IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 57, No. 4, August, 2024, pp. 1207-1214
Journal homepage: http://iieta.org/journals/jesa

Product Matching with Two-Branch Neural Network Embedding

Agus Mistiawan[1*], Derwin Suhartono[2]

[1] Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia
[2] Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding Author Email: agus.mistiawan@binus.ac.id

https://doi.org/10.18280/jesa.570427

Received: 2 July 2024
Revised: 7 August 2024
Accepted: 16 August 2024
Available online: 27 August 2024

Keywords:
deep learning, BERT, CharacterBERT, EfficientNet, ArcFace, product matching

ABSTRACT

E-commerce platforms play a crucial role in facilitating transactions between buyers and sellers, with technological advancements significantly influencing consumer behaviors. Efficiently managing product catalogs is essential, particularly for identifying and matching products across various channels. This paper explores deep learning techniques for product matching, leveraging both text and image modalities to enhance the accuracy and efficiency of this process. We propose a novel approach using a branch neural network embedding space integrated with K-nearest neighbors (KNN), treating image and text as distinct modalities. For text embedding, we utilize pre-trained BERT and CharacterBERT models, while for image embedding, we employ EfficientNet. Our methodology incorporates the ArcFace loss function to enhance intra-class compactness and inter-class discrepancy, thereby improving classification performance. Our results demonstrate that integrating multimodal embeddings with advanced loss functions like ArcFace significantly enhances the performance of product matching systems. This approach offers valuable insights for developing robust e-commerce platforms.

1. INTRODUCTION

E-commerce serves as a digital marketplace facilitating transactions between sellers and buyers. The technological progressions in e-commerce significantly influence consumer buying habits, demonstrated by the vast number of visits to e-commerce sites and transactions accounting for over 19 percent of retail sales worldwide [1]. Numerous platforms exist for e-commerce, allowing sellers to market their products across multiple channels and efficiently oversee all their e-commerce operations through a unified platform known as omnichannel [2]. Different e-commerce platforms offer vast amounts of products. Matching these products across different platforms is a pivotal aspect for this system and a challenging task, as it requires accurately identifying similar items within the e-commerce ecosystem, as visually presented in Figure 1. Product matching serves as a technique for identifying identical products within a catalog in response to a specific query product [3]. The intricate challenges associated with product matching are highlighted by Shah et al. [4], wherein product duplicates emerge as a primary source of adverse product encounters, complicating the matching process due to multiple instances of the same product containing overlapping information. Conversely, seeking similar products can employ the same methodology, differing only in the desired outcome.

Several research initiatives have explored product matching, employing a variety of deep-learning techniques and methodologies. These encompass classification approaches for categorizing products based on category leaf [4-6], contrastive learning utilizing multi-modal image and text [7], and more recently, the two-stage retrieval-enhanced dual encoder [3]. The majority of these studies aim to match products by either integrating attributes or utilizing a single attribute. Determining similar products can be achieved through product titles, product images, or a combination of both attributes. Products are distinguished based on the similarity or dissimilarity between their images, text descriptions, or attributes. Enforcing strict alignment between product image and text yields limited utility in this context. Each attribute should be evaluated independently, and subsequently, the most appropriate matches should be consolidated.

This research paper aims to implement a novel approach using a branch neural network embedding space, integrated by K-nearest neighbors (KNN) where both image and text are treated as distinct modalities. The KNN-embedding space approach entails measuring the distance between text and image embeddings. For each modality, we employ pre-trained models, specifically BERT for text and EfficientNet for images. These models generate separate embedding spaces, subsequently utilized in the KNN framework. The recommendation results are aggregated based on distance and a predefined threshold. Moreover, it is essential to distinguish between measuring the similarity of individual products and classifying them into broader product categories. In the former case, products are compared without considering their category, which can result in a large number of labels with only a small dataset for each label. To address this, ArcFace

[8], a method well-known for its effectiveness in face recognition, can be employed to enhance intra-class compactness and inter-class discrepancy [9]. By applying ArcFace to each model, the challenge of handling a vast number of labels can be mitigated.

