# Multimodal Representation Learning-Based Product Matching

Changkai Feng, Wei Chen, Chao Chen, Tong Xu(✉), and Enhong Chen

School of Data Science, University of Science and Technology of China, Hefei, China
{changkaifeng,chenweicw,chenchao11}@mail.ustc.edu.cn,
{tongxu,cheneh}@ustc.edu.cn

**Abstract.** This paper describes our methodology for the identical product mining task organized by the China Conference on Knowledge Graph and Semantic Computing (CCKS) 2022. This identical product mining task has two main challenges: 1) How to perform text representation to refine product representation. 2) How to more effectively combine text representation and image representation. For the first challenge, we propose the K-Gram Exponential Decay scheme in the text representation module to aggregate the information of surrounding words. For the second challenge, we apply conventional multimodal representation learning to combine text representation and image representation to generate the item representation. We view the identical product mining task as a binary classification task for product pairs, for which we adopt sample pair-based contrastive learning. Extensive experiments have demonstrated the effectiveness of our method. We won first place in the competition by utilizing model ensemble and post-processing.

**Keywords:** Multimodal representation learning · Product matching · Contrastive learning

## 1 Introduction

Knowledge Graphs are a significant component of enterprise data infrastructure and a core element of upper-layer applications [1]. In January 2022, Alibaba released AliOpenKG, the first Open Knowledge Graph for digital commerce. The process of creating e-commerce product relationships is a crucial step in the creation of the Knowledge Graph for digital commerce. However, the personalization of merchants' published product information has caused inadequate standardization and structuring of product, and different categories of product have various unique and important attributes, making it challenging to align fine-grained similar product.

Existing techniques [2] for fine-grained product alignment are mostly based on representation learning due to the large number of product on e-commerce platforms. Specifically, the item representation is obtained by characterizing the item with unstructured and structured information of the item. The same item is then obtained via vector retrieval. However, this identical product mining task

N. Zhang et al. (Eds.): CCKS 2022, CCIS 1711, pp. 180–190, 2022.
https://doi.org/10.1007/978-981-19-8300-9_20

views product alignment as a binary classification task based on product pairs, and employing vector-based retrieval is too complicated. For identical product mining, we employ sample pair-based contrastive learning. We first create separate textual and visual representations of the product using representation learning, then concatenate the two to create the final product representation. Finally, we utilize CoSENT[1] to gradually refine the product representation. In representation learning of text, since the traditional text representation method cannot highlight the local continuous token information, we propose the K-Gram exponential decay scheme, inspired by N-gram, for capturing and aggregating the surrounding continuous token information, which in turn refines the text representation. Additionally, inspired by Circle Loss [3] and Curricular Loss [4], we improved CoSENT further to create Circle-CoSENT and Curricular-CoSENT to promote contrastive learning between sample pairs.

In summary, this paper makes the following contributions:

- We propose the K-Gram Exponential Decay scheme refine text representation.
- We apply CoSENT for contrastive learning of sample pairs and further improve it to create Circle-CoSENT and Curricular-CoSENT.
- We adopt model ensemble for multimodal representation learning. It contains two sub-models for image representation and two for text representation.

## 2 Related Works

### 2.1 Product Matching

Product matching is generally based on representation learning. Tracz et al. [5] proposed category hard batch construction strategy and applied Triple Loss for product matching. Li et al. [6] utilized product titles and attributes to match product across platforms. Li et al. [7] proposed the Path-based Deep Network, which combines diversity and personalization to enhance matching performance. Peeters et al. [8] proposed the application of supervised contrastive learning for product matching.

### 2.2 Multimodal Representation Learning

Existing methods for multimodal information fusion generally use simple operations (e.g., concatenation, weighted summation) or attention-based methods. We utilize concatenation for fusion. Bi et al. [9] proposed characterizing three different types of news textual information (e.g., title, topic category, and entities) separately and obtaining news embedding by attention mechanism. Yu et al. [10] applied Cross-Modal Attention Mechanism to obtain textual representation of fused images and image representation of fused text and connect them for multimodal interaction.

