Retrieval-Enhanced Dual Encoder Training for Product Matching

Justin Chiu
Rakuten Institute of Technology, Rakuten Group Inc.
Boston, USA
justin.chiu@rakuten.com

Abstract

Product matching is the task of matching a seller-listed item to an appropriate product. It is a critical task for an e-commerce platform, and the approach needs to be efficient to run in a large-scale setting. A dual encoder approach has been a common practice for product matching recently, due to its high performance and computation efficiency. In this paper, we propose a two-stage training for the dual encoder model. Stage 1 trained a dual encoder to identify the more informative training data. Stage 2 then train on the more informative data to get a better dual encoder model. This technique is a learned approach for building training data. We evaluate the retrieval-enhanced training on two different datasets: a publicly available Large-Scale Product Matching dataset and a real-world e-commerce dataset containing 47 million products. Experiment results show that our approach improved by 2% F1 on the public dataset and 9% F1 on the real-world e-commerce dataset.

1 Introduction

Product matching is the task of finding the same product in a catalog for a specific query product. Dual encoders had been proposed as a state-of-the-art solution for information retrieval tasks(Karpukhin et al., 2020; Yamada et al., 2021; Luan et al., 2021), including product matching(Shah et al., 2018; Tracz et al., 2020). Training dual encoder requires positive and negative pairs as training data. The positive pairs are usually given as part of training data, and the negative pairs can be formed according to the positive pairs. Recent work (Zhan et al., 2021) tries to find negative pairs that are similar to positive pairs, which improved the quality of training data.

In this paper, we propose retrieval-enhanced dual encoder training. It is a two-stage training process to improve the dual encoder with better positive and negative pairs. For every pair, we define the first item in pairs as a query and the second item in pairs as a target. In stage 1 we train a dual encoder using the human-annotated positive pairs and form the negative pairs with in-batch negative. We then use the stage 1 dual encoder to retrieve queries in positive pairs on all possible targets, the retrieved results can form pairs to serve as training data for stage 2 training. Some of the positive pairs used in stage 1 might be excluded in the stage 2 training, while some extra positive pairs will be found for stage 2 training. We achieve better performance with the stage 2 dual encoder. We analyze how adding back the stage 1 positive pairs will impact the model performance, and find that the model performance dropped in several threshold settings even with stage 1 positive pairs included. This shows the stage 1 positive pairs left out in stage 2 training might not be helpful for the trained model, and our approach successfully identify those data and excluded them from stage 2 training.

In this paper, we make two major contributions. First, we introduce the two-stage dual encoder training for product matching. The stage 2 training data can be used to train a more robust dual encoder model. We demonstrate the effectiveness of this approach on a public dataset and a real-world e-commerce dataset.

Second, we analyze how adding stage 1 positive pairs back into our stage 2 training data will impact our system performance. Despite extra training data, the performance dropped in several threshold settings. This shows adding certain human-annotated training data sometimes reduces the performance. Our approach is a way to identify and exclude these training data automatically.

2 Product Matching and Dual Encoder

Given a product entry, the product matching system finds the same product in a product corpus. A product entry represents a set of information for a specific product, such as a title, description, image,

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 216–222

or category. Multiple entries of the same product could be sold by different vendors, and each vendor can give it a vendor-specific product title.

A dual encoder is a popular approach for information retrieval. It performs retrieval by encoding queries and targets into dense low-dimensional vectors and computes the distance between two vectors. The distance can be used as the search score. We can use the same encoder for both queries and targets if they are using the same information in the product entries. Training dual encoders requires pairs of query and target for contrastive learning. The training data contains positive and negative pairs. The positive pairs are created by a query and its related target. The in-batch negative is proposed to create the negative pairs from the existing positive pairs examples. Each query in a positive pair can form a negative pair with the other positive pair's target.

