A Pre-trained Matching Model Based on Self- and Inter-ensemble For Product Matching Task

Shiyao Xu[1,2], Shijia E[2], Li Yang[1,2], and Yang Xiang[1]

[1] Tongji University, Shanghai, China
[2] Tencent, Shanghai, China
{xushiyao, li.yang}@tongji.edu.cn, tjdxxiangyang@gmail.com
e.shijia@gmail.com

Abstract. The product matching task aims to identify that if a pair of product deriving from different websites refer to the same product or not. While the accumulated semantic annotations of products make it possible to study deep neural network-based matching methods, product matching is still a challenging task due to suffering from the class imbalance and heterogeneity of textual descriptions. In this paper, we directly regard product matching as a semantic text matching problem and propose a pre-trained matching model based on both self- and inter-ensemble. BERT is the main module in our approach for binary classification of product pairs. We perform two types of ensemble methods: self-ensemble using stochastic weight averaging (SWA) for the same model, and inter-ensemble combing the prediction of different models. Additionally, the focal loss is adopted to alleviate the imbalance problem of positive and negative samples. Experimental results show that our model outperforms existing deep learning matching approaches. The proposed model achieves an F1-score of 85.94% on the test data which ranks second in the SWC2020 on Mining the Web of HTML-embedded Product Data Task One. Our implementation has been released [3].

Keywords: Product Matching · BERT · Focal Loss · SWA.

1 Introduction

In recent years, online shops (e-shops) are increasingly adopting semantic markup languages to describe their products to improve their visibility. Those semantic annotations are conducive to the further analysis of product offers. However, different annotation systems used by e-shops often lead to data inconsistencies or conflicts. Product matching is the task of identifying the similarity of product offer pairs, which is the fundamental technology to construct a unified system such as product knowledge graphs. Moreover, product matching can improve the efficiency and experience of online purchasing. As for customers, matching the

[3] https://github.com/englishbook/product-matching

same product on different websites is convenient for them to compare and find the best choice quickly. As to the e-commerce platform, the matching results can be used for product recommendation. Therefore, product matching is a crucial problem in the e-commerce domain.

Product offers are described by the textual information (e.g. title, description, and brand). We can think of product matching as a semantic text matching problem. Although many previous works have been done on text matching and have shown great success with deep neural networks [15, 12, 6], the task remains a great challenge in the e-commerce environment. On the one hand, the semantics of the natural language are diverse and complex, especially on the Internet. E-shops like to use many exaggerated words and new words to attract customers. On the other hand, there is a problem of class imbalance. Despite the huge quantity of product offers, most of the pairs of products are not matched which makes the number of negative samples much larger than the positive ones.

To address those limitations, we propose a pre-trained matching model based on both self- and inter-ensemble for product matching in this paper. 1) For semantic complexity, we apply pre-trained BERT to model text pairs. BERT [3] is pre-trained on a large-scale unlabeled dataset and then fine-tuned for downstream product matching task. Compared to traditional DNN models, BERT has learned richer semantic information. 2) For class imbalance, we adopt the focal loss [8] to better optimize parameters. It makes the training process can focus on a few uncertain samples. 3) For generalization, we combine both self- and inter-ensemble methods. Self-ensemble integrates model weights of the same model at different training epochs. Inter-ensemble averages the matching score resulting from different models. Our ensemble model achieves an F1-score of 85.94% in the final evaluation of the product matching task on the SWC2020 challenge.

2 Related Work

With the development of deep learning, large amounts of matching models based on deep neural networks have emerged and shown their effectiveness. Previous works mainly focus on the siamese architecture [1]. Those approaches generally take word embeddings of text pairs pre-trained by word2vec [10] as input, convert word embeddings to text representations, and then compute the similarity between two text representations. Convolutional neural network (CNN) [4] and recurrent neural network (RNN) [11] are the two mainstream methods used for text modeling. ESIM [2] and EACNNs [13] incorporate attention mechanisms into models to pay more attention to the relevant parts of text pairs. However, they all depend on the quality of the training dataset and face the difficulty of polysemy.