[1] https://kexue.fm/archives/8847.

# 3 Methodology

## 3.1 Text Representation Module

In our text module, we choose RoBERTa [11] as encoder and feed the embedding of final layer into the following two sub-modules.

**Conventional Method.** First, we obtain the last layer of hidden layer features from the output of RoBERTa and perform the average pooling operation. Then, we use the dropout strategy to enhance the robustness. Finally, the final text representation is obtained by a layer of MLP.

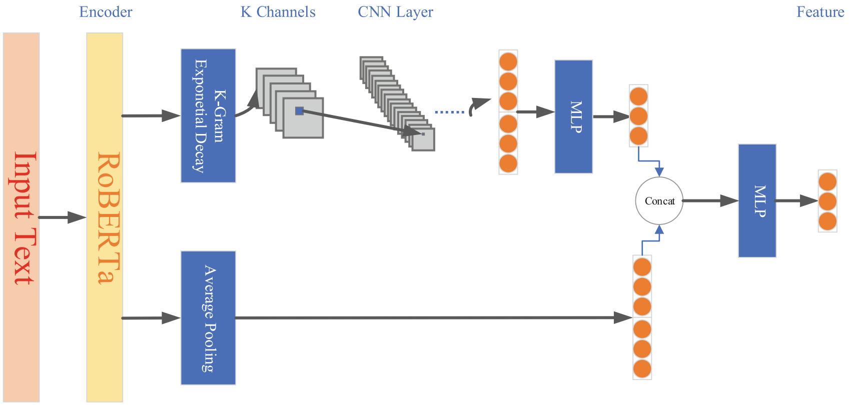

**Fig. 1.** Text representation module with K-Gram Exponential Decay and CNN

**K-Gram Exponential Decay.** We consider that text representation can be categorized into token representation, word representation, phrase representation, etc. The conventional text representation method can only highlight the global token information gained by the attention mechanism, not the local continuous token information that is crucial for forming word representation and phrase representation. Considering that the associated words generally occur consecutively, we are inspired by N-Gram and propose K-Gram Exponential Decay, a sliding window-like mechanism to capture phrase expressions of $K$ consecutive tokens. We employ the K-Gram Exponential Decay scheme to process the token embedding output by RoBERTa to obtain multi-channel embedding, which is then enriched with the hidden information extracted by CNN [12]. Finally, the obtained embedding are concatenated with the embedding from the average pooling module and passed through an MLP layer to obtain the final text representation of the product. The general framework of the module is shown in Fig. 1. The K-Gram Exponential Decay scheme is described in the following.

