A Semantic Alignment System for Multilingual Query-Product Retrieval

The first-place entry for Query-Product Ranking of ESCI Challenge at KDD Cup 2022

Qi Zhang, Zijian Yang, Yilun Huang, Ze Chen, Zijian Cai, Kangxu Wang, Jiewen Zheng, Jiarong He, Jin Gao*
{zhangqi21,yangzijian,huangyilun,jackchen,caizijian01,wangkangxu,zhengjiewen,gzhejiarong,jgao}@corp.netease.com
Interactive Entertainment Group of Netease Inc.
Guangzhou, China

ABSTRACT

This paper mainly describes our winning solution (team name: *www*) to Amazon ESCI Challenge of KDD CUP 2022, which achieves a NDCG score of 0.9043 and wins the first place on task 1: the query-product ranking track.

In this competition, participants are provided with a real-world large-scale multilingual shopping queries data set and it contains query-product pairs in English, Japanese and Spanish. Three different tasks are proposed in this competition, including ranking the results list as task 1, classifying the query/product pairs into Exact, Substitute, Complement, or Irrelevant (ESCI) categories as task 2 and identifying substitute products for a given query as task 3.

We mainly focus on task 1 and propose a semantic alignment system for multilingual query-product retrieval. Pre-trained multilingual language models (LM) are adopted to get the semantic representation of queries and products. Our models are all trained with cross-entropy loss to classify the query-product pairs into ESCI 4 categories at first, and then we use weighted sum with the 4-class probabilities to get the score for ranking. To further boost the model, we also do elaborative data preprocessing, data augmentation by translation, specially handling English texts with English LMs, adversarial training with AWP and FGM, self distillation, pseudo labeling, label smoothing and ensemble. Finally, Our solution outperforms others both on public and private leaderboard.

KEYWORDS

Shopping Queries Data Set, Query-Product Ranking, KDD Cup, Multilingual Language Model

ACM Reference Format:
Qi Zhang, Zijian Yang, Yilun Huang, Ze Chen, Zijian Cai, Kangxu Wang, Jiewen Zheng, Jiarong He, Jin Gao. 2022. A Semantic Alignment System for Multilingual Query-Product Retrieval: The first-place entry for Query-Product Ranking of ESCI Challenge at KDD Cup 2022. In *KDD Cup 2022 Workshop: ESCI Challenge for Improving Product Search, August 17, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 4 pages.

1 INTRODUCTION

Amazon ESCI Challenge [2] for Improving Product Search of KDD CUP 2022 is aiming to improve the customer experience and their engagement when searching for products. The primary objective of this competition is to build new ranking strategies and, simultaneously, to identify interesting categories of results by using their real-world Shopping Queries Dataset.

1.1 Dataset Description

The provided Shopping Queries Dataset [17] involves 3 languages: English (about 54.5% of the total training sets) , Japanese (about 26.5% of the total training sets) and Spanish (about 19% of the total training sets). In online shopping applications, the notion of binary relevance limits the customer experience. To keep high accuracy in ranking, the competition organizers break down relevance into the following four classes (ESCI) which are used to measure the relevance of the items in the search results. Exact (E) and Substitute (S), stands for the item is relevant and somewhat relevant to the query respectively. Complement (C) and Irrelevant (I) denotes respectively the item does not fulfill the query and the item is irrelevant.

For each query, the dataset provides a list of up to 40 potentially relevant product results, together with ESCI relevance judgements and an annotated locale label. For each product, the dataset provides associated information such as product title, description, bullet points, brand, color and locale.

A total of about 1.2 million products and 2 million query-product pairs are provided in this challenge, which are used for model training and offline validating. Online test set is split into public and private ones, and the final ranking is based on the score on the private leaderboard.

As shown in Table 1, the label distribution is very imbalanced. Most of the labels are Exact with the percentage up to 62.78%, while Complement class only accounts for 3.16%, Substitute and Irrelevant class account for 23.28% and 10.78% respectively. And according to our statistics, 54% of product brands focus on providing only one product, while only 7.1% of brands provide more than 10 products. At the same time, more than 80% of the color names are only customized for a single product. Such results help us to further understand and quantify the characteristics and distributions of the corpora provided in this competition.

1.2 Task Description

There are three tasks in this competition and we mainly involved in task 1: Query-Product Ranking. The goal of this task is to rank a

[*] Corresponding author

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

Table 1: Distribution of ESCI Labels

# (E) Exact	62.78%
# (S) Substitute	23.28%
# (C) Complement	3.16%
# (I) Irrelevant	10.78%

list of matched products of a specified user query. NDCG is used as relevance metric in this task, with a class gain of 1.0, 0.1, 0.01, 0.0 setting for ESCI respectively. The input for this task is a list of queries with their product identifiers. And the participants is asked to build a system to sort the product candidates, with the most relevant product in the first row and the least relevant product in the last.