3 Retrieval Enhanced Dual Encoder Training

Retrieval-enhanced dual encoder training tries to find the ideal pairs for training. The in-batch negative might create trivial negative pairs that are not useful for training. The stage 1 retrieval step avoids these trivial negative pairs. This retrieval step also includes extra useful positive pairs that might not be present in the training data and exclude the positive pairs that might not be useful for training. The loss function we use is the same with the one reported in (Karpukhin et al., 2020).

3.1 Stage 1: Data pre-filtering and pair generation

When given human-annotated training data, we train a dual encoder using in-batch negative. The purpose of this stage 1 dual encoder is to create better training data for stage 2 training. We extract all the first items in the positive training pairs as queries and search those queries on the collection of possible targets using the stage 1 dual encoder. The possible targets include all the second items in the training pairs or the entire catalog. For each of the queries, we can take the top N retrieved results and their annotation to serve as the training data for the stage 2 dual encoder. Some stage 1 positive pairs could be low in ranking for the top N retrieved results, hence are excluded from stage 2 training. Some other positive pairs that are high in ranking for top N retreived results despite not used in stage 1 training will be included in stage 2 training. This is the main difference between our approach and the "hard-negative" training that had been reported in other papers, we also adjust the positive pairs. In the analysis section, we will analyze the difference in positive pairs between stage 1 and stage 2 training data, and how including stage 1 positive pairs in stage 2 training will impact system performance.

3.2 Stage 2: Training with pre-filtered data

With the stage 2 training data created by the stage 1 retrieval results, we can use the created data to train the stage 2 dual encoder. This avoids the trivial negative pairs that are common in the in-batch negative since every pair in this training data is close in the distance for the stage 1 dual encoder. The positive pairs that are far in distance also get excluded in the training data, and some positive pairs that are close in distance will be included.

4 Experiments

Our experiments are done on two different datasets, the WDC Product Data Corpus and Gold Standard for Large-Scale Product Matching (LSPM) (Primpeli et al., 2019)[1] and our in-house English catalog. The former dataset enables us to compare performance with other approaches reported by other papers, while the latter dataset can give us insight into how our approach can improve real-world deployed e-commerce systems.

4.1 Experiments Setup

4.1.1 WDC corpus

The system is given information such as the title or description for a pair of products, and the product matching system needs to predict whether a pair of products is the same product or not. The pairs we used for evaluation are the gold standard pairs provided by the dataset. The benefit of this setup is this enables us to do direct comparisons between the results reported on the WDC corpus website. This also avoids the need for large-scale comparison between a specific item and every other item in the corpus, which reduced the computation requirement significantly.

4.1.2 In-house English catalog

The system is given a product title as a query, and the product matching system needs to find the iden-

[1] http://webdatacommons.org/largescaleproductcorpus/v2/index.html

tical product in the English catalog corpus. This is how the system will be used when deployed for production. This does not require predetermined pairs to do prediction, hence the approach needs to be more efficient since it requires to do matching between the query and every item in the catalog.

4.2 Dataset

4.2.1 WDC corpus

The WDC Product Data Corpus was presented as a big product-matching benchmark dataset for evaluating different matching methods. The product data were provided with annotations from schema.org including some form of product ids like GTIN (Global Trade Item Number) or MPN (Manufacturer Part Number). It provides pre-assembled training and validation sets for comparison between different methods. For each product, information such as product title, description, and category are provided.

4.2.2 In-house English catalog

Our in-house English catalog contains 47 million products that contain GTIN information. We use GTIN as the identifier to decide whether two products are the same. Our training data is created by pairing up using 25.7 million queries from user activities and product titles for the product in our catalog. We use queries that have matching products in our catalog for our experiment. We only use product titles to avoid the mismatch where some sellers provide rich product descriptions while others provide limited or no descriptions. Each unique product can only form one entry in the training data, avoiding the training data to be biased toward popular products. If there are multiple products with the same GTIN, we randomly select one that matches the user queries and create pairs from it. We build 700,000 pairs as training data, and 30,000 pairs as development data. We select another 19,683 user queries as evaluation queries. The evaluation query has no overlap with all the products used in the training and development data.