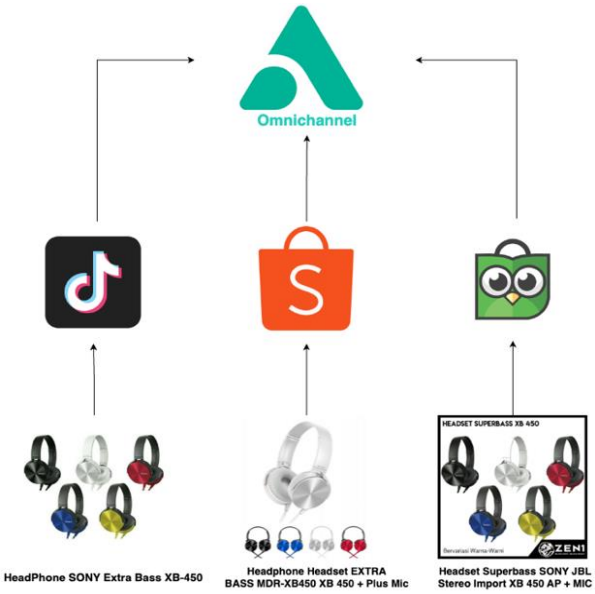

Figure 1. Product management for an omnichannel platform

2. LITERATURE REVIEW

The predominant focus of product matching research revolves around identifying similarities within and between classes. Intra-class similarity pertains to likeness within a specific group or category, involving the grouping of products based on specific attributes. Conversely, inner-class similarity directly addresses the likeness between individual entities or objects, such as comparing two products. Various methodologies have been proposed for text-based product categorization. Li et al. [10], Shah et al. [4], and Gupte et al. [11] introduced text-based product categorization methods, leveraging textual information like product titles, descriptions, and brand attributes. An alternative approach presented by Arroyo et al. [9] involves extracting textual information from images to ascertain product categories, while Tracz et al. [6] explored zero-shot learning with a triplet loss objective to learn product similarity using multiple representations and attributes.

Mapping products into categories tackles the challenge of intra-class product matching by grouping products based on similar attributes within specific categories. However, this method may not effectively discern individual product similarities, potentially leading to instances where similar products are erroneously classified as distinct items. Matching within classes using multimodal representations often employs siamese neural networks as the foundational model. Gupte et al. [11] refined product embeddings by merging image and text embeddings with a siamese network to ascertain product presence in a catalog. Similarly, Ko [12] combined image and text embeddings and inputted them into a multi-layer perceptron to evaluate similarity. Mazhar et al. [7] conduct a study on join text-embedding, utilizing the embedding network developed by Wang et al. [13]. The researchers propose the use of Multimodal neural networks, incorporating techniques such as Element-Wise Multiplication (MNN-EM) and Bidirectional Triplet Loss (MNN-BTL). The aim is to compare textual representations with combined text-image representations in order to facilitate product matching. This methodology demonstrates promising results, particularly in the context of siamese neural networks. These networks achieve improved performance by learning embeddings through the incorporation of an additional contrastive loss, although effective training necessitates the use of pairwise products.

Another approach involves utilizing information retrieval and implementing a two-stage training method with dual encoders [3] that utilize text as a unimodal input. The initial stage follows a conventional retrieval model approach, while the subsequent stage improves the outcomes of the first stage by incorporating additional data with positive labels. Cross-modal retrieval or text-to-image matching has been proposed by Gao et al. [14] in a model called FashionBERT. Similar to other dual encoder models, FashionBERT uses separate encoders for text and images, and the outputs of these encoders are then combined to form a joint representation of the fashion item. Likewise, Zhuge et al. [15] introduced KaleidoBERT, which employs a kaleidoscope data augmentation technique to generate multiple augmentations of images, subsequently aligning the image and text representations.

On product matching task BERT have been used several times and easy to fine-tuning, intermediate fine-tuning of BERT for product matching [16]. Mazhar et al. [7] and Falzone et al. [17], particularly, employ character-level CNNs to process product titles [18], aiming to enhance performance on product matching tasks by refining the representation of text data associated with the products. Additionally, Ma et al. [19] propose CharacterBERT, a character-aware pre-trained language model, as an improvement over previous models. Moreover, the integration of ArcFace [8] has tackled inter-class issues by generating distinct embedding distances for different labels or categories. Our work draws inspiration from multi-modal retrieval approaches to explore product similarity, incorporating the implementation of ArcFace loss for each modal embedding.

