning methods have performed well.

In this study, we explored some transformer-based models. We had employed BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), A Lite BERT for Self-supervised Learning of Language Representations (ALBERT) (Lan et al., 2020), DistilBERT (Sanh et al., 2020; Gou et al., 2021), besides this, we used them as base models, merged them together with ensemble learning, and the prediction result is used as a new training set, and then SVR is used as the meta model. The training set generated by the base model is put into the meta-model. Afterwards, the final result is predicted by the meta-model. The advantage of the pre-trained model is that the upstream corpus has already trained the parameters of the model well. We only need to fine-tune it, and we don't need a huge training set for training (Kong et al., 2022).

The rest of this paper is organized as follows. Section 2 describes all the models which are used for measuring similarity between sentence pairs. Experimental results are summarized in Section 3. Conclusion is drawn in Section 4.

2 Model Description

This section will describe what models we have used, and how they organized. Because of the attention mechanism, pre-trained model had been made a huge success in Nature Language Processing (NLP). We use transformer-based models such as BERT, RoBERTa, ALBERT, DistilBERT to produce hidden representations. Then, a stacking ensemble strategy was used to ensemble the results. The details are presented as follow.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1202 - 1207
July 14-15, 2022 ©2022 Association for Computational Linguistics

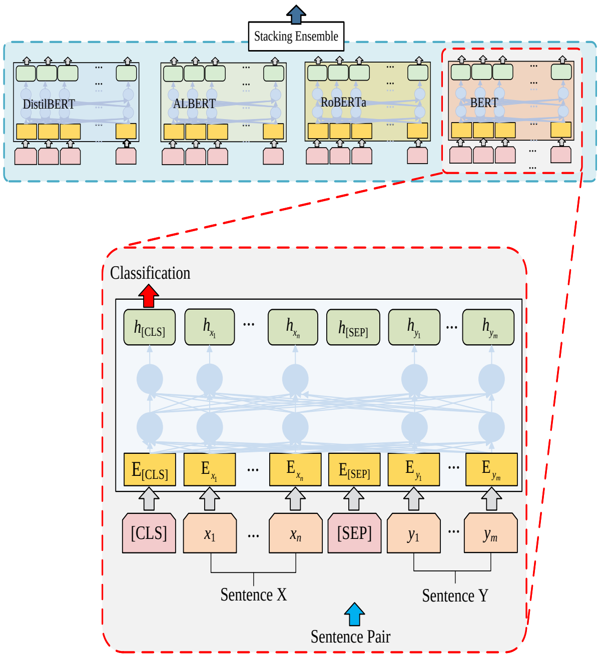

Figure 1: The overall architecture of the proposed method.

2.1 Base Models

Based on the self-attention mechanism, transformer-based model become a popular method in NLP. The models, such as BERT, ALBERT, RoBERTa, DistilBERT are variants on the improvement of the transformer architecture (Huang et al., 2021).

For four Transformer encoders, we applied a similar architecture as sentence pair classification to learn representation for the final regression. Given two sentences and $\mathbf{X} = [x_1, x_2, ..., x_n]$ and $\mathbf{Y} = [y_1, y_2, ..., y_m]$. The model used WordPiece tokenizer to obtain subwords sequences. Two special tokens, i.e., [CLS] and [SEP], were added to the beginning of the whole sequence and between two sentences.The model architecture we use is shown in Fig. 1 The details of each Transformer encoder are presented as follows.

BERT. BERT stands for Bidirectional Encoder Representations from Transformers. BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. The checkpoint we use is “bert-base-multilingual-cased”, which uses 12-layer, 768-hidden, 12-heads, 110M parameters. Trained on cased text in the top 104 languages with the largest Wikipedias. The outputs tensor contains the batch_size, sequence_length, hidden_state and we use the first token to regress (Zhang et al., 2021). Moreover, BERT uses character-level BPE encoding.

