CMB AI Lab at KDD Cup 2022 ESCI Task2 and Task3: A Domain Adapted PLM with Context Enhancement for Query-Product Classification

Haobo Yang
yanghaobo@cmbchina.com
China Merchants Bank

Shiding Fu
fu_shiding_joris@cmbchina.com
China Merchants Bank

Guidong Zheng
zhengguidong@cmbchina.com
China Merchants Bank

Junjie Wen
wenjunjieee@cmbchina.com
China Merchants Bank

Jinlong Li
lucida@cmbchina.com
China Merchants Bank

Xing Zhao
zhao_xing@cmbchina.com
China Merchants Bank

ABSTRACT

We present our solution on task2 and task3 of KDD Cup 2022 ESCI Challenge for Improving Product Search. Our approach mainly consists of four parts: 1) multilingual pretrained language model for context-aware embeddings and domain adaptive pretraining on this dataset for a more informative encoder. 2) product title concatenation to obtain adjacent products' information under each query. 3) further positive techniques such as data resampling and R-Drop. 4) model ensembling. The evaluation result of our approach achieves F1 score 0.8251 on task 2 , and F1 score 0.8734 on task 3 respectively, which ranks the fourth on the corresponding leaderboard.

KEYWORDS

product search, multilingual text classification, text similarity, consistency learning

ACM Reference Format:
Haobo Yang, Shiding Fu, Guidong Zheng, Junjie Wen, Jinlong Li, and Xing Zhao. 2022. CMB AI Lab at KDD Cup 2022 ESCI Task2 and Task3: A Domain Adapted PLM with Context Enhancement for Query-Product Classification. In *KDD Cup 2022 Workshop: ESCI Challenge for Improving Product Search, August 17, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

1.1 Task Description

In this challenge, we are expected to build new ranking strategies on a given shopping queries data set. Given a query and a result list of products retrieved for this query, we are expected to classify each candidate product in task 2, and to identify whether the classification of substitution is correct in task 3. Three different languages are covered in this challenge, including English, Spanish and Japanese. Task 2 emphasizes the classification of four labels named as "exact", "substitute", "complement" and "irrelevant", while task 3 focuses on the identification of "substitute".

KDDCup '22, August 17, 2022, Washington, DC, USA

Table 1: Detail of training data

Language	Queries	Judgements	Avg. Length
English	68,139	1,272,626	18.7
Japanese	10,624	249,721	23.5
Spanish	12,687	312,397	24.6

1.2 Data Analysis

There are 1,834,744 samples of query-product pairs in the data set with average sentence length of 20.1. Besides, the data set contains 1,815,216 products with the following fields: product title, product description, product bullet-point, product brand and product color. Some of the above fields could be vacant except the product title. The detail of the data set is listed in Table 1. Despite the data set of task 2 and 3 being the same, the two tasks are independent.

1.3 Task Solution

The foundation model we choose is InfoXLM-large[3] which has been proved to be a powerful multilingual pre-trained language model compared with other models like mBERT[5] and it can process all the languages existing in Task 2 and 3.Based on that, various strategies have been tested along with our exploration like data augmentaion, adaptive pretraining, utilization of adjacent products, R-Drop[9], etc. In this paper, we present an overall analysis of this competition and introduce our PLM based solution framework to the product search challenge.The related work of textual similarity and language model is briefly introduced in section 2. The detail of our method is presented in section 3. The experimental results are exhibited and analyzed in section 4. Section 5 summarizes the paper.

2 RELATED WORK

2.1 Text Similarity

In this challenge, the classification focuses on different relationships between a user query and each of its retrieved products, which can be measured by the text semantic relevance and could be viewed as a text similarity problem. Text similarity is an important research issue in natural language processing, which is quite widely used in many downstream tasks. Previously, statistical methods like BM25[15], Levenshtein distance[14] are widely used to evaluate the relevance between two texts. In order to compare

Table 2: Distribution gap between train set and dev set

	Exact	Substitute	Complement	Irrelevant
train	0.626	0.234	0.032	0.108
dev	0.738	0.168	0.021	0.073

the fine-grained differences between two texts, there are mainly two solution paradigms developed: representation based models and interaction based models. The early representation based models such as WMD[8] and SIF[1] are based on bag of words representation. Further more, other supervised models like DSSM[7], SiamLSTM[10], Sentence-BERT[13] could build more robust and precise embedding presentation of texts. Meanwhile,depending on interactive layer after embedding of two texts, interaction based models like MatchPyramid[12], ESIM[2], RE2[16] etc. could dig out deeper relation between two sentences.