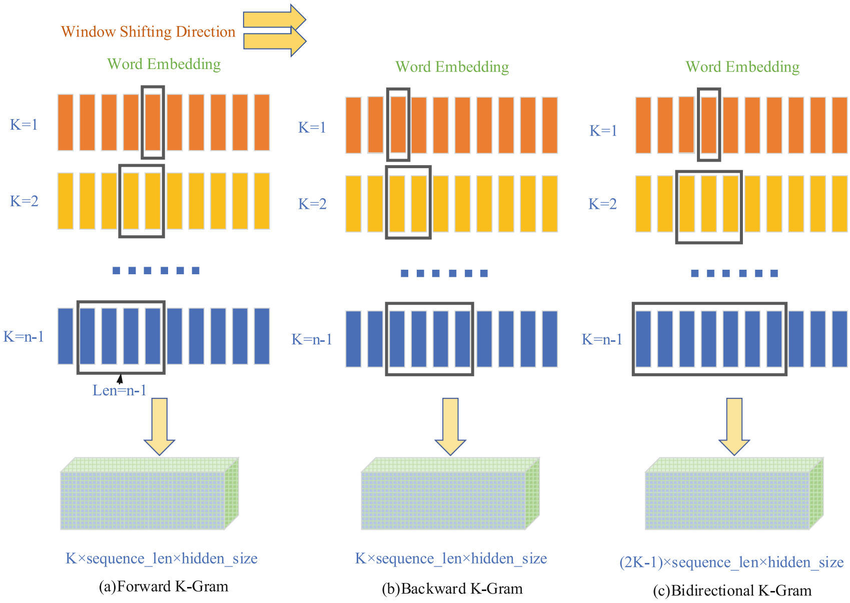

**Fig. 2.** K-Gram Exponential Decay

The K-Gram exponential decay scheme refines word embedding by aggregating information from surrounding token embedding. To reduce the computational cost, we parallelize the computation using a circular shift operation to improve computational efficiency and reduce running time. In addition, inspired by the decay factor of MDP in reinforcement learning, we exponentially decay the weights of the token embedding within the window to highlight the effect of relative position on the token embedding. The exponential decay weight $\alpha$ is a hyper-parameter, and we fix it to **0.8** in experiments. Given that token embedding fusion involves directionality, we consider forward K-Gram, backward K-Gram, and their combined form as choices for our downstream processing. The specific forms of the three K-Grams are shown in Fig. 2, and the formulas are as follows:

**Forward K-Gram:**

$$w_i^k = \sum_{j=0}^{k} e_{i-j} \times \alpha^j \tag{1}$$

**Backward K-Gram:**

$$w_i^k = \sum_{j=0}^{k} e_{i+j} \times \alpha^j \tag{2}$$

**Bidirectional K-Gram:**

$$w_i^k = \sum_{j=0}^{k} e_{i-j} \times \alpha^j + \sum_{j=1}^{k} e_{i+j} \times \alpha^j \tag{3}$$

where $w_i^k$ denotes the word embedding obtained after K-Gram Exponential Decay, and $e$ denotes the token embedding output by text encoder. $j$ stands for the relative distance, and the greater the relative distance, the lower its weight.

## 3.2 Image Representation Module

We employ Swin-Transformer [13] as the image encoder in the era when Transformer architectures were widely used in computer vision. An increasing body of research contends that the Swin-Transformer, which inherits the notion of CNN hierarchical receptive fields, may be the ideal replacement for CNN. Specifically, Swin is separated into four stages, each of which results in a smaller input feature map and a larger receptive field. Each stage consists of a Patch Merging module and a Swin-Transformer Block. The role of Patch Merging is to downsample the image, similar to the pooling layer in CNN. The Swin-Transformer Block consists of Window Multi-Head Self-Attention,Shifted-Window Multi-Head Self-Attention, Layer Norm, MLP and Residual Connection.

Swin-Transformer and MLP are used to transform the image in order to obtain the final image embedding, then image embedding is utilized to calculate product similarity.

## 3.3 Contrastive Learning Objective

Since identical or different product always occur in pairs in this product mining task, we apply Cosine Sentence (CoSENT) to explicitly distinguish the difference among items. Inspired by Circle Loss, we add weight and margin to CoSENT to increase the weight of difficult pairs and separate them from each other. Inspired by Curricular Loss, we gradually increase the weight of the difficult sample pairs during the training process, so that the model gradually focuses on the difficult sample pairs. The following is an introduction to CoSENT, Circle-CoSENT and Curricular-CoSENT respectively.

**CoSENT.** The essence of CoSENT is comparative learning based on sample pairs, loss function is as follows:

$$\mathcal{L} = \log\left(1 + \sum_{(i,j)\in\Omega_{pos},(u,v)\in\Omega_{neg}} e^{\lambda\left(cos(e_u,e_v) - cos(e_i,e_j)\right)}\right) \tag{4}$$

where $\Omega_{pos}, \Omega_{neg}$ are positive sample pairs set and negative sample pairs set, respectively. $e_u$ represents representation of product $u$. $\lambda$ is a hyper-parameter set to **20** in our experiment.

The optimization goal of CoSENT is to increase the cosine similarity of positive sample pairs while decreasing the cosine similarity of negative sample pairs. By subtracting the cosine similarity of positive sample pairs from the cosine similarity of negative sample pairs, it increases the distance between positive and negative sample pairs. The benefit of CoSENT is that the threshold for identifying whether a sample pair is a positive or negative pair does not need to be predetermined.