RoBERTa. RoBERTa is a robustly optimized BERT pretraining approach, it’s an improved recipe for training BERT models, that can match or exceed the performance of all of the post-BERT methods. Our modifications are simple, they include: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. The checkpoint we use is “roberta-base”, which uses 12-layer, 768-hidden, 12-heads, 125M parameters. RoBERTa using the BERT-base architecture. So the output of RoBERTa is similar to BERT. The token of RoBERTa is called ‘sos’. After that, RoBERTa uses byte-level BPE encoding.

ALBERT. ALBERT is a lite BERT for self-supervised learning of language representations which lead to models that scale much better compared to the original BERT and it uses a self-supervised loss that focuses on modeling inter-sentence coherence, and show it consistently helps downstream tasks with multi-sentence inputs. The checkpoint we use is “albert-base-v2”, which uses 12 repeating layers, 128 embedding, 768-hidden,

MODEL	PEARSON	MSE
DistilBERT+ALBERT+BERT+RoBERTa	**0.880**	**0.298**
DistilBERT+BERT+ALBERT	0.879	0.301
DistilBERT+ALBERT+RoBERTa	0.815	0.450
DistilBERT+BERT+RoBERTa	0.879	0.301
ALBERT+RoBERTa	0.815	0.452
ALBERT+RoBERTa+BERT	0.880	0.298
BERT+ALBERT	0.877	0.306
BERT+RoBERTa	0.878	0.303
BERT+DistilBERT	0.879	0.301
ALBERT+DistilBERT	0.805	0.472
RoBERTa+DistilBERT	0.799	0.483
DistilBERT	0.749	0.587
BERT	0.874	0.310
ALBERT	0.792	0.504
RoBERTa	0.790	0.509

Table 1: The Pearson Score and MSE of Each Model in Test Data.

12-heads, 11M parameters. ALBERT base model with no dropout, additional training data and longer training. And ALBERT is also using the first position of token to regress which is similar to the token of BERT.

DistilBERT. DistilBERT is a distilled version of BERT,which pre-train a smaller general-purpose language representation model and can then be finetuned with good performances on a wide range of tasks like its larger counterparts. The checkpoint we use is "distilbert-base-cased", which use 6-layer, 768-hidden, 12-heads, 65M parameters. The DistilBERT model distilled from the BERT model bert-base-cased checkpoint. And the tokenization of DistilBERT is also similar to BERT.

2.2 Ensemble Learning

In ensemble learning, we train multiple models to solve the same problem and combine them to get better results. The most important assumption is that when weak models are combined correctly, we can get more accurate or robust models. We decide to use stacking as our ensemble learning model. Stacking usually considers heterogeneous weak learners and stacking learning to combine base models with meta-models.

We concatenate the output of the base regressor. Then we put the output into the meta-model which we use SVR as. After that, we use grid sweep to get the optimizer parameters, using SVR to output the prediction results.

We divide the base models into RoBERTa, BERT, DistilBERT and ALBERT, we first train each base model and save the best performing model, and then we combine them separately. We use SVR as our meta-model, take the output of the base model as the input of the meta-model, and then train the input data through the meta-model.The data we use is the test set divided from the training set, and the pearson score and MSE are used to judge the quality of the entire model. For different base models, we will adjust the parameters on the meta-model, so that each set of base models perform as best as possible. The final result is shown in Table 1.

3 Experimental Results

In this section, we will describe how the whole experimental part is done, and the main focus will be on the part that implements the model. The experimental part will be divided in to 5 parts as follows.

3.1 Datasets

In raw dataset, there are many descriptions about the news from different part such as Geography, Entities, Time, Narrative, Overall, Style, Tone. However, as the issue overview said, the annotation task consists of carefully reading each of the two news articles in a pair and selecting the Overall similarity score. As written in the description, systems will be evaluated on their ability to estimate the Overall similarity between two pairs of news stories, not any of the other scores. So we focus on the relationship between the Overall and the sentences, we use sentences separately.