2.2 Pretrained Language Model

With capturing latent information of sentences, transformer based pretrained language models have been verified to be the state of the art models for most of the NLP tasks. Due to the multilingual feature of this challenge, we consider multilingual pretrained language models like mBERT[5], ERNIE-M[11] and XLM-R[4]. Using multilingual masked language modeling and translated language modeling as pretraining tasks, InfoXLM-large[3] achieves SOTA performance on XTREME[6] leaderboard.

3 METHOD

3.1 Overall Architecture

Our baseline model is simply providing a pair of query and product title to InfoXLM[3] model and predicting its classification distribution with the [CLS] token representation. All optimizations we applied are based on this framework and the overall architecture of our approach is shown in Figure 1.Above our baseline model, we concatenated the product titles near the target product as supplementary information. According to our experiments, such additional input can improve the accuracy of classification. Furthermore, we changed the number of selected adjacent products and used resampling technique to train different models. After training with all the optimization methods discussed above, we made an ensemble of the best models on each fold for the final prediction.

3.2 Dataset Split

We merged the public data sets of task1 and task2 to get more data for training. Noticing that the task 2 test set does not contain any data from task1 public data and samples on task1 are obviously more difficult than those on task2, We randomly extracted 4000 samples for validation from task2 data set that does not appear on task1 data set , which ensures that our offline test performance can be closer to online scores. We didn't do any other data cleaning other than converting all texts to lowercase.

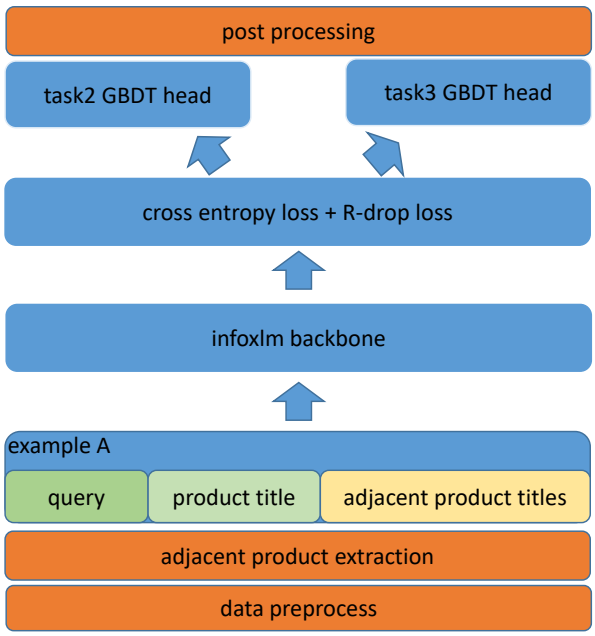

Figure 1: model architecture

3.3 Domain Adaptive Pretraining

This challenge focused on the scene of e-commerce. To capture the relevance between queries and products effectively, we firstly performed domain-adaptive pretrain tasks on the full training and test data. In additions, we also use the concatenation of query and product title pair with the exact label in the domain adaptive pretraining step.

3.4 Utilization of Adjacent Products

We suggest that the relevance of a product to a query is not only determined by the product itself, but also is affected by other retrieved products by the same query. At the same time, we found that the annotation standards for different queries seem to be inconsistent. For example, a product with its color different from the query's color could be labeled as exact, but in some other queries, the different colors could make a product labeled as substitute or irrelevant. Therefore, for each query and a target product, we selected 2-6 other retrieved products of this query near the target product as context information. The number is limited to the max length of the input texts of the model. We concatenate the query, the product titles and context with [SEP] to form the final input of the model. In this way, the model can distinguish the factors that really affect the correlation between a specific query and the corresponding products through the context information.