**Circle-CoSENT.** We add weight and margin to CoSENT, the loss function of Circle-CoSENT is as follows:

$$\mathcal{L} = \log\left(1 + \sum_{(i,j)\in\Omega_{pos},(u,v)\in\Omega_{neg}} e^{\lambda\left(\omega_{neg}\left(cos(e_u,e_v)+m_{neg}\right) - \omega_{pos}\left(cos(e_i,e_j)-m_{pos}\right)\right)}\right) \tag{5}$$

$$\omega_{neg} = \frac{cos(e_u,e_v)+1}{2}, \omega_{pos} = 1 - \frac{cos(e_i,e_j)+1}{2} \tag{6}$$

where $\omega_{pos}, \omega_{neg}, m_{pos}, m_{neg}$ are positive sample pairs weight, negative sample pairs weight, positive sample pairs margin, negative sample pairs margin, respectively. $\omega_{neg}, \omega_{pos}$ imply respectively that negative sample pairs are more difficult the closer they are to 1 and positive sample pairs are more difficult the closer they are to 0. Furthermore, we hope that the positive sample pair will be accurately predicted even if $m_{pos}$ is subtracted and the negative sample pair will be correctly predicted even if $m_{neg}$ is added, further separating the positive and negative sample pairs.

**Curricular-CoSENT.** We compel CoSENT to master the straightforward sample pairs before moving on to the challenging ones, the loss function of Curricular-CoSENT is as follows:

$$\mathcal{L} = \log\left(1 + \sum_{(i,j)\in\Omega_{pos},(u,v)\in\Omega_{neg}} e^{\lambda\left(f\left(cos(e_u,e_v)\right) - f\left(cos(e_i,e_j)\right)\right)}\right) \tag{7}$$

$$f\left(cos(\cdot,\cdot)\right) = \begin{cases} cos(\cdot,\cdot), & \text{if } (\cdot,\cdot) \text{ is easy sample pair} \\ cos(\cdot,\cdot)\left(t + cos(\cdot,\cdot)\right), & \text{if } (\cdot,\cdot) \text{ is hard sample pair} \end{cases} \tag{8}$$

where $t$ grows gradually from 0 to 1. Negative sample pairs greater than a particular threshold and positive sample pairs under a particular threshold are challenging samples.

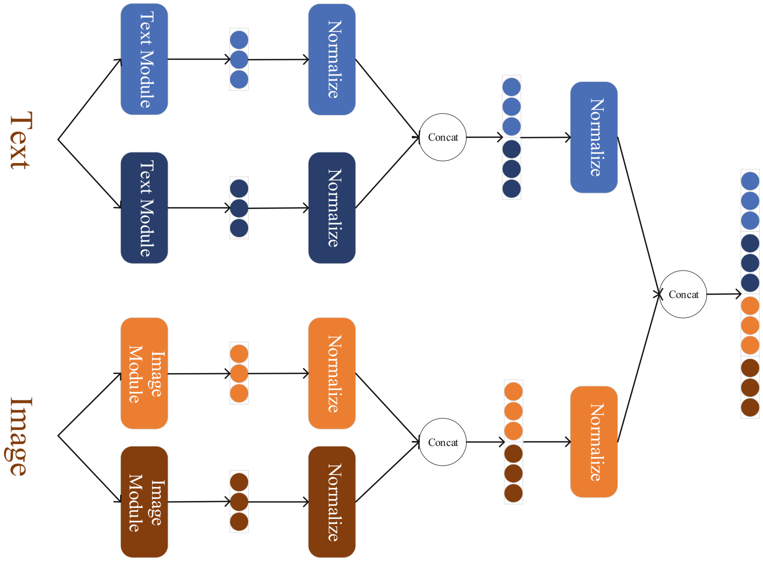

**Fig. 3.** Model ensemble

### 3.4 Model Ensemble

We use four models for integration, RoBERTa-Base for text representation $\mathbb{R}^{128}$, RoBERTa-Large for text representation $\mathbb{R}^{128}$, Swin-Transformer for image representation $\mathbb{R}^{256}$, and Swin-Transformer for image representation $\mathbb{R}^{512}$. Separate concatenations of the two text representations and the two image representations are performed, and then the text representation is concatenated with the image representation. Note that normalization is required before each concatenation. The model ensemble is shown in Fig. 3.