2 RELATED WORK

Our work is mainly related to the pre-trained LMs and some specific strategies on Learning-to-Rank(LTR) systems, such as adversarial training, model ensemble and so on.

2.1 Cross-Encoder Models

Neural approaches have greatly improved the information retrieval results in recent years. Prior to this, similarity metrics primarily rely on keyword matching, with some limited thesaurus and phrase-based expansion. BERT[1, 8] uses Cross-Encoder architecture to achieve further improvements in the field of text understanding by passing the query and product simultaneously to the transformer networks and producing an output representation that indicates the similarity of the input pairs.

2.2 Multilingual Language Model

Defining textual features in a cross-lingual representation space has always been a challenge. The more languages there are, the confusing the representation contents will be. XLM [13] use Byte-Pair Encoding that splits the input into the most common sub-words across different languages instead of using word or characters as the input of the model. On the other hand, the Translation Language Modeling (TLM) task also increases the ability of contextual encoding. Nowadays, a set of large-scale Transformer-based pre-trained language models, such as RemBERT[5], XLM-RoBERTa[6], InfoXLM[4] and mDeBERTa[10, 11] have created new state of the art in many downstream fields. It turns out that training cross-lingual language models can improve performance on many NLP tasks.

2.3 Learning to Rank

Learning to rank (LTR) is a class of algorithmic techniques which are used to solve ranking problems in search relevancy[3]. For a specific query, we can use the model to do the ranking so that the relevant products will be ranked above the non-relevant ones. During the modeling process, query embedding and product embedding are concatenated as input to refine the self-attention mechanism for learning semantic alignments between queries and product descriptions. It is significant for the ranking task to establish a strong semantic alignment between the queries and the products[16].

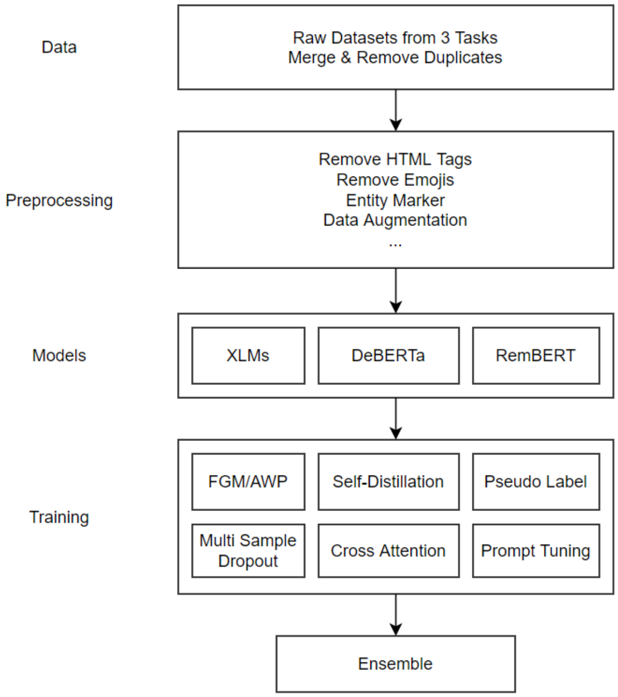

Figure 1: An overall framework and pipeline of our solution

3 METHODOLOGY

Our solution to this task mainly consists of 3 parts, which are data preprocessing, model training and ensemble. The overall framework is shown in Figure 1.

3.1 Data Processing

Given that datasets from the 3 tasks are exactly in the same format, we use all of the data from 3 tasks to train the model for task 1. The raw data is quite noisy containing many useless html tags, symbols and emojis, so we do some data cleaning work before model training by simply remove these trivial stuffs. After data cleaning, we use Google API to translate all of the data into English, Spanish and Japanese separately to do data augmentation.

We also use typed entity marker[20] to incorporate the NER information into the input of models. We add special tokens [TYPE], [/TYPE] near the entities in input text, where TYPE is the entity type recognized by a named entity tagger. For example, given the query “I want to buy an iPhone 8 Plus”, it will be modified to “I want to buy an [Product] iPhone 8 Plus [/Product]”.

3.2 Model Architecture

The single model architecture is shown in the Figure 2, we use the cross-encoder architecture based on DeBERTa, XLMs, and RemBERT. For the downstream task, [CLS] embedding is used to do the ESCI 4-class classification. We concatenate the query and product information in the way of “[CLS]query[SEP]color:<color> brand:<brand> description:<title+bullet_point+description>[SEP]” before feeding into the models.