3.5 Consistency Learning

We introduced the idea of consistency learning to improve the robustness and generalization ability of the model. Consistency learning improves the overall performance of the model by constraining the inconsistency of high-level representations caused by certain aspects of the same sample, such as input representation,

Table 3: Model Used in Ensemble

Model Description	Task2 F1 Score(Public)
task2	
resample + R-Drop + 4 adjacent products	0.820
R-Drop + 4 adjacent products	0.821
R-Drop + 2 adjacent products	0.819
only task2 data + R-Drop + 4 adjacent products	-
task3	
resample + R-Drop + 2 adjacent products	0.819
resample + R-Drop + 4 adjacent products(another seed)	0.820

dropout, and model architecture.Specifically, We utilized R-Drop[9] technique to minimize the bidirectional KL-divergence between the output distributions of two sub models sampled by dropout.For this classification problem,cross entropy loss is used as the basic loss function, which is:

$$L_{CE}^{i} = -log(P_i(y_i|x_i)) \tag{1}$$

where the y_i, $x_i$means the label and the input of the i_{th} sample respectively. The loss of regularization term is described as the formula 2.

$$L_R^i = KL(P_1^i(y_i|x_i)||P_2^i(y_i|x_i)) + KL(P_2^i(y_i|x_i)||P_1^i(y_i|x_i)) \tag{2}$$

where KL denotes the Kullback-Leibler divergence, P_1^i means the distribution of the model output in the first forward pass. Our final loss is the combination of L_R and L_{CE}:

$$L = L_{CE1} + L_{CE2} + \alpha \cdot L_R \tag{3}$$

α is a hyper parameter which could be adjusted.

3.6 Data Resampling

There are two facts why we used resampling method. Firstly, we merged the data sets of task1 and task2; Secondly, some online test data appeared in task1 had been removed.Those resulted in changes of the distribution of data. The distributions of training set and validation set are shown in Table 2. Therefore, we resampled the training set according to the distribution of the validation set where the distribution is closer to the test set. The resampled data did not directly improve the performance of our single model, but it played a significant role in the phase of ensemble.

3.7 Ensemble

For task2, we trained four different models as shown in Table 3 and took their output probabilities as features to train a 4-class GBDT classifier. For task3, We utilized a 2-class GBDT classifier to fuse the outputs of the models above for task2 and another two models as shown in Table 3.

3.8 Inference Optimization

We needed to consider model inference acceleration due to time constraints at submission and multi-models ensemble that caused inference time to grow exponentially. The first optimization is to convert the float32 model parameters into float16 which can crop the model size into half. And then with the help of amp inference toolkit, the inference speed is doubled. Furthermore, we removed redundant operations, performed constant folding and used kernel fusion under onnxruntime framework. Finally, We achieved nearly three-times speed-up as shown in Table 4 during inference stage.

Table 4: Inference time Comparison

batch-size 128 1000times	min	avg	max	tp99
pytorch fp32	17.64ms	18.32ms	27.75ms	21.67ms
pytorch fp16	9.01ms	10.66ms	25.16ms	11.32ms
onnx optimization	6.29ms	6.46ms	23.26ms	8.60ms

Table 5: Single Model Performance

Model based on InfoXLM-large	Task2 F1 Score (Public)
w/ dataset split	
DAPT	0.810
DAPT + resample	0.811
DAPT + R-Drop	0.814
DAPT + resample + R-Drop	0.816
DAPT + R-Drop + 2 adjacent products	**0.819**
R-Drop + adjacent 2 products	0.813
w/o dataset split	
DAPT + R-Drop	0.812

4 RESULTS

4.1 Overall Performance

As shown in Table 7, our best single model achieved the F1 score of 0.821 on task2, and ensemble model F1 scored of 0.824 and 0.871 on public test set of task2 and task3 respectively. The effects on task2 of methods introduced in Section 3 are shown in Table 5, and we will introduce some strategies mentioned individually in the following subsections.

4.2 Dataset Split

Using the validation set obtained according to method mentioned in Section 3 enabled us to stop the training of the model more precisely and to select the model with the best online performance. As shown in Table 5, the replacement of validation set brought us an improvement of 0.002 points.