3.2 Evaluation Metrics

In the evaluation dataset, we find that the evaluation dataset is mixed with many languages that does not appear in the training set, such as Chinese, so we try to add some languages that appear in the evaluation dataset in the text back translation. We use Google's translation API, we translate non-Chinese source language into Chinese, and increase the amount of train data through this form The submissions were scored using Pearson's correlation with the 'Overall' column. We use Pearson's correlation as our evaluation metrics. The definition of Pearson's correlation is as follows:

$$\rho_{x,y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}} \tag{1}$$

where the X is the predicted value, and Y is the ground-truth value. Further, mean squared error is calculated as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \overset{\wedge}{y_i}\right)^2 \tag{2}$$

where y_i is the predicted value, and $\overset{\wedge}{y_i}$ is the ground-truth value.

3.3 Implementation Details

The train data is split into 3 parts, the base data which is for the part of base regressor, the ensemble data which is for the part of ensemble learning and the test data for the final test, and the test data is used in Table 1. According to the paper (Sun et al., 2020), we truncate the middle part of the text of tokens larger than 512.

In the part of base regressor. At first, we clean the data, fill and delete the missing values in the dataset. The cleaned data is split to train data, validation data and test data. The raw sentences are put into the tokenizer that the tokenizer is corresponding to each model such as BertTokenizer and the tokenizer uses the upstream model to complete the tokenization. After that, we use tf.data.dataset to wrap the tokenizer so that the it can be used by regressor. Then The transformer-based model is used as our regressor such as BertForSequenceClassification, when the parameter num_labels=1, it can be used as a regressor. Adam (Kingma and Ba, 2017) is chosen as our optimizer and MSE is used as the loss function and Pearson as evaluation metrics to train the model. The validation data is used in each epoch to judge the performance of model. After training, we use the trained model to make predictions. The test data is used to detect which model perform better. We choose the best model and save it. In addition, we start to tune the parameters such as learning rate, weight decay and epochs. Because of the limit of the memory, we set batch size to 8 so that the model can run smoothly. We use grid search so that we can find the optimal parameters accurately and quickly.

3.4 Parameters Fine-tuning

WeightsBiases is a visualization tool to supervise the model training process. When tuning the parameters, we use it so that we can record the change curve of the parameters and easy to find optimal parameters (Wang et al., 2022). After tuning the parameters, we use test data to test. Finally, we set learning rate to 1e-5, set weight decay to 1e-6, set epochs to 50. And the C parameter of the SVR is set to 10, and the C parameter is essentially a regularisation parameter, which controls the trade-off between achieving a low error on the training data and minimising the norm of the weights.The kernel parameter is set to “linear”. We show the adjustment process of our parameters through two line graphs fig 2 and fig 3.

3.5 Comparative Results

We use the test data which is split from train data. We use this test data to calculate the Pearson score and MSE for each model's predictions, and the result is shown in Table 1.

We submit these models to the organizer. But the method of stacking doesn't achieve a good result. The model of BERT gets the best score in this competition which the score is 0.715. After submitting our final prediction, the best score is obtained by BERT instead of stacking. We submit these models to the final evaluation, but the only scores returned to us are BERT and stacking. The model of BERT got 0.715, the model of stacking only got 0.464. The reason why we set the topic as stacking ensemble learning is because we spend most of our time on it during the entire competition, and we think it does achieve better results on the training set, so we set the topic to stacking ensemble learning. And we reflect that it may be caused by insufficient generalization ability of our base model or meta-model. It may also be caused by insufficient differences in the base model during

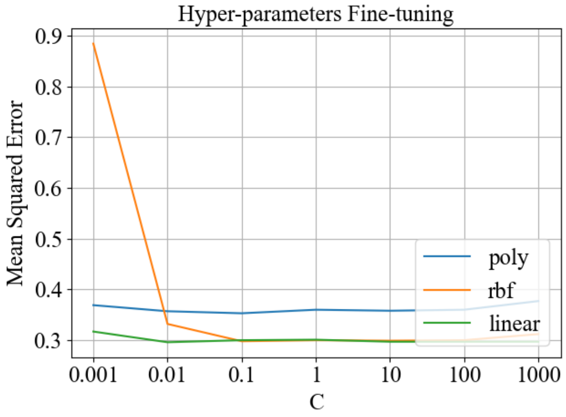

Figure 2: The performance of different parameters on MSE.

