A Multi-model Fusion Approach for Product Classification and Product Substitute Identification on Shopping Queries Data

Speaker: Yanbo J. Wang

Longying Zhida (Beijing) Technology Co., Ltd.

CONTENTS

Abstract

The main goal of this competition is to establish the ranking strategy, which is divided into three sub-tasks. Our team mainly designed solutions for task 2 and task 3. The goal of task 2 is to classify the list of results of a given query and multiple products retrieved for the query into four categories: Exact, Substitute, Complement, or Support. The goal of Task 3 is the ability to detect the Substitute category mentioned in Task 2.

We used the multi-classification model of task 2 to accomplish the objectives of two tasks at the same time. In order to achieve better results, we design the downstream task structure independently for fine-tune, and different post-processing units for two tasks. According to the experimental results, the optimal three models were selected as the final solution.

Data preprocessing

Since the product context has multiple fields (title, brand, etc.), we use the [SEP] token to segment the text of each field, connect it to the model input text, and use the [CLS] token vector as the potential feature of the data.

Input token connected as:

[CLS] <query_content> [SEP] <title_content> [SEP] <bullet_point> [SEP] <brand> [SEP] <color_name> [SEP] <locale> [SEP] <description>

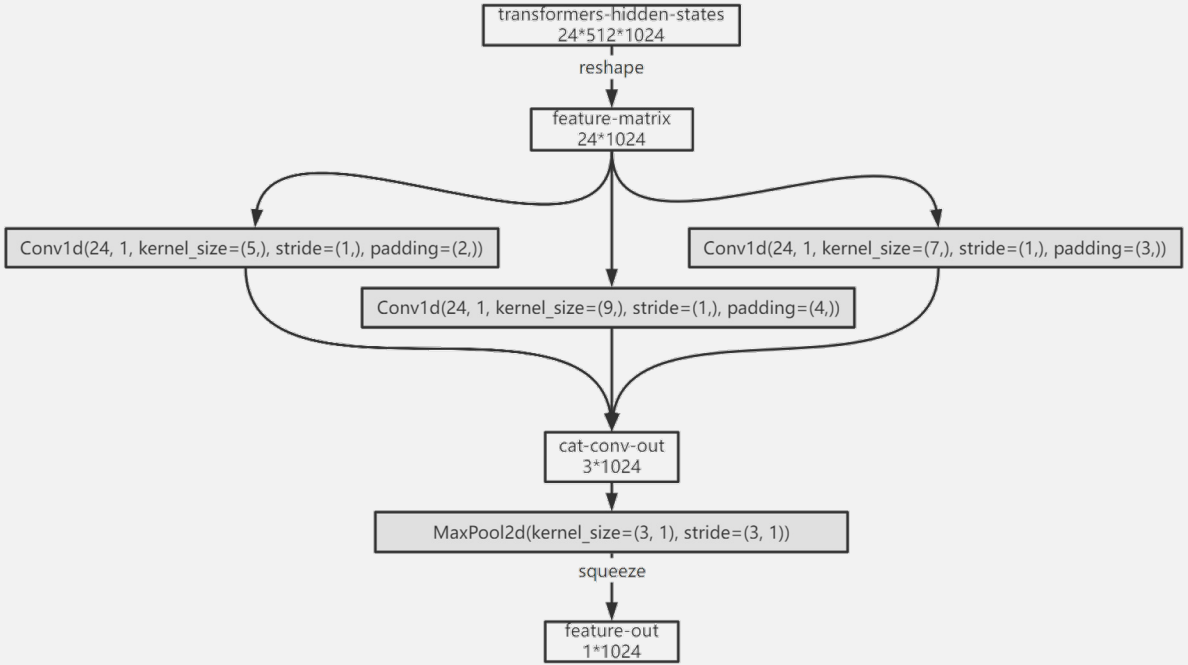
Model selection

In order to compare the effect of the pre-trained models, we only control the difference between the pre-trained models and the classification models in this part of the experiment. The performance of the following models on the validation set and test set is statistically analyzed, and three pre-trained models, XLM-Roberta-Large, InfoxLm-Large and Rembert, are selected.

model_name	oof_score	sub_score
Multilingual-MiniLM-L12-H384	0.72018	0.723
bert-base-multilingual-cased	0.72862	0.734
infoxlm-base	0.73432	0.742
xlm-roberta-large	0.7566	0.76
rembert	0.7561	0.759
twitter-xlm-roberta-base	0.7312	0.738
infoxlm-large	0.7554	0.759
roberta-large-us	0.7686	\