Recently, pre-trained language models are proposed and have achieved significant improvement in various NLP tasks, with state-of-the-art models such as BERT [3], RoBERTa [9], and XLNET [14]. The main idea of them is the pre-training language model on large-scale unlabeled corpus before fine-tuning on

downstream tasks. Therefore, pre-trained models contain rich semantic information and generate contextualized embeddings instead of fixed ones. In this paper, we adopt pre-trained BERT as the base model to solve the product matching task where the semantics of textual descriptions is relatively complex.

3 Model Description

3.1 Data

The WDC product data corpus, the largest publicly available product data corpus, is released by the Web Data Commons project in 2018. The corpus consists of 26 million products originating from 79 thousand websites. Products in the corpus are described by these properties: id, cluster id, category, title, description, brand, price, and specification table. Products with the same cluster id indicate the same product. In the SWC2020 challenge product matching task [4], organizers provide a dataset containing matching and non-matching pairs of products only from *Computers* & *Accessories* category. It is sampled from the product data corpus according to the clusters (68K product pairs for training, 1.1K for validation, and 1500 for testing). We also utilize an extended training dataset [5] with all the four product categories derived by the same sample strategy (214K for training, and 4.4K for validation). Matching models can learn more information from extra categories and thus make a more accurate prediction on samples from *Computers*. The statistics of these training datasets are listed in Table 1.

Table 1. The statistics of product matching training sets.

Category	**#clusters**	**#positive**	**#negative**	**#samples**
Computers	745	9690	58771	68461
All	2481	30198	184463	214661

Besides, some data preprocessing operations are performed before inputting product information into models. We remove stopwords (using NLTK) and lowercase all textual descriptions.

3.2 Matching Model

Pre-trained language models have learned from the large-scale corpus, so they possess certain advantages to transfer to the e-commerce domain. In this section, we build a product matching model based on pre-trained BERT. The overall framework is shown in Figure 1.

[4] https://ir-ischool-uos.github.io/mwpd/index.html
[5] http://webdatacommons.org/largescaleproductcorpus/v2/index.html

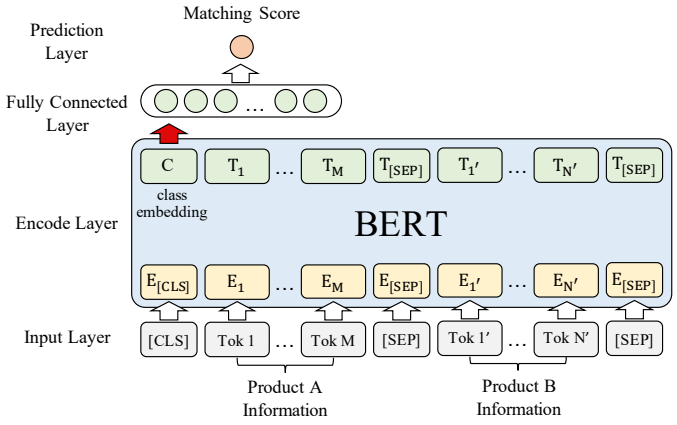

Fig. 1. The overall framework of BERT matching model.

Although products are described by many attributes, most of the fields contain NULL values. The title attribute of all products is filled, and the filling rate of the description attribute is relatively high. Therefore, we mainly focus on these two attributes. We concatenate the textual information of product pair by [SEP] token at first, and then add [CLS] and [SEP] tokens at the beginning and end respectively as the input of BERT, $x = [[CLS]\ t_A\ d_A\ [SEP]\ t_B\ d_B\ [SEP]]$. BERT can model the input tokens through the multi-layer bidirectional Transformer encoder and generate high-level representations. The output state of BERT that corresponds to [CLS] token is used as the pair representation. We feed the representation into a fully connected layer with the sigmoid activation function and obtain the final matching score p between two product offers.

3.3 Focal Loss

In general, we often adopt binary cross-entropy loss directly for binary classification tasks. However, as shown in Table 1, the number of positive and negative samples in the dataset is imbalance making easily classified negatives comprise the majority of the loss and dominate the gradient. To solve the problem, Lin et al. [8] scale the cross-entropy loss and propose the focal loss which is formulated as follows:

$$FL = -\alpha_t(1-p_t)^{\gamma}log(p_t) \tag{1}$$

where $p_t = p$ when the ground truth is 1, otherwise $p_t = 1 - p$. The weight factor $\alpha \in [0, 1]$ is set according to class frequency to balance the importance of positive and negative samples. For convenience, we define α_t similar to p_t. Moreover, the focusing parameter $\gamma \in [0, 5]$ is introduced to differentiate easy and hard examples. In that way, the samples that have been accurately classified contribute less to the loss so that model can focus training on few hard cases.

