A Domain Adapted PLM with Context Enhancement for Query-Product Classification*

Haobo Yang and Shiding Fu
China Merchants Bank

*works by H Yang, S Fu, G Zheng, J Wen, J Li and X Zhao

Outline

- INTRODUCTION
- MODEL ARCHITECHTURE
- TRAINING STRATEGIES
- RESULTS
- CONCLUSION

Introduction

We present our solution on task2 and task3 of KDD Cup 2022 ESCI Challenge for Improving Product Search.

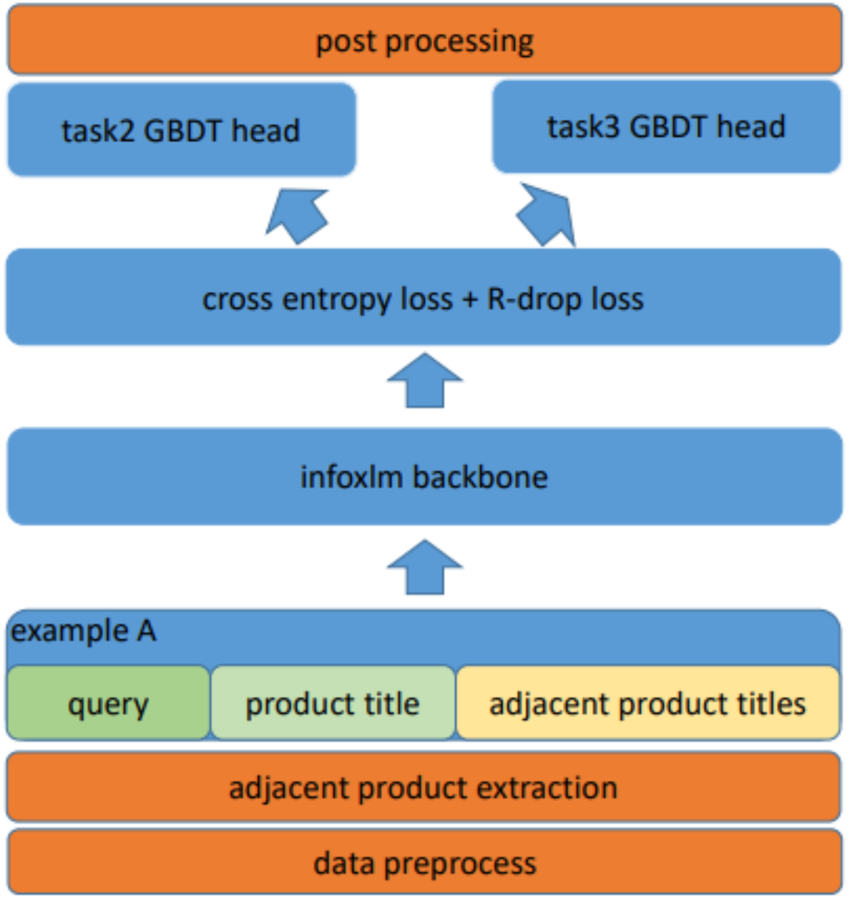
- both classification task
- sharing the same dataset
- The dataset is multilingual.

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

example_id	esci_label	substitute_label
example_1	exact	no_substitute
example_2	substitute	substitute
example_3	complement	no_substitute
example_4	irrelevant	no_substitute

Model Architechture

- Data preprocess
- Infoxlm-large as backbone
- Consistency loss
- Context Enhancement
- Same backbone, different GBDT heads



Training Strategies

- Dataset split

  - merge datasets of task1 and task2
  - extract dev set only task2
  - data resampling