3. METHODOLOGY

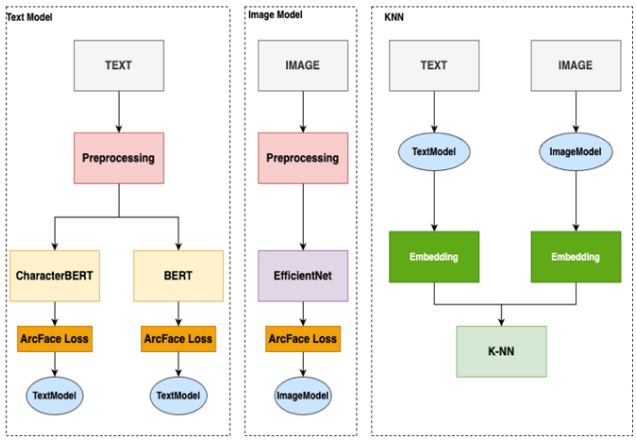

Figure 2. Proposed model network embedding with K-NN

This section provides a detailed explanation of the proposed model and methodology. The aim is to collect and measure product similarity using both text and image modalities. The

product similarity assessment is conducted on an industrial product list available online. The dataset comprises text and images, which will be processed independently using pre-trained BERT-family and EfficientNet models. The detailed proposed model is illustrated in Figure 2.

3.1 Text embedding

In the proposed model, text embedding plays a crucial role in extracting features from neural networks to determine the class of a product. The integration of BERT and CharacterBERT models enhances the capability to process text data more effectively. BERT, a pre-trained model developed by Google [20], employs a bidirectional approach to understand the context of words in a sentence, transforming natural language processing tasks. On the other hand, CharacterBERT [21], a variant of BERT, utilizes a Character Convolutional Neural Network to process entire words without splitting them into subword units. This modification makes CharacterBERT more suitable for handling out-of-vocabulary words and noisy inputs, especially in specialized domains such as medical text processing.

The pre-training process for CharacterBERT involves tasks similar to BERT, including Masked Language Modeling and Next Sentence Prediction. However, CharacterBERT predicts entire words instead of masked subword units, leveraging character-level context to enhance its understanding and robustness. This approach has demonstrated improvements in various NLP tasks within specialized domains, showcasing its effectiveness in processing unique and diverse vocabularies.

3.1.1 BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers. Its groundbreaking impact on natural language processing (NLP) tasks stems from its unique approach of comprehending the context of a word in a sentence bidirectionally. Unlike unidirectional models that only consider preceding words or bidirectional RNNs that consider both preceding and succeeding words, BERT takes into account all surrounding words. BERT is built on the transformer architecture [22], which employs attention mechanisms to capture dependencies between words, regardless of their position in the text. The primary innovation of BERT lies in its training strategy, which involves two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). By pre-training on extensive amounts of text data and subsequently fine-tuning on specific tasks, BERT achieves state-of-the-art performance on various NLP benchmarks, including question answering and sentiment analysis.

3.1.2 CharacterBERT

CharacterBERT is a variant of the original BERT model designed to enhance its performance and robustness, particularly in specialized domains like medical text processing. Unlike BERT, which employs a wordpiece tokenization system that breaks words into subword units, CharacterBERT utilizes a Character Convolutional Neural Network (Character-CNN) [18] to generate word representations based on character-level information. This approach allows CharacterBERT to process entire words without fragmenting them, thereby improving its ability to handle out-of-vocabulary words, misspellings, and noisy inputs common challenges in real-world text data.

The key advantage of CharacterBERT lies in its ability to adapt to domain-specific vocabularies that may not be adequately represented in general-purpose wordpiece vocabularies. For instance, in the medical domain, where specialized terminology is prevalent, CharacterBERT can accurately represent complex terms without requiring a pre-defined extensive vocabulary. This flexibility makes CharacterBERT particularly effective in environments with unique and diverse vocabularies.

The model architecture of CharacterBERT is rooted in the Character-CNN, as shown in Figure 3, which generates context-independent token representations by capturing the nuances of character-level features. These token embeddings are then combined with position and segment embeddings, similar to BERT, before being processed by Transformer layers. However, unlike BERT, which assigns multiple subword embeddings to a single word, CharacterBERT assigns a singular contextual representation to each token. This method enhances the model's ability to understand and process complete words, leading to improved generalization across different forms of a word. The pre-training process of CharacterBERT closely resembles that of BERT, involving tasks such as Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). However, instead of predicting individual masked subword units, CharacterBERT predicts entire words, leveraging character-level context to enhance comprehension and robustness. This design has shown improvements across various NLP tasks, particularly in specialized domains, by effectively handling complex and varied text inputs.