4.3 Model

The dual encoder model is a common practice for large-scale search and matching(Shah et al., 2018; Tracz et al., 2020). We adopt BERT base uncased from Huggingface(Wolf et al., 2019)[2] for our product matching model. The hyperparameters we used for training are reported in Table 1.

Parameter	WDC	English catalog
Batch size	128	100
Max seq. length	64	64
Learning rate	1e-05	1e-05
Temperature	1.0	1.0
Vocabulary size	30,522	30,522
Max epoch	20	20

Table 1: Hyperparameters for each model.

4.4 Training

4.4.1 WDC Corpus

We take the extra large train and valid set for all categories from the WDC corpus to use as our training and development data. Only positive pairs in the training data are used. We trim down the training set from 24194 pairs to 24192 pairs and trim down the development set from 6079 pairs to 6048 pairs. The reason for this trim down is that the training data need to be multiples of batch size (128) in order to do proper in-batch negative training. We use this setup to train our stage 1 dual encoder. The training for stage 1 dual encoder takes about 35 minutes on 4 Quadro P6000 GPUs.

After obtaining the first dual encoder, we then split the 24192 training pairs into queries and targets. The training data for the dual encoder contains product title pairs, hence we can collect every first item in pairs to use as a query, and every second item in pairs as a target. This gives us a set of 9518 unique query items and 9520 unique target items.

We then use the stage 1 dual encoder to retrieve the most relevant products from targets for each query. We collect the top 32 retrieved results for every query to form stage 2 training pairs. The same process is also done on the development data. This gives us 304576 pairs as the stage 2 training data, and 141664 pairs as stage 2 development data. We then train the stage 2 dual-encoder with these data. The training for stage 2 dual encoder takes about 7.5 hours on 4 Quadro P6000 GPUs.

4.4.2 In-house English catalog

We use 700,000 training pairs and 30,000 development pairs for the stage 1 dual encoder training. The training for the first dual encoder takes about 3.5 hours on 8 A100 GPUs.

After training the first encoder, we use the first

[2] https://huggingface.co/bert-base-uncased

encoder as the search engine to search the 700,000 queries from training data on the English catalog that contains 47 million products. The same search is also done on development data. We take the top 5 retrieved results, and this gives us 3,500,000 pairs of training data and 150,000 pairs of development data. The reason for not using the top 32 retrieval results is the computation time will be too much for the stage 2 training. From this new set, we train the stage 2 dual-encoder. The training for the stage 2 dual encoder takes about 17.5 hours on 8 A100 GPUs.

4.5 Inference

After training the dual encoder, we encode the targets in the WDC gold standard or the in-house English catalog with the trained model and then index them using FAISS (Johnson et al., 2019) offline. FAISS is an open-source library for efficient similarity search. We then encode the queries with the same encoder and retrieve the top k product titles from the FAISS index.

For the WDC corpus task, we predict a match if the target of evaluation pairs is in the top 10 retrieved results for the query. We use the development data to decide the parameter 10 for our experiment. This helps us use ranked search results for binary classification, and makes it comparable with other baselines.

For the in-house English catalog task, we search the evaluation query on our English catalog. We then check whether the top 1 returned product from the catalog is the same item or not. We can set a threshold on the distance reported by the dual encoder and consider the retrieved result with a distance shorter than the threshold not being a match, which can also function as a precision-recall tradeoff. Our development data shows that setting no threshold will achieve the highest F1 score for our task.