Model structure

We use the features of the 0th token of all hidden layers output by the pre-trained model, a total of 24 feature vectors are connected into a 24 * hidden_size feature matrix, and then use three convolution kernels with the sizes of 5 * 24, 7 * 24 and 9 * 24 respectively to extract features. After maximum pooling, a 1 * hidden_size feature is obtained for classification. The model results of this part are shown in the following figure:



Trick on training

In order to optimize the prediction effect of a single model, we tested a variety of training techniques. Here, the XLM-Roberta-Large model structure is uniformly used to record the performance of each trick on the validation set and the test set. According to the actual effect, we combined MLM task pre-training and pseudo-labeled TRICK to train the model.

Trick	oof_score	sub_score
no-trick	0.7547	0.8087
Pretrain-MLMTask	0.7594	0.8113
Adversarial-Training	0.7589	0.8082
Pseudo-Labelling	0.7596	0.8117
Focal-Loss	0.7429	0.8032
Label-Smooth	0.7524	0.8083

Trick on training

- **Pseudo-Labelling**

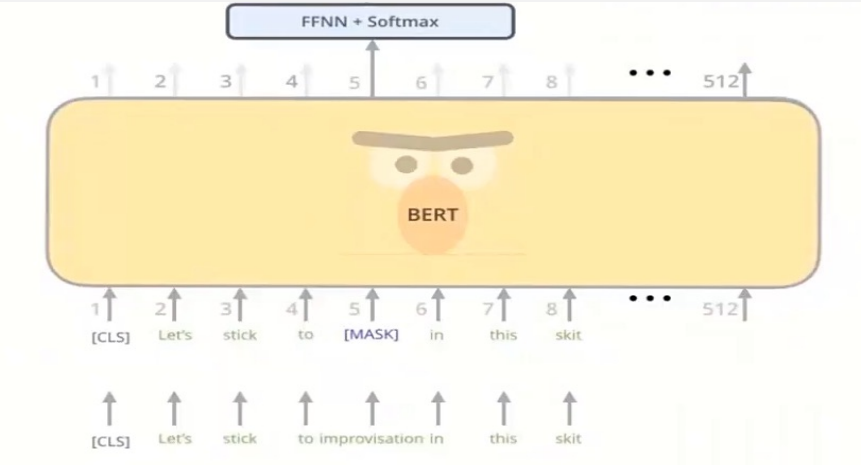
We use the XLM Roberta large model, which performs best in the first round of model selection experiments, to predict the public test set data, and then sort according to the maximum value of the probability vector, find a threshold, select a part of the public test set data with high confidence as pseudo label data, and add it to the training process of the model.

In order to determine the appropriate threshold, we use the same method to sort the verification set, select 10000 continuous predictions, and slide the calculation accuracy. When the accuracy is close to the overall verification set, we calculate the average value of the maximum probability of the current region as the threshold of the segmentation verification set. Through such segmentation, pseudo tag data can be selected as much as possible while ensuring the accuracy of pseudo tag data.

Trick on training

- **Pretrain-MLM Task**

Using all the data of this task, on the basis of the original pre-training model, the model is pre-trained again, so as to better fit the pre-training model of this task data. The pre-training is optimized only for MLM Task, and the training data is constructed using 30% Mask proportion. The original XLM-Roberta-Large and InfoxLm-Large were pre-trained, and the Rembert model was not pre-trained due to time and equipment reasons.



Model fusion

- **Task2**

All models were trained using task 2 data, and the input was more consistent with the data distribution of task 2, so the model fusion was carried out using the weighted average method.

Logits = 0.31* xlm-Roberta-large + 0.31* InfoxLm-large + 0.38* Rembert

- **task3**

In task 3, the best average threshold of the validation set was used to select Substitute data. The minimum probability threshold classified as Substitute was set in the validation set of three models, and then the grid search method was used to search with an interval of 0.001. It is used to process the probability results after the average fusion of the three models in the test set. Finally, the optimal average threshold was found to be 0.552.

Inference speed up

In order to get the prediction results of the three models within the specified time, we mainly used two methods.

- Semi-precision FP16 to make the final prediction.
- Sort the input data according to the length of tokens, and dynamically complete the input data according to the maximum length of a single bath in Dataloader, so as to reduce the unnecessary computation generated by large area zero complement pairs.

2022

Thanks for listening

LYZD-fintech