ZhichunRoad at Amazon KDD Cup 2022: MultiTask Pre-Training for E-Commerce Product Search

Xuange Cui
cuixuange@jd.com
JD.com
Beijing, China

Wei Xiong
xiongwei9@jd.com
JD.com
Beijing, China

Songlin Wang
wangsonglin3@jd.com
JD.com
Beijing, China

Contents:

1.Introduction

i. DataSet:

[Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search]

ii. Data Analysis: (transformers.tokenizer)

a.Query-Text 99% data length <= 18

b.Product-Text

title, 99% data length <= 68

brand, 99% data length <= 9

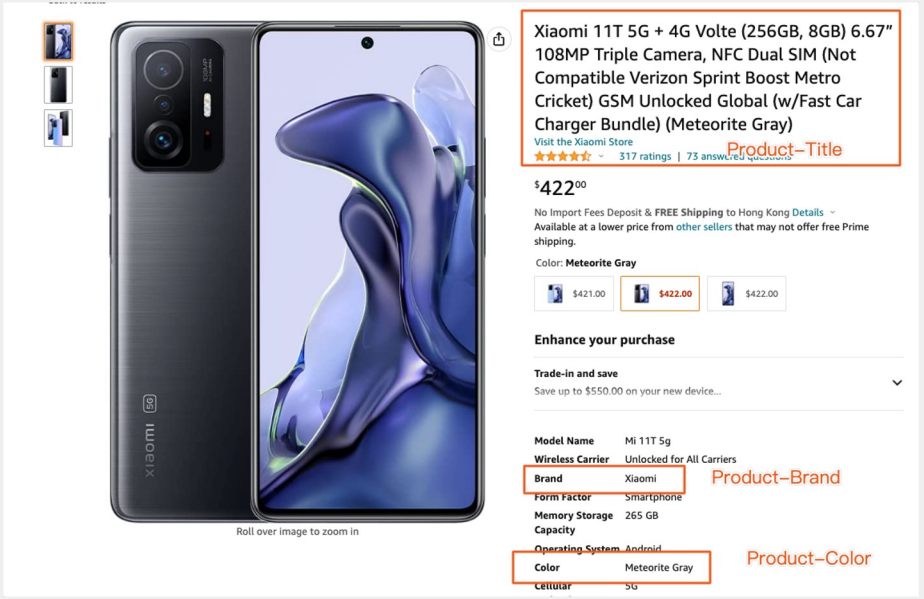
color, 99% data length <= 11

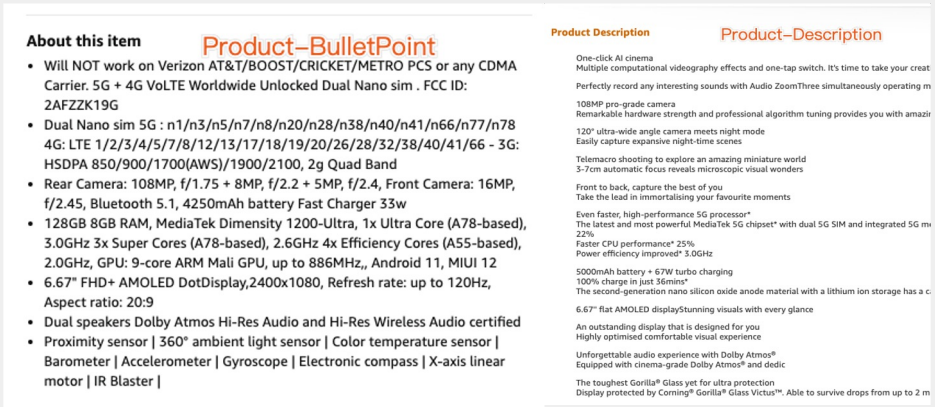
bullet-point, 99% data length <= 519

description, 99% data length <= 642

SubTask	Train Dataset	Test dataset	Languages
Task1	781K	48K	Spanish
Task2	1834K	277K	& English
Task3	1834K	277K	& Japanese

Table 1: The statistics of datasets.





1.Introduction

It seems to me that this competition mainly contains two challenges:

Q1: How to improve the search quality of those unseen queries？
Q2: There is very rich text information on the product side, how to fully characterize it ?

