Second place solution of Amazon KDD Cup 2022: ESCI Challenge for Improving Product Search

Xiaolei Qin, Nan Liang, Hongbo Zhang, Wuhe Zou, Weidong Zhang

Model Architectures

The overall architecture of our method is shown in Figure 1. The first step is to obtain semantic representation of query and product by fine-tuning a pretrained language model where we use RoBERTa-large and DeBERTa-v3-large for English, mDeBERTa-v3-base and XLM-RoBERTa-large for Japanese and Spanish. Then we feed the last hidden states of the pretrained models into a Bi-LSTM layer. Finally, the output of Bi-LSTM is fed into the dense layer to get the prediction.

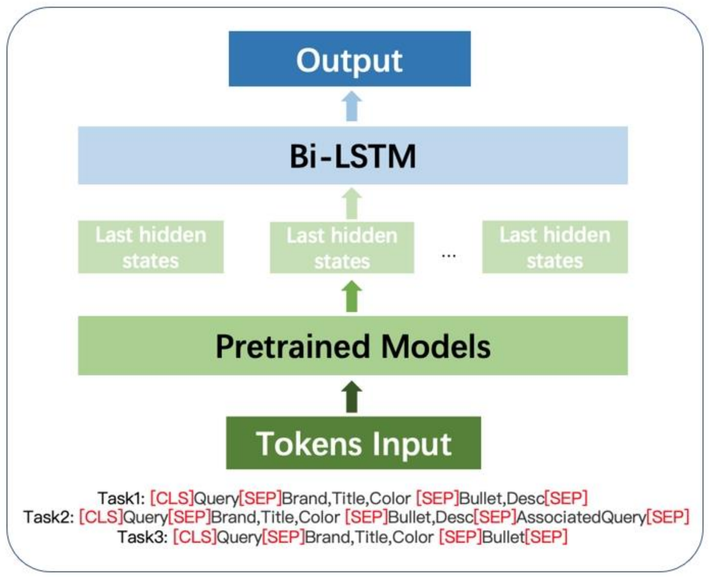

Figure 1. Overall Model Architecture

Train Strategies

- 5-fold cross-validation
  We train all the models by 5-fold cross-validation, but we do not use all the 5-fold results in the final submission due to the limitation of the online submission system
- Loss Functions
  we use MSE loss function (the four categories are mapped as 1.0, 0.1, 0.01 and 0). For task2 and task3, we use Cross Entropy loss function.
- Domain-adaptive pretraining
  a second phase of pretraining in domain leads to performance gains. In our methods, we continued to pretrain XLM-RoBERTa-large and RoBERTa-large using the competition dataset and achieved better result in task1.
- Ensemble
  For task1, we train several different versions of models and apply weigthed average to get better result. Due to the limitation of time the computation resources, we only use 3 of 5 folds models in task2 and 2 of 5 folds models in task3.
- Multi-task learning
  On task2 and task3, we use multi-task learning and achieve about 0.1%~0.2% improvement on local validation set. In task2, we jointly train four 2-class classifiers while training the main 4-class classifier. In task3, we jointly train the four classifier and the classifier of substitute label. The loss functions are as follows:

$$L_{task2} = 0.5 * L_{esci} + 0.125 * (L_e + L_s + L_c + L_i)$$

$$L_{task3} = 0.5 * L_{esci} + 0.5 * L_s$$

- Hyperparameters
  max length of input: 500
  learning rate of pretrained model: 7e-6 for DeBERTa-v3-large and 2e-5 for others
  learning rate of Bi-LSTM and MLP parameters: 1e-3
  batch size: 32
  freeze embedding parameters
  linear schedule with warmup: 5% warmup steps

Discussion

- Cross Features
  The dataset consists of Query-Product pairs, and there are some products that appears in different queries, which we believe should be useful.

Query	Product	Label
q1	p1	exact
q1	p2	substitute
q1	p3	irrelevant
q2	p4	complement
q2	p5	exact
q2	p1	exact

q2

Query	Product	Brand	Label
q1	p1	b1	exact
q1	p2	b2	substitute
q1	p3	b3	irrelevant
q1	p4	b1	complement
q1	p5	b1	exact
q1	p6	b2	exact

b1

Figure 2. Cross Features

- Data augmentation
  we experimented with translation and text generation to augment the data. Specifically, we translated the Japanese and Spanish dataset into English and also trained a BART text generation model to generate query from product title and bullet point, but these methods didn't work due to the extra noise probably introduced.
- Post-processing
  In this challenge, we find that given a query, the more non-Exact categories in the candidate products, the lower NDCG score is likely to be. These Query-Product pairs can be considered as hard cases. The intuition of post-processing is to pick out these hard pairs according to the score distribution predicted by stage-1 models, and then train a stage-2 model with the query and product title of multiple pairs, an example is shown in Figure3. Ultimately, we did not achieve higher NDCG score than stage-1 model in this way, so we think maybe better modeling methods and more experiments are needed.

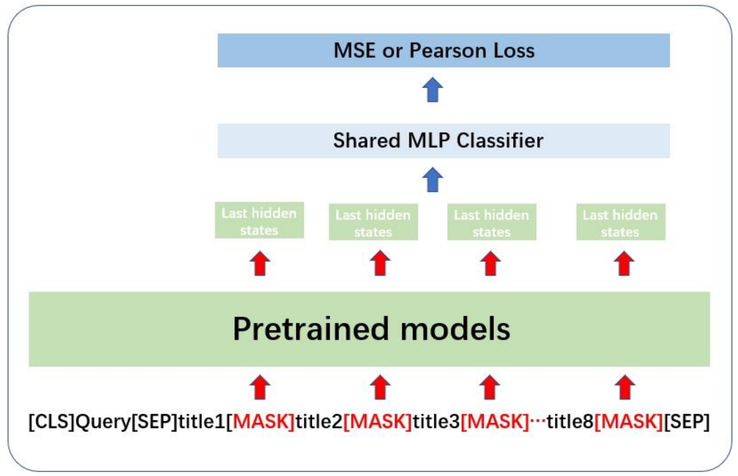

Figure 3. a possible post-processing architecture