A simple but effective solution for Task 1 of KDD Cup 2022 Challenge on improving product search

Jinrui Liang
liangjinrui@corp.netease.com
Netease Games Operation Department
Guangzhou, China

Yali Shangguan
shangguanyali@corp.netease.com
Netease Games Operation Department
Guangzhou, China

Zhaohao Liang
gzlzh@corp.netease.com
Netease Games Operation Department
Guangzhou, China

ABSTRACT

In this report, we present our solution of KDD Cup 2022 ESCI Challenge for Improving Product Search. The model is based on the DeBERTaV3-large and the mDeBERTaV3-base. Moreover, we adapt a regression for targeting. In the training, both the English DeBERTaV3-large, Japanese and Spanish mDeBERTaV3-base model are trained by 2-fold cross-validation. In the final submission, we take the average of the outputs of the DeBERTaV3-large in English and the mDeBERTaV3-base models in Japanese and Spanish. In the competition, our team achieved a nDCG of 0.9008 on test set, which placed in 8th in task1.

KEYWORDS

ESCI, Product Search, DeBERTa, mDeBERTa

ACM Reference Format:
Jinrui Liang, Yali Shangguan, and Zhaohao Liang. 2022. A simple but effective solution for Task 1 of KDD Cup 2022 Challenge on improving product search. In *KDD Cup 2022 Workshop: ESCI Challenge for Improving Product Search, August 17, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 3 pages.

1 DATASETS

The official data provided in task 1 is smaller than in task 2 and task 3 [2]. The data set contains a list of query-result paired with annotated E/S/C/I labels, and it includes queries from English, Japanese and Spanish. Every example contains the following fields: *example_id*, *query*, *query_id*, *product*, *product_title*, *product_description*, *product_bullet_point*, *product_brand*, *product_color*, *product_locale*, and *esci_label*.

In order to increase the generalization of the models, we add the supplementary data from task 2 and task 3 to task1. All the data are processed in this way.

2 FEATURES

We remove the unnecessary text through regularization like < [^>]+ > and concatenate text like the following sequence *product_title*; *product_description*; *product_bullet_point*; *product_brand*; *product_color_name*. In this way, the *product_title* can provide more difference.

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

Table 1: Relationship between Product type and Gain

Product type	Gain
Exact(E)	1.0
Substitute(S)	0.1
Complement(C)	0.01
Irrelevant(I)	0.0

3 MODELS

We describe our models for Task1 challenge in this section.

3.1 Problem Definition

For this ranking problem, we want to get the correct sorted list of product items for the query, where the gains of different product types are different. Generally, we need to accurately judge the gain of the candidate product. We treat the problem as a regression problem, where inputs of the model are the query and the candidate product, and the output are the predicted gain for the <query, product> pair. In the validation phase, for the query, we use the predicted gain to rank candidate products to form the ranked list. The relationship between product type and gain is shown in Table 1.

Shortly, the inputs of our model is *query* + <SEP> + *product title* + ' ' + *product description* + ' ' + *product bullet point* + ' ' + *product brand* + ' ' + *product color name.*

We use '' to fill in missing values. The output of the model is a value, which is regarded as the predicted gain for the input <query, product>.

3.2 Dataset Split

According to the query, the data set is divided into training set and validation set, which means that queries in the validation set will not appear in the training set to prevent query leakage. If the partition is not conducted according to the query, the experimental results will not accurately reflect the performance of the model. We set the proportion of the validation set to 0.1 in normal model training, where k-fold is not used. We also use the data that appear in Task 1 from Task 2 as supplementary data (data-sup), which are only used in the training set.

3.3 Model Selections

In the modeling phase, we use a simple but effective unified paradigm, namely BERT Encoder + Regression Layer, where the regression layer is a normal fully connection layer. For different locales,

Table 2: Model Selections

Locale	Selected Model
us	DeBERTaV3-large [1]
es	mDeBERTaV3-base [1]
jp	mDeBERTaV3-base [1]

Table 3: Training settings for DeBERTaV3-large

Hyper-parameters	Value
epochs	3
learning rate	8e-6
batch size	32
random state	42
learning rate scheduler	linear scheduler
weight decay	0.01
Adam betas	(0.9, 0.999)
Adam eps	1e-6
grad max norm	1.0
warmup steps	100
dropout rate	0.15
validation steps	500

Table 4: Training settings for mDeBERTaV3-base

Hyper-parameters	Value
epochs	3
learning rate	2e-5
batch size	32
random state	42
learning rate scheduler	linear scheduler
weight decay	0.01
Adam betas	(0.9, 0.999)
Adam eps	1e-6
grad max norm	1.0
warmup steps	100
dropout rate	0.1
validation steps	500

we select different BERT Encoders. We choose different models for 3 locales and finetune them for the ranking problem. Summary information is as shown in Table 2.

4 TRAINING SETTINGS

We strictly follow the implementation described in the original paper [1] for the settings of the hyper-parameters. At the same time, some adjustable parameters are modified. We verify the model in the fixed step and select the best model. Specific settings are show in Table 3 and Table 4.

Table 5: Local results of us locale

Model	Fold	Val nDCG
baseline	-	0.8972
baseline-sup	-	0.8998
baseline-2fold	0	0.8975
baseline-2fold	1	0.8967
baseline-2fold	ensemble	0.9005
baseline-sup-2fold	0	0.9013
baseline-sup-2fold	1	0.9007
baseline-sup-2fold	**ensemble**	**0.9021**

Table 6: Local results of es locale

Model	Fold	Val nDCG
baseline	-	0.8997
baseline-sup	-	0.9011
baseline-2fold	0	0.8995
baseline-2fold	1	0.8989
baseline-2fold	ensemble	0.9010
baseline-sup-2fold	0	0.9051
baseline-sup-2fold	1	0.9043
baseline-sup-2fold	**ensemble**	**0.9053**

Table 7: Local results of jp locale

Model	Fold	Val nDCG
baseline	-	0.8971
baseline-sup	-	0.8974
baseline-2fold	0	0.8984
baseline-2fold	1	0.8969
baseline-2fold	ensemble	0.8985
baseline-sup-2fold	0	0.8993
baseline-sup-2fold	1	0.8989
baseline-sup-2fold	**ensemble**	**0.9001**

Table 8: Online(Final) results

Model	Public Test nDCG	Private Test nDCG
baseline(official)	0.8503	-
baseline(ours)	0.8968	-
baseline-sup-us	0.8981	-
baseline-sup-all	0.8983	-
baseline-2fold-us	0.8983	-
baseline-2fold-all	0.8988	-
baseline-sup-2fold-us	0.9009	-
baseline-sup-2fold-es-jp	0.9008	-
baseline-sup-2fold-all	**0.9012**	**0.9008**

5 EXPERIMENTAL RESULTS

Due to limited resources and time for verification, we set k=2 in k-fold cross validation and model ensemble (average results for each fold). The baseline is DeBERTaV3-large for the us locale, while the baseline is mDeBERTaV3-base for the es locale and the jp locale. '-' in *Fold* column means that the model do not use k-fold, and '-' in *Private Test nDCG* column means that we can not get the result. Our final model is **baseline-sup-2fold-all** in table 4, which is simple but effective.

REFERENCES

[1] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *CoRR* abs/2111.09543 (2021). arXiv:2111.09543 https://arxiv.org/abs/2111.09543

[2] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. arXiv:2206.06588