A1: We need more general encoded representations.
A2: As the bert-like model's "max_length paramter" increases, the training time increases more rapidly. We need an extra semantic unit to cover all the text infomation.

2.System Overview

Q1: How to improve the search quality of those unseen queries？

A1: We need more general encoded representations.

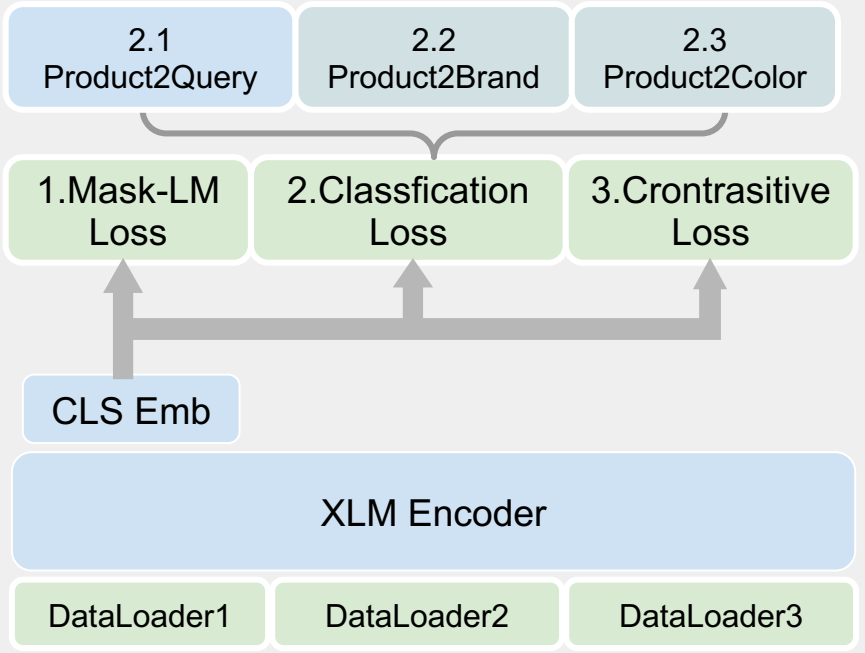
In pre-training stage, we adopt

i. mlm task,

ii. classification task,

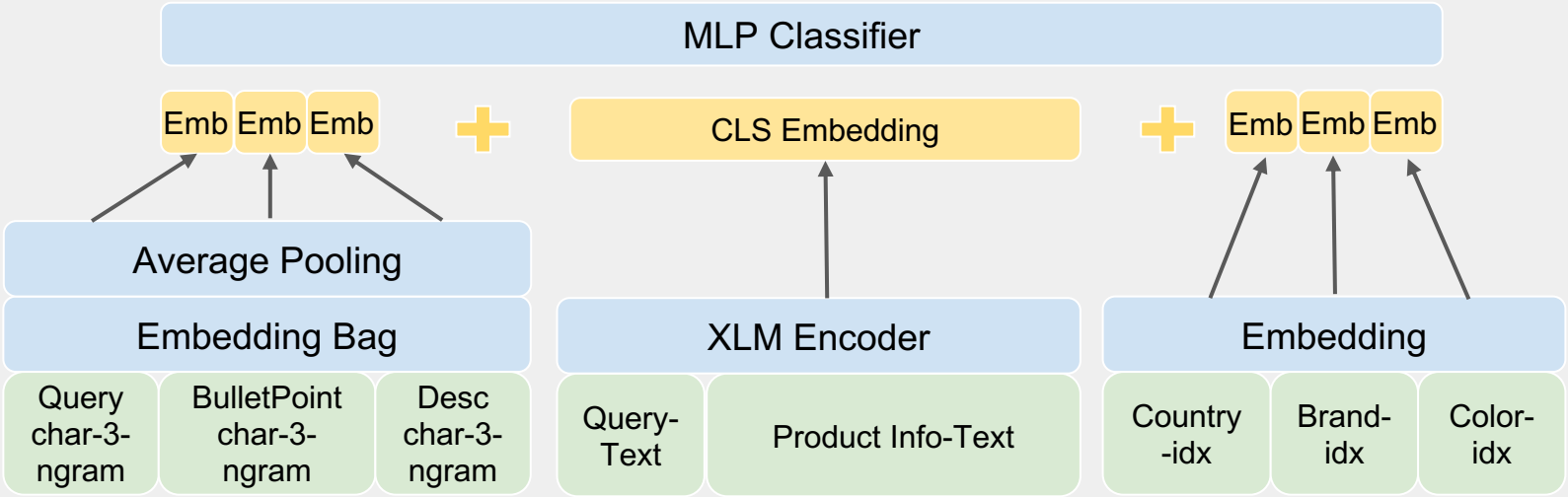
iii.contrastive learning task

to achieve considerably performance.