## 4 Experiments

### 4.1 Dataset

We conduct experiments on CCKS2022 identical product mining competition dataset. The training set contains 71,452 product information and 57,741 pairs of labeled product pairs data, and the validation set contains 16,876 product information and 20,707 pairs of unlabeled product pairs data, and the test set contains 17,132 product information and 15,909 pairs of unlabeled product pairs data. The product information data contains ten features such as id, industry_name, cate_name, cate_id, cate_name_path, cate_id_path, image_name, title, item_pvs, and sku_pvs.

Furthermore, we divide the local-train set, local-valid set, and local-test set on the basis of the training set in the ratio of 8:1:1.

### 4.2 Data Pre-processing

We primarily preprocess the product information data's item_pvs feature. First, we remove the redundant values and overlength values from them. Then, as new features, we copy the values of brand, item number, and model number from item_pvs. Finally, we remove some symbols to shorten the text.

### 4.3 Experimental Setup

We choose RoBERTa-Base, RoBERTa-Large, and Swin-Transformer-Large as our pre-trained models. For the text module, we set the learning rate to 2e–5, the batch size to 128, the epoch to 30, the maximum sequence length to 256, and the threshold is set to 0.8. And for the image module, the learning rate is 1e–5, the batch size is 32, the epoch is 50 and the threshold is set to 0.76.

In addition, while training the model, we only unfreeze the last three layers of RoBERTa and the last two layers of Swin-Transformer.

### 4.4 Post-processing

We add some rules to do further threshold processing when concatenating the final text representation and image representation. For example, if the brand and model number of two products differ but the image similarity is low, we may choose to increase their threshold. In addition, if two products have a high image or text similarity and the same brand, we choose to lower the threshold.

### 4.5 Experimental Results

The main results on local-valid data are shown in Table 1. As can be seen from the table, the Circle-CoSENT, Curricular-CoSENT and K-Gram Exponential Decay (KGED) we adopted have some improvement on the local-valid data. The image module performs best on local-valid data, with an F1 score of 90.21%.

The experimental results of our methodology using validation data are shown in Table 2. We can find that Circle-CoSENT has some degradation on validation data, and we guess that this may be due to the difference between the two datasets. Therefore, we only utilize the ordinary CoSENT, subsequently. Text model ensemble and image model ensemble represent the ensemble of RoBERTa-Base and RoBERTa-Large, and the ensemble of Swin-Transformer-Large and Swin-Transformer-Large, respectively. The results of Text model

**Table 1.** The main results on local-valid data.

| Model | P | R | F1 |
|---|---|---|---|
| RoBERTa | 87.36 | 84.27 | 85.79 |
| RoBERTa + Circle-CoSENT | 86.20 | 86.70 | 86.45 |
| RoBERTa + Curricular-CoSENT | 86.90 | 85.90 | 86.40 |
| RoBERTa (KGED) | 87.34 | 86.73 | 87.03 |
| Swin-Transformer | **88.41** | **92.20** | **90.27** |

ensemble (KGED) are better than those of Text model ensemble, which proves the effectiveness and robustness of KGED. Additionally, the model ensemble effect has improved significantly, indicating a higher level of complementarity between the input from various modalities. This further demonstrates the effectiveness of our concatenate-based multimodal information fusion method in this task. Given the decreasing effectiveness of Model Ensemble (KGED) with post-processing, we infer that the current image model ensemble is implemented more effectively with text model ensemble than it is with text model ensemble (KGED). The model ensemble and post-processing combination, with an F1 score of 90.57%, makes the best results overall.