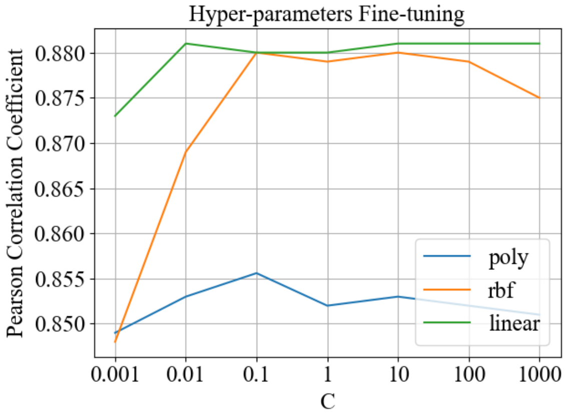

Figure 3: The performance of different parameters on PEARSON.

stacking, or it may be that the selected parameter adjustment method does not make the parameters optimal. This is what we reflect on after getting the feedback.

4 Conclusions

In this paper, we are participating in SemEval-2022 Task 8 (Chen et al., 2022). In this task, we perform regression prediction on the similarity of multilingual news articles, and we use various methods such as BERT, ALBERT, RoBERTa, DistilBERT, and the stacking method built with them as the base model. The model we proposed can effectively predict this task. Among the multiple models we submitted, the BERT model we finally submitted achieved the best score with a score of 0.715, ranking 21st in the leaderboard. At present, in terms of deep learning, the processing methods of multilingual texts have not been widely popularized. So in the future, we hope to go further in the processing of multilingual texts.

References

Xi Chen, Ali Zeynali, Chico Q. Camargo, Fabian Flock, Devin Gaffney, Przemyslaw A. Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 Task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129(6):1789–1819. ArXiv: 2006.05525.

Bo Huang, Yang Bai, and Xiaobing Zhou. 2021. hub at SemEval-2021 Task 2: Word Meaning Similarity Prediction Model Based on RoBERTa and Word Frequency. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 719–723, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. ArXiv: 1412.6980.

Jun Kong, Jin Wang, and Xuejie Zhang. 2022. Hierarchical BERT with an adaptive fine-tuning strategy for document classification. *Knowledge-Based Systems*, 238:107872.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv:1909.11942 [cs]*. ArXiv: 1909.11942.

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:1907.11692 [cs]*. ArXiv: 1907.11692.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*. ArXiv: 1910.01108.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to Fine-Tune BERT for Text Classification? *arXiv:1905.05583 [cs]*. ArXiv: 1905.05583.

D. Viji and S. Revathy. 2022. A hybrid approach of Weighted Fine-Tuned BERT extraction with deep

Siamese Bi – LSTM model for semantic text similarity identification. *Multimedia Tools and Applications*.

Jin Wang, You Zhang, Liang-Chih Yu, and Xuejie Zhang. 2022. Contextual sentiment embeddings via bi-directional GRU language model. *Knowledge-Based Systems*, 235:107663.

Xi Yang, Xing He, Hansi Zhang, Yinghan Ma, Jiang Bian, and Yonghui Wu. 2020. Measurement of Semantic Textual Similarity in Clinical Texts: Comparison of Transformer-Based Models. *JMIR Medical Informatics*, 8(11):e19735.

You Zhang, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2021. MA-BERT: Learning Representation by Incorporating Multi-Attribute Knowledge in Transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2338–2343, Online. Association for Computational Linguistics.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: Systems for Measuring Semantic Text Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada. Association for Computational Linguistics.