2.System Overview

Q2: There is very rich text information on the product side, how to fully characterize it ?
A2: We need an extra semantic unit to cover all the text infomation.



In fine-tuning stage, we concatenate the multi-granular semantic units, the [CLS] embedding from XLM encoder and the IDs' embeddings.

3.Experiments

3.1 Multi-task Pretraining

Algorithm 1: Training a MultiTask model.

Input: DataSet $\mathcal{D} = \{(x, y, z)_i\}_{i=1}^{|\mathcal{D}|}$

Initialize model parameters Θ randomly ;
Model trainer T that takes batches of training data as input to optimize the model parameters Θ ;
Set the max number of epoch: $epoch_{\max}$;
for *epoch in* $1, 2, ..., epoch_{\max}$ **do**
    Shuffle $\mathcal{D}$ by mixing data from different tasks ;
    for $\mathcal{B}$ *in* $\mathcal{D}$ **do**
        // $\mathcal{B}$ is a mini-batch of pre-training task ;
        Compute loss : $L(\Theta)$;
        1. $L(\Theta)$ = Mask LM Loss ;
        2. $L(\Theta)$ += Classification Loss ;
        3. $L(\Theta)$ += Contrastive Learning Loss ;
        Optimize the model using $L(\Theta)$;
    end
end

Output: Pre-trained Model Θ

Product2Query-Task:

Based on the Poisson distribution[1], a piece of text is intercepted from commodity text information as the faked query.

Product2Brand-Task and Product2Color-Task:

The multi-class classification that using product text information to predict the brand and the color of current item.

[1] The Poisson Distribution can be found in appendix.

Pre-Training Task	CV-MLM Loss	CV-Micro F1
Mask LM	1.966	74.97
+Product2Query	1.969	75.05
++Product2Brand	1.978	75.08
+++Contrastive Learning	2.047	75.08

Table 3: The effect of different pre-training tasks and keep accumulating from top to bottom. We report the cross validation MLM-Loss and Micro-F1 Score × 100 in the task2 setting.

3.Experiments

3.2 Fine-Tuning Methods

Methods	CV-Micro F1
+EMA	75.19
++FGM	75.30
+++R-Drop	75.43
++++Embedding Mixup	75.43

Table 5: The effect of different strategies and keep accumulating from top to bottom. We report the cross validation Micro-F1 Score × 100 in the task2 setting.

Confident Learning	CV-Metric
with-in-task1	NDCG, +0.005
with-in-task2	Micro-F1, -0.003
with-in-task3	Micro-F1, -0.002

Table 6: The effect of removing 4% noisy labels.

1.EMA, FGM, R-Drop

We used exponential moving average method (EMA), adversarial training (FGM) and regularized dropout strategy (R-Drop) to improve the model's generalization and robustness.

2.Confident Learning:

we consider using smaller datasets with removing ~4% noisy labels.

We used the smaller dataset to achieve an 0.005 improvement in task1, but we get worse results in tash2 and task3.

It could be explained that since task1 contains more difficult samples, the manually annotated data contains more label errors.

4.Conclusion

SubTask	Methods	Metric	Ranking
task1	ensemble 6 **large** models	ndcg=0.9025	5th
task2	**only** 1 **large** model	micro f1=0.8194	7th
task3	**only** 1 **large** model	micro f1=0.8686	8th

In this work:

We use data augmentation, multitask pretraining strategy and several fine-tuning methods to achieve considerably performance.

Moreover, we use a multi-granular semantic unit to discover the queries and products textual metadata for enhancing the representation of the model.

Future work includes:

1) Comparing with other pre-trained language models, such as deberta model.
2) Using other training strategies, such as self-distillation.