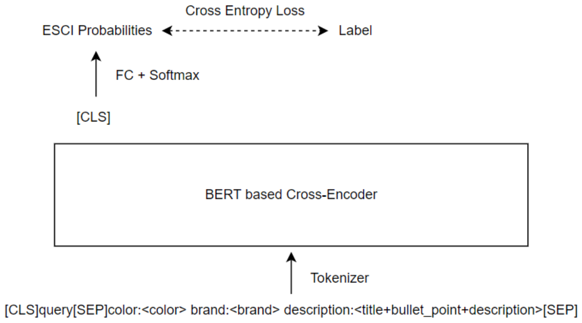
Cross Entropy Loss
ESCI Probabilities
Label
FC + Softmax
[CLS]
BERT based Cross-Encoder
Tokenizer
[CLS]query[SEP]color:<color> brand:<brand> description:<title+bullet_point+description>[SEP]

Figure 2: Model architecture for a single model

3.3 Training Details

We found that lots of mislabeling cases may exist in the dataset when doing bad cases analysis based on our local validation set in ESCI classification task. We realize that the data, labeled by crowdsourcing, could be quite noisy inevitably. To avoid misleading by the mislabeling, we use label smoothing to make the model less confident to the labels, which is proven to be effective. Training models in complex and ambiguous contexts can badly affects its generalization. To improve the model robustness, some training tricks are adopted during our experiments.

3.3.1 Self Distillation[19]. Given that the data is labeled by crowdsourcing, it could be quite noisy for the model to extract the real information. To make the model more robust, we use self-distillation training to further boost our single model. The model itself is used as its own teacher to do distillation training. To be specific, we use 3-fold bagging training and make prediction on the out-of-fold datasets to generate the soft labels for all of the training examples. And then we merge the soft labels with the ground true hard labels with weights 0.3 and 0.7 to get the new training labels: $y\_new = 0.7 * hard\ \ labels + 0.3 * soft\ \ labels$

We also tried to use two loss functions to compute the soft label loss and hard label loss separately and sum it with different weights. Unfortunately, it didn't work better than directly merge the labels as mentioned above.

3.3.2 Pseudo Label[14]. We also use our trained models to generate pseudo labels from the public test set to do further training. To avoid making the training data more noisy, only samples from the public test set with predicted probabilities above 0.7 are used as pseudo labels, as shown in Figure 3. And soft labels work better than hard labels during most of our experiments, we guess that hard labels may increase the risk of overfitting to some extent.

3.3.3 Multi Sample Dropout[12]. Dropout is a simple but efficient regularization technique for achieving better generalization. By combining a group of dropout layers with different dropout ratios, multi-sample dropout can achieve further improvement for the model. In this scenario, we use multi-sample dropout before the output layer to make our model more robust.

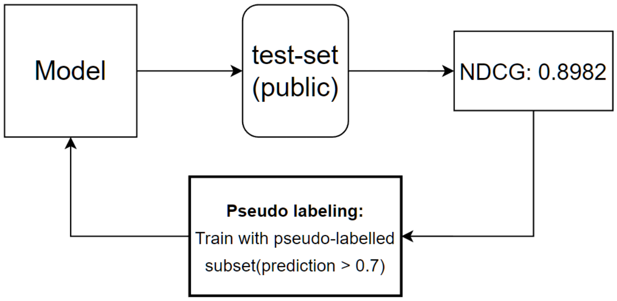

Figure 3: Train model with pseudo-labelled subset

3.3.4 Prompt Tuning[15]. In addition to increasing the robustness and recall of the model, we also make some adjustments to improve the precision. On the step of data processing, we prepared a series of specific tokens as implicit templates to provide extended reference features during encoding. By introducing this information, our model can have a better performance to handle diverse features.

3.3.5 Cross Attention[7]. Furthermore, it is obviously to found that the textual information between the query and the content of product are different in most cases (the text length of the product is much longer than the query). In order to prevent the query information vanishing after the neural transmission, our model automatically generates an attended attention over original self-attention, which makes the feature of latent distribution not only contain the query-to-product part, but also the query-to-mix_sequence.

3.3.6 Adversarial Training. Adversarial-training let us train networks with significantly improved resistance to adversarial attacks, thus improving robustness of models.

When the loss is below some threshold (like 0.6), we start using Adversarial Weight Perturbation (AWP) [18] in training steps that adversarially perturbs both model weights and the embeddings. In addition, the feature distribution of input data is attacked in each step. Besides, we also tried Fast Gradient Method (FGM) [9] which performs slightly worse than AWP does in public leaderboard.