4.6 Baselines

For experiments on the WDC corpus, we include the results of TFIDF-cosine and Deepmatcher systems reported on their website as the baseline. The TFIDF-cosine system in the best result they reported by only using the product title. The Deepmatcher system uses the product title and description and is the best system reported. We also include a baseline of using the BERT (Devlin et al., 2019) model to encode two product titles jointly for binary classification. We use BERT as an encoder, and put a classifier layer on top of BERT. We convert product title pairs in the form of `[CLS] Title 1 [SEP] Title 2`. We regard the embeddings of `[CLS]` token obtained from BERT as a representation of the title pairs and feed it to the classifier layer to judge if both titles refer to the same product. The performance for our stage 1 dual encoder is also reported as a baseline.

For experiments on the in-house English catalog, the baseline is the stage 1 dual encoder and will compare with the stage 2 dual encoder. Using the BERT model like the WDC experiments is hard in this real-world setup, as it will require encoding 925,000,000,000 pairs to perform retrieval on the entire English catalog for all evaluation queries.

Setup	P	R	F1
TFIDF-cosine	46.00	74.00	57.00
Deepmatcher	92.04	88.36	90.16
BERT	93.54	89.33	91.39
Stage 1	52.24	95.08	67.43
Stage 2	**92.28**	**93.75**	**93.01**

Table 2: Results for WDC corpus

4.7 Evaluation

4.7.1 WDC corpus

The evaluation is based on the provided gold standard for all category sets. We benchmark the precision, recall, and F1 for our system's prediction, comparing with the labels for the pairs in the gold standard set. This enables us to compare with reported results.

4.7.2 In-house English catalog

We collected a set of 19,683 user queries as evaluation queries, and these query products are not in the training and development set. We evaluate whether the retrieved top 1 results are the correct match. We also report precision, recall, and F1 score.

4.8 Results

4.8.1 WDC corpus

Table 2 shows our results on the WDC dataset. We include two baselines from the WDC website. Our BERT baseline can be considered as a state-of-the-art product matching, with high accuracy yet relying on heavy computation.

Our results show that the stage 1 dual encoder is far from the quality of the BERT baseline. The stage 2 dual encoder has better performances and

Setup	P	R	F1
Stage 1	38.57	38.57	38.57
Stage 2	**49.15**	**49.15**	**49.15**

Table 3: Results for English catalog

relies on much less computation during inference compared with the BERT baseline.

4.8.2 In-house English catalog

Table 3 shows our results on the in-house English catalog. Since every query used in the evaluation existed in the catalog, we will have the same precision and recall if we set no distance threshold. We will discuss more regarding the distance threshold in the analysis section.

5 Analysis

5.1 Comparing Stage 1 and 2 training and development data

Retrieval-enhanced training impacts both positive and negative training and development data. This is the main difference between our approach and the "hard-negative" presented in another paper (Zhan et al., 2021). Our retrieval step might exclude some positive pairs used in stage 1 training while including other positive pairs that are not presented in the stage 1 training data. We will provide an analysis of the English catalog experiments since it has richer data and is set up for a real-world production system.

Our stage 2 training data has 579,326 positive pairs. This includes 314,190 pairs that showed up in stage 1 training data and 265,136 positive pairs that are added through our retrieval step but are not presented in stage 1 training data. This shows that more than half of stage 1 positive pairs are excluded from stage 2 training data, and the retrieval step creates around 46% of the positive pairs in stage 2 training data. A similar trend can be observed in the development data. It is common to achieve better performance with more data. However, our approach removes positive pairs yet achieves better performance.

5.2 How adding back the excluded stage 1 positive pair impacts the performance

Our approach excludes more than half of the positive pairs in stage 1 training data. We can manually add back the excluded positive pairs used in stage 1 on both training and development data to see how that impacts the performance. We also list out the possible threshold setting. The 20% threshold means that among all the search results, the top 20% with the shortest distance is considered a match, and the others are considered not a match. As we relax the threshold to a higher percentage, we will gain more recall and lose precision.