4.3 Domain Adaptive Pretraining

As shown in Table 5, compared with the model using domain adaptive pretraining (dapt) technique, the score of the model without dapt technique decreased by 0.006, from 0.819 to 0.813, which shows that there is a great difference between the data distribution of e-commerce scenarios and general scenarios. It also proves that dapt can alleviate this inconsistency and improve the classification effect of the model in the field of e-commerce.

4.4 Utilization of Adjacent Products

Based on our experimental results, the utilization of adjacent products improved our single model from 0.814 to 0.821. We further explored the impact of different numbers of adjacent products on the effect of the model, the results are shown in Table 6. All the scores of the experiments with adjacent products information are higher than the basic model with R-Drop, which proves that the

Table 6: Affects of Title Quantity in Context

# Titles in context	Task2 F1 Score (Public)
0	0.814
2	0.819
3	0.819
4	**0.821**
6	0.820

Table 7: Model Ensemble Performance

System	F1 Score (Public)
task2 single model	0.821
ensemble with naive average	0.823
ensemble with LightGBM	**0.824**
task3 single model	-
ensemble with naive average	-
ensemble with LightGBM	**0.871**

adjacent products are strong features for the model to calculate the similarity between query and product. We suggest the reason why the context title is effective is that these adjacent products provide more keyword information related to the query. The matching degree of target product with these keyword information can reflect the correlation degree between the target product and the query. It should be noted that the number of adjacent products is not the more the better. When the number increases to 6, the effect of the model is lower than that when the number is 4. We believe that as the number of products increases, the more query-irrelevant information is included in the titles of these products, which leads to a decrease in model effectiveness.

4.5 Consistency Learning

The R-drop technology helped us get the best single model scored 0.821 on task2 public test set. In fact, we have tried other consistency learning methods. For example, we used some data enhancement strategies such as translation, adding random noise, and then minimized the bidirectional KL-divergence between the output distributions of origin sample and enhanced sample like R-Drop. Among them, dropout strategy worked better than those other enhancement methods.

4.6 Ensemble

Due to the competition deadline, we didn't have enough time to try too many model-fusion strategies, other than averaging method and using GBDT. According to the Table 7, we can see that with the same models, the LightGBM method scores 0.001 higher than the simple average ensemble method, which proves the effectiveness of LightGBM.

5 CONCLUSIONS

In this paper, we introduced our solution for KDD Cup 2022 ESCI Challenge for Improving Product Search, including 4 parts: 1) A domain adaptive pretrained model which can capture the correlation effectively between a query and a product. 2) We proposed to take the adjacent products of the target product as an important feature to provide context information, so as to improve the classification performance of the model. 3) Using consistency learning techniques like R-Drop to improve model robustness. 4) Other positive strategies such as data resampling and model ensemble.

In the future, We will try knowledge distillation to reduce the impact of noise data, and we plan to further refine our approach from the perspective of cross-lingual text representation, considering that the dataset is multilingual.

REFERENCES

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

[2] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced LSTM for natural language inference. *arXiv preprint arXiv:1609.06038* (2016).

[3] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3576–3588. https://doi.org/10.18653/v1/2021.naacl-main.280

[4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[5] 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* (2018).

[6] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. *CoRR* abs/2003.11080 (2020). arXiv:2003.11080

[7] P. S. Huang, X. He, J. Gao, D. Li, and L. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information knowledge management*.

[8] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*. PMLR, 957–966.

[9] Xiaobo* Liang, Lijun* Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-Drop: Regularized Dropout for Neural Networks. In *NeurIPS*.

[10] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[11] Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674* (2020).

[12] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[14] E. S. Ristad and P. N. Yianilos. 1996. Learning string edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1996).

[15] A. Trotman, A. Puurula, and B. Burgess. 2014. Improvements to BM25 and Language Models Examined. In *Australasian Document Computing Symposium*. 58–65.

[16] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. *arXiv preprint arXiv:1908.00300* (2019).

Table 8: Hyperparameters tuned in our system

Parameter	Value
Epoch	2
Batch Size	64,128
Learning Rate	1e-5
Warm-up Steps	6000
Weight Decay	1e-8
Sequence Length	256,384
R-Drop α	0.5

A APPENDIX

Table 8 shows the implementation detail of our solution which could help reproduce our solution to the challenge.