3.3.7 English BERT Model. Although the task is multilingual, the English part is of large proportion, accounting for 54.5% of the total training sets. Take this into account, we also use DeBERTa-v3-large to train and predict for the English queries and products only besides the cross-lingual models. Combining with adversarial training, our single model gets improved from 0.899 to 0.9022 in the public leaderboard.

3.4 Ensemble

At last, model ensemble is used to get the final improvement. In detail, we use DeBERTa, RemBERT and XLM based models trained with different settings mentioned above as our base models for ensemble.

The weights for summing different model predictions are mainly determined by the public scores of the models and also the local cross-validation scores. We also lower the weights of the models with high correlation coefficients. Our score is improved from 0.9022 to 0.9057 on the public leaderboard, and from 0.9015 to 0.9043 on the private leaderboard after ensemble.

4 RESULTS & DISCUSSION

Some results of our experiments in Task 1 are shown in Table 2. The scores of our single models without any pre-processing or post-processing are around 0.8930 in the public leaderboard. DeBERTa, InfoXLM, XLM-RoBERTa, and RemBERT are used as the model backbone, then we concatenate the texts of query, title, description, bullet point together and truncate it with `max_length=128` after tokenizing as the model inputs.

After doing some data cleaning and hyper-parameters tuning, our single model is improved to 0.8960 on the public leaderboard. Batch size and learning rate are quite important in this task based on our experiments, we use `batch size=64`, `learning rate=3e-5` and `gradient accumulation=8` to train the model after tuning. Self distillation, pseudo labels and label smoothing help us further boost the model performance to 0.8990 on the public leaderboard.

With English pre-trained LMs and adversarial training, we can achieve 0.9022 in the public leaderboard and 0.9015 for the private, and this is our best single model. At last, we do model ensemble to get the final boost from 0.9022 to 0.9057 on the public leaderboard, and from 0.9015 to 0.9043 on the private leaderboard.

Table 2: Some results of our experiments in Task 1

Methods	NDCG (Public)	NDCG (Private)
mDeBERTa Baseline	0.8930	-
+ Data Clean + Parameter Tuning	0.8960	-
+ Self Distillation + Pseudo Labeling	0.8982	0.8975
+ Label Smoothing	0.8990	-
+ DeBERTa-v3-large + AWP/FGM	0.9022	0.9015
+ InfoXLM	0.9032	0.9032
+ XLM-RoBERTa	0.9041	0.9026
+ RemBERT + DeBERTa-v3-large + Translation augmentation + 2 of 7 folds bagging + Weighted multi-layer Pooling + Multi sample dropout	**0.9059***	0.9039
Model Ensemble Re-weighting	0.9057	**0.9043***

5 CONCLUSION

In this paper, we detailed our winning solution to the Query-Product Ranking task in Amazon ESCI Challenge of KDD Cup 2022. We use multilingual and English pre-trained LMs as backbone, with the combination of data processing, data augmentation, self-distillation, pseudo-labelling, label-smoothing and adversarial training, we improve the model step by step. For single model, we achieve NDCG score of 0.9022 on the public leaderboard and 0.9015 on the private leaderboard. At last, we do model ensemble to get the final boost from 0.9022 to 0.9057 on the public leaderboard, and from 0.9015 to 0.9043 on the private leaderboard, which ensures us to win the first place.

ACKNOWLEDGMENTS

Amazon and AIcrowd organizing team paid a lot of efforts during the whole process of the competition, we really appreciate it for hosting this fantastic competition. And we would like to thank everyone associated with organizing and sponsoring the KDD Cup 2022.

REFERENCES

[1] Amin Abolghasemi, Suzan Verberne, and Leif Azzopardi. 2022. Improving BERT-based query-by-document retrieval with multi-task optimization. In *European Conference on Information Retrieval*. Springer, 3–12.

[2] aicrowd. 2022. ESCI Challenge for Improving Product Search. https://www.aicrowd.com/challenges/esci-challenge-for-improving-product-search.

[3] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[4] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834* (2020).

[5] Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking embedding coupling in pre-trained language models. *arXiv preprint arXiv:2010.12821* (2020).

[6] 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).

[7] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423* (2016).

[8] 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).

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[10] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543* (2021).

[11] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654* (2020).

[12] Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788* (2019).

[13] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* (2019).

[14] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. 896.

[15] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[16] Yiqun Liu, Kaushik Rangadurai, Yunzhong He, Siddarth Malreddy, Xunlong Gui, Xiaoyi Liu, and Fedor Borisyuk. 2021. Que2Search: fast and accurate query and document understanding for search at Facebook. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3376–3384.

[17] 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 preprint arXiv:2206.06588* (2022).

[18] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems* 33 (2020), 2958–2969.

[19] Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345* (2020).

[20] Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for entity and relation extraction. *arXiv preprint arXiv:2010.12812* (2020).