Table 4 shows the results of adding back the excluded positive pairs used in stage 1 to stage 2. The combined set has more than 10% new data compared with the stage 2 set. However, we see the combined set is performing worse on multiple thresholds. This could be caused by adding positive pairs that are very different in text, which disturbs the quality of embedding space. Our retrieval step identify these less useful positive pairs automatically and excluded them from stage 2 training data. We believe finding these less useful data and excluding them from the training process is a direction worth exploring.

6 Related Work

Earlier works (Mauge et al., 2012; Ghani et al., 2006) tried to product matching based on certain extracted attributes from product entries, but recently (Shah et al., 2018; Tracz et al., 2020) it started to move towards using the text in the product entries directly, which avoids the possible errors caused by the attribute extraction process.

There are two ways of using the text in product entries to solve product matching that had been studied. One (Shah et al., 2018) is to treat the task as an extreme classification problem, and another (Tracz et al., 2020) is to treat the task as an information retrieval problem. To treat product matching as extreme classification, the paper built a multi-class classifier that considers each product as a separate class. In real world scenario, this could easily mean over million classes, and there are two main challenges to the approach. First, how to maintain the performance for a classifier with millions of classes. Second, the classifier will need to be retrained when a new product enters the catalog.

Another way for product matching is to treat the task as an information retrieval task. The work in this direction started from using a standard retrieval engine to more deep learning-based approaches. Among deep learning approaches, utilizing a dual encoder as a retrieval system proved to be efficient compared with more complex joint encoding ap-

Threshold	Stage 2 data P	R	F1	Stage 2 + 1 data P	R	F1
10%	68.05	6.80	12.38	68.87	6.89	**12.53**
20%	68.55	13.71	**22.85**	63.80	12.76	21.27
30%	67.93	20.38	**31.35**	61.97	18.59	28.61
40%	66.20	26.48	**37.83**	60.65	24.26	34.66
50%	63.96	31.98	**42.64**	59.38	29.69	39.59
60%	61.60	36.96	**46.20**	58.56	35.13	43.92
70%	58.57	40.99	**48.23**	57.15	40.00	47.06
80%	55.89	44.71	**49.68**	55.58	44.46	49.40
90%	52.77	47.49	49.99	53.23	47.90	**50.42**
None	49.16	49.16	49.16	50.03	50.03	**50.03**

Table 4: Comparison on different thresholds for adding excluded stage 1 positive pairs

proaches. The in-batch negative training was first proposed for the dual encoder training. Later the "hard negative" training was also proposed (Zhan et al., 2021) to address the quality issue of the negative pairs for training. Our work is inspired by the idea of hard-negative training, yet we further explore the idea of selecting the training data. We should not only improve the quality of negative data, but also the positive data.

7 Conclusion

We demonstrated retrieval-enhanced dual encoder training for product matching. This approach can utilize the available training data in an efficient way to achieve improvement even with no extra annotated training data available. Our stage 2 training use the same annotated training data as stage 1, the difference is on what pairs do we select for training.

Our empirical results on two different datasets show that our approach can achieve improvement comparing the standard in-batch negative dual encoder training. Our analysis further shows that the approach not only provided valuable negative pairs for training but also adjusted positive pairs used in training data to achieve better results.

As a result of this proposed training, we obtained a new way to train dual encoders for product matching. We can identify better training data automatically, instead of relying on the training data given by any specific dataset.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Karin Mauge, Khash Rohanimanesh, and Jean-David Ruvini. 2012. Structuring E-commerce inventory. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 805–814, Jeju Island, Korea. Association for Computational Linguistics.

Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The wdc training dataset and gold standard for large-scale product matching. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 381–386.

Kashif Shah, Selcuk Kopru, and Jean-David Ruvini. 2018. Neural network based extreme classification

and similarity models for product matching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 8–15, New Orleans - Louisiana. Association for Computational Linguistics.

Janusz Tracz, Piotr Iwo Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. 2020. BERT-based similarity learning for product matching. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, pages 66–75, Barcelona, Spain. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.