3.4 Ensemble

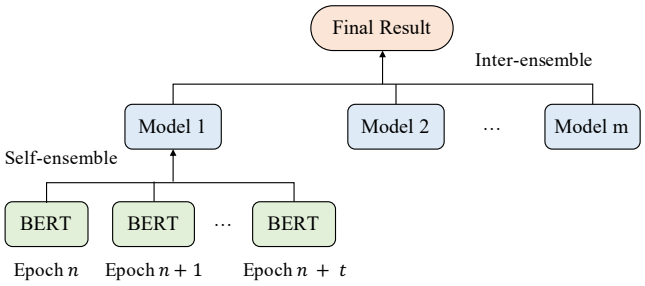

Fig. 2. The combination of self- and inter-ensemble strategies.

Self-ensemble: for the same model at different training. As illustrated in Figure 2, we use the stochastic weight averaging strategy [5] in the training phase. When the model is about to converge, the model weights trained at different epochs are averaged as the final weights of the model. Compared with the traditional method that only retains the weights at the final epoch, SWA can help avoid the local optimal solution and improve the generalization ability without increasing training cost.

Inter-ensemble: for different models. Different models have their advantages. For example, models trained by the cross-entropy loss should predict more accurately on substantial non-matched samples, while models with the focal loss should perform better on few hard examples. In this paper, multiple models are obtained by training on different datasets, inputs, and loss functions. As illustrated in Figure 2, we average the prediction probability of these models as the final results to combine the strengths of different models.

3.5 Post-processing

Many attributes are not fed into the model for training, but they are undoubtedly useful for product matching. In the SWC2020 challenge, we attempt to take full advantage of them to correct the prediction results. The values of category attributes are assigned to four unified categories. Two products belonging to different categories must be non-matched. For test pairs with prediction results of 1 but different categories, we correct their results to 0. The following experiments demonstrate the effectiveness of the post-processing operation. Similarly, the brand and price can also be used for correction in the future.

4 Experiments

4.1 Experimental Setups

Product textual information is padded or truncated to a fixed length. The max length is set to 64 and 200 for the input only with product title and the concatenation of title and description, respectively. For pre-trained BERT, we initialize the model by the weights of $\mathbf{BERT}_{base}$. For focal loss, α is set to 0.75, and γ is set to 2 to focus on hard positive samples. The optimizer we adopt is Adam [7] with constant learning rate of 2×10^{-5}. We start to use the SWA and early stopping strategy after the fifth training epoch.

4.2 Results on the Validation Set

The F1 score on the positive class is used as the evaluation metric. We have trained multiple models with different architectures and training strategies. Models trained on the All dataset are also evaluated on the validation set with four categories. Table 2 shows the experimental results of these models on the validation set in detail. We can find that the performance of pre-trained BERT for the product matching task is significantly better than other classic matching models (e.g. CNN, ESIM). Incorporating focal loss and SWA strategy further improves the BERT models. Moreover, post-processing can indeed correct some prediction errors effectively.

Table 2. The results on validation set. CE, FL represents cross-entropy and focal loss, respectively. Post_F1 means the F1 score after post-processing.

	Model	Dataset	Input	Loss	SWA	F1	Post_F1
1	Siamese CNN	All	title	CE	×	0.8445	0.8479
2	ESIM	All	title	CE	×	0.9167	0.9174
3	BERT	All	title	CE	×	0.9410	0.9426
4	BERT	All	title	CE	✓	0.9427	0.9431
5	**BERT**	**All**	**title**	**FL**	✓	**0.9481**	**0.9496**
6	BERT	All	tit+desc	CE	✓	0.9369	0.9381
7	BERT	All	tit+desc	FL	✓	0.9384	0.9411
8	BERT	Computers	title	CE	×	0.9630	0.9646
9	BERT	Computers	title	CE	✓	0.9646	0.9662
10	BERT	Computers	title	FL	✓	0.9585	0.9633
11	**BERT**	**Computers**	**tit+desc**	**CE**	✓	**0.9700**	**0.9700**
12	BERT	Computers	tit+desc	FL	✓	0.9672	0.9688

4.3 Final Results

After obtaining various models, we select several models that perform better on the validation set and try to integrate them by inter-ensemble strategy. Averaging the matching score predicted by multiple models can combine their strengths.