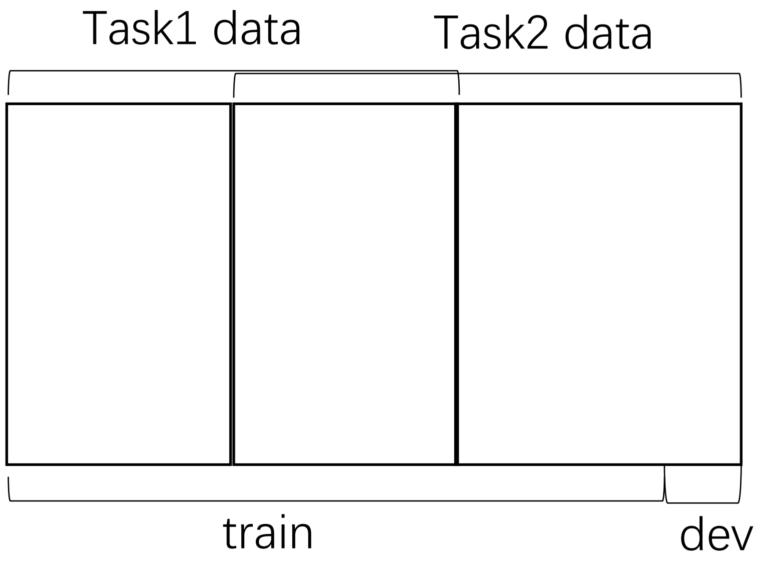


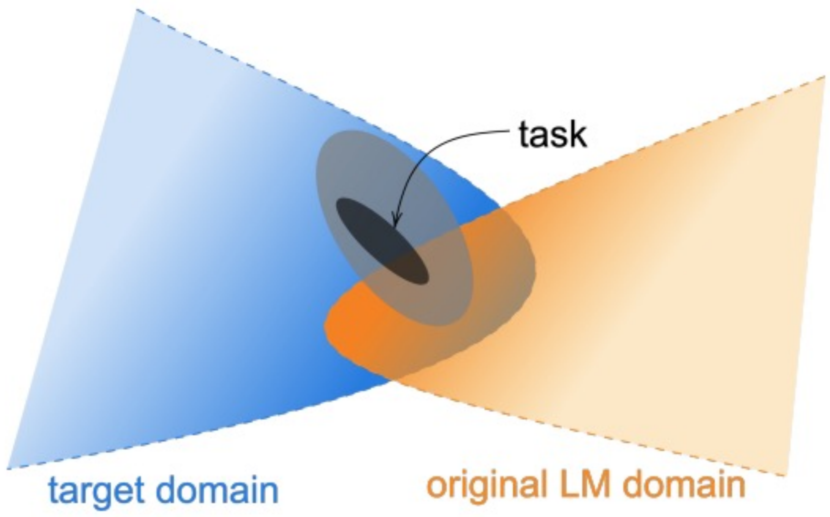
Table 2: Distribution gap between train set and dev set

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

Training Strategies

- Domain Adaptive Pretraining (DAPT)

  - single query
  - single product title
  - query-product pair labeled exact



Training Strategies

- Consistency Learning

  - original cross entropy loss:

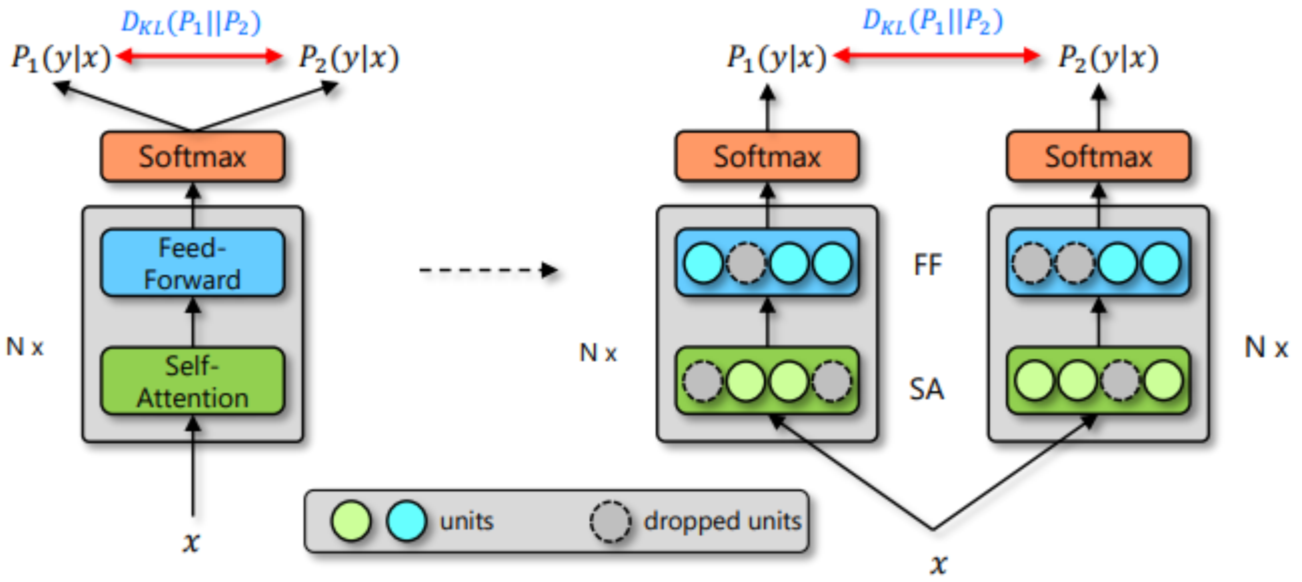
$$L_{CE}^{i} = -log(P_i(y_i|x_i))$$

  - consistency loss:

$$L_R^i = KL(P_1^i(y_i|x_i)||P_2^i(y_i|x_i)) + KL(P_2^i(y_i|x_i)||P_1^i(y_i|x_i))$$

  - final loss:

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



Training Strategies

- Utilization of Adjacent Products
  - classify target product with adjacent products under the same query
  - the input of infoxlm is like:

[CLS] query [SEP] target product title [SEP] product1;product2;⋯;[SEP]

- different number of adjacent products

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

Training Strategies

- Ensemble

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

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**

Training Strategies

- Inference Optimization

  - fp16
  - onnxruntime

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

Results

- Final ensemble score
  - 0.8251 rank 4_{th} on task2
  - 0.8734 rank 4_{th} on task3

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

Conclusion

- A domain adaptive pretrained model which can capture the correlation effectively between a query and a product.
- We proposed to take the adjacent products of the target product as an important feature to provide context information
- Using consistency learning techniques like R-Drop to improve model robustness.
- Other positive strategies such as data resampling and model ensemble

Thank you