**Table 2.** The main results on valid data.

| Model | P | R | F1 |
|---|---|---|---|
| RoBERTa | 82.65 | 83.51 | 83.07 |
| RoBERTa + Circle-CoSENT | 82.11 | 83.05 | 82.58 |
| Text Model Ensemble | 84.36 | 84.78 | 84.57 |
| Text Model Ensemble (KGED) | 85.09 | 85.57 | 85.33 |
| Image Model Ensemble | 85.49 | 86.15 | 85.82 |
| Model Ensemble | 88.50 | 90.69 | 89.58 |
| Model Ensemble + Post-processing | **90.23** | **90.92** | **90.57** |
| Model Ensemble (KGED) + Post-processing | 89.96 | 90.52 | 90.24 |

## 5 Conclusion

In this paper, we apply multimodal representation learning to product matching. For the text representation module, we propose K-Gram Exponential Decay scheme with CNN to refine text representation. In order to enhance the distance between matched and unmatched product pairs, we also utilize sample pair-based contrastive learning. Last but not least, we combine the two text representation modules with the two image representation modules to lower the variance of the separate models and enhance the ability of the product to be represented. The experimental results demonstrated that our model performs significantly with an F1 score of 90.57% on the validation set and 90.77% on the test set, and is ranked first in the identical product mining competition for CCKS2022. Our long-term research goal is to improve the K-Gram Exponential Decay in the text representation module such that, when combined with the image representation module, it can better represent product.

**Acknowledgement.** This work was supported by the grants from National Natural Science Foundation of China (No. 62072423), and the USTC Research Funds of the Double First-Class Initiative (No. YD2150002009).

## References

1. Zhang, N., et al.: AliCG: fine-grained and Evolvable Conceptual Graph Construction for Semantic Search at Alibaba. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining, pp. 3895–3905 (2021). ACM, Virtual Event Singapore. https://doi.org/10.1145/3447548.3467057
2. Fang, Y., Wang, J., Jia, L., Kin, F.W.: Shopee price match guarantee algorithm based on multimodal learning. In: 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE), pp. 84–87 IEEE, SC, USA (2021). https://doi.org/10.1109/CSAIEE54046.2021.9543217
3. Sun, Y., et al.: Circle loss: a unified perspective of pair similarity optimization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6397–6406 IEEE, Seattle, WA, USA (2020). https://doi.org/10.1109/CVPR42600.2020.00643
4. Huang, Y., et al.: CurricularFace: adaptive curriculum learning loss for deep face recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5900–5909. IEEE, Seattle, WA, USA (2020). https://doi.org/10.1109/CVPR42600.2020.00594
5. Tracz, J., Wójcik, P.I., Jasinska-Kobus, K., Belluzzo, R., Mroczkowski, R., Gawlik, I.: BERT-based similarity learning for product matching, pp. 66–75 (2020)
6. Li, J., Dou, Z., Zhu, Y., Zuo, X., Wen, J.-R.: Deep cross-platform product matching in e-commerce. Inf. Retrieval J. **23**(2), 136–158 (2019). https://doi.org/10.1007/s10791-019-09360-1
7. Li, H., et al.: Path-based deep network for candidate item matching in recommenders. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1493–1502 ACM, Virtual Event Canada (2021). https://doi.org/10.1145/3404835.3462878
8. Peeters, R., Bizer, C.: Supervised contrastive learning for product matching (2022). https://doi.org/10.1145/3487553.3524254
9. Wu, C., Wu, F., Huang, Y., Xie, X.: User-as-Graph: user modeling with heterogeneous graph pooling for news recommendation. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, pp. 1624–1630. International Joint Conferences on Artificial Intelligence Organization, Montreal, Canada (2021). https://doi.org/10.24963/ijcai.2021/224

10. Yu, J., Jiang, J., Yang, L., Xia, R.: Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3342–3352. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.306
11. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. http://arxiv.org/abs/1907.11692 (2019)
12. Yao, H., Liu, H., Zhang, P.: A novel sentence similarity model with word embedding based on convolutional neural network: sentence similarity model with word embedding based on convolutional neural network. Concurrency Computat. Pract. Exper. **30**, e4415 (2018). https://doi.org/10.1002/cpe.4415
13. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9992–10002. IEEE, Montreal, QC, Canada (2021). https://doi.org/10.1109/ICCV48922.2021.00986