The results of our ensemble models on the validation set in the Computers dataset are presented in Table 3. Ensemble models are generally better than single models. In the final evaluation of the test data, we submitted the prediction result of our best ensemble model. As shown in Table 4, we achieve an F1-score of 85.94% on the test set which ranks second. The experimental results demonstrate the effectiveness and generalization ability of our proposed model.

Table 3. The results of our ensemble models on validation set. The model number is given in Table 2.

Ensemble Model	F1
5+7+9	0.9718
5+7+10	0.9689
5+7+11	**0.9754**
5+7+12	0.9737
5+7+11+12	0.9735

Table 4. The final results on test set.

Team	Precision	Recall	F1
Baseline	0.7089	0.7467	0.7273
Megagon	0.8268	0.6552	0.7311
ISCAS-ICIP	0.8389	0.8133	0.8259
ASVinSpace	0.8620	0.8210	0.8410
PMap	0.8204	0.9048	0.8605
Ours (5+7+11)	0.8286	0.8838	0.8553
Ours (5+7+12)	**0.8063**	**0.9200**	**0.8594**

5 Conclusion and Future Work

In this paper, we propose a pre-trained matching model based on both self- and inter-ensemble for product matching. Pre-trained BERT is adopted as the base matching model. We incorporate the SWA strategy in the training phase to improve the generalization ability of models and combine the output of different models to make full use of their advantages. Experimental results show that our model achieves great improvement compared with existing state-of-the-art matching models.

An interesting direction of future work is to pre-train BERT on product data corpus so that it can learn more product description ways. Also, post-processing operations are worthy of further study, especially in industry practice.

Acknowledgments This work was supported by the National Key Research and Development Program of China (Grant No. 2019YFB1704402) and 2019 Tencent Marketing Solution Rhino-Bird Focused Research Program.

References

1. Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Sckinger, E., Shah, R.: Signature Verification using a "Siamese" Time Delay Neural Network. In: Proceedings of the International Journal of Pattern Recognition and Artificial Intelligence. vol. 7, pp. 669–688 (1993)
2. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpent, D.: Enhanced LSTM for Natural Language Inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. pp. 1657–1668. Association for Computational Linguistics (July 2017)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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. pp. 4171–4186. Association for Computational Linguistics (June 2019)
4. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional Neural Network Architectures for Matching Natural Language Sentences. In: Proceedings of the 27th International Conference on Neural Information Processing Systems. vol. 2, pp. 2042–2050. Curran Associates, Inc. (December 2014)
5. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging Weights Leads to Wider Optima and Better Generalization. Computing Research Repository **arXiv:1803.05407** (2018)
6. Kim, S., Kang, I., Kwak, N.: Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence. pp. 6586–6593. AAAI press (January 2019)
7. Kingma, D.P., Ba, J.L.: Adam: A Method for Stochastic Optimization. In: Proceedings of the 3rd International Conference for Learning Representations (May 2015)
8. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (December 2017)
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. Computing Research Repository **arXiv:1907.11692** (2019)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
11. Mueller, J., Thyagarajan, A.: Siamese Recurrent Architectures for Learning Sentence Similarity. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 2786–2792. AAAI Press (Feburary 2016)
12. Wang, Z., Hamza, W., Florian, R.: Bilateral Multi-Perspective Matching for Natural Language Sentences. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 4144–4150 (August 2017)
13. Xu, S., E, S., Xiang, Y.: Enhanced attentive convolutional neural networks for sentence pair modeling. Expert Systems with Applications **151**, 113384 (2020)

14. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 5754–5764 (December 2019)
15. Yin, W., Schütze, H., Xiang, B., Zhou, B.: ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. Transactions of the Association for Computational Linguistics **4**, 259–272 (2016)