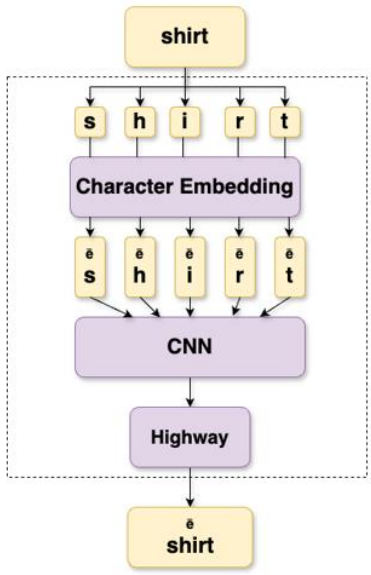

Figure 3. CharacterBERT embedding

In summary, CharacterBERT's architecture centered around character-level processing provides significant advantages over regular BERT, especially in tasks that involve noisy data, out-of-vocabulary words, and specialized vocabularies. This makes CharacterBERT a powerful tool for domains where precise language understanding is crucial.

3.2 Image embedding

Feature extraction from images typically involves using Convolutional Neural Networks (CNNs) to transform images

into feature vectors. There are many pre-trained CNN models available that yield excellent results, making them suitable for image embedding. These pre-trained models can be fine-tuned for specific tasks, providing a strong foundation for various image processing applications. EfficientNet [23] is a CNN architecture that aims to enhance both accuracy and efficiency of models through a unique scaling method. It offers a more systematic approach to scaling models compared to traditional methods, which often increase network dimensions like depth or width arbitrarily. EfficientNet achieves this by utilizing a compound scaling method that uniformly scales the network's depth, width, and resolution using carefully chosen scaling coefficients.

Instead of independently scaling network dimensions, EfficientNet uses a compound coefficient to scale depth, width, and resolution together. This balanced approach ensures that the network is optimized in all dimensions, leading to improved performance and efficiency. EfficientNet's base architecture was developed using NAS, which automates the design process to find the most efficient architecture within given constraints. The resulting network, called EfficientNet-B0, serves as the baseline for further scaling. The models, ranging from EfficientNet-B0 to EfficientNet-B7, demonstrate significant improvements in accuracy and computational efficiency. For example, EfficientNet-B7 achieves state-of-the-art accuracy on ImageNet while being 8.4 times smaller and 6.1 times faster than previous models. The combination of these innovations allows EfficientNet to outperform many existing CNNs in terms of both speed and accuracy, making it a powerful tool for tasks that require high performance with limited computational resources.

3.3 ArcFace loss

ArcFace, also known as Additive Angular Margin Loss [8], is a widely used loss function in face recognition tasks. It enhances the compactness within classes and the discrepancy between classes by utilizing an Additive Angular Margin Loss. The main innovation of ArcFace lies in the normalization of feature vectors and weights, projecting them onto a hypersphere, and applying an angular margin penalty to improve classification performance. This approach establishes distinct decision boundaries between classes, which is particularly advantageous for tasks with high intra-class variability, such as face recognition.

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,\,j\neq y_i}^{n}e^{s\cos\theta_j}} \tag{1}$$

ArcFace normalizes both the feature vectors and weight vectors to reside on a unit hypersphere. This normalization ensures that the classification decision is solely based on the angular distance between features and weights. The margin m is added to the angle θ_{yi} corresponding to the correct class. This margin poses a challenge for the classifier to assign the correct class unless the feature vector is very close to the correct weight vector. Consequently, it enhances the compactness within classes and the discrepancy between classes. The scale factor s is employed to control the magnitude of the logits, which stabilizes the training process by preventing the logits from becoming too small. By incorporating an angular margin, ArcFace ensures that the feature vectors of samples from the same class are closer together (intra-class compactness) and farther apart from feature vectors of other classes (inter-class discrepancy). This results in well-defined decision boundaries, augmenting the model's capacity to effectively differentiate between different classes.

ArcFace has demonstrated success beyond face recognition, such as in text-to-speech (TTS) data augmentation for Automatic Speech Recognition (ASR) training. In this context, ArcFace aids in generating improved pseudo-labels by optimizing the representation of discrete speech units, showcasing its versatility in handling various types of data. While ArcFace is primarily associated with visual and speech data, its principles can be adapted for text applications to enhance feature discrimination. This adaptability is demonstrated in its utilization with Hidden-Unit BERT (HuBERT), where it aids in optimizing the representation of speech units, indicating potential for similar applications in text classification tasks where distinguishing subtle differences between classes is crucial.

3.4 K-nearest neighbors

The KNN algorithm is widely used in information retrieval tasks because of its simplicity and effectiveness. In these tasks, the goal is to find items in a dataset that are most similar to a given query item. K-NN is well-suited for this purpose as it directly compares the query item with all items in the dataset to determine the most similar ones. Each element in the dataset, as well as the query element, is represented by a feature vector. These vectors are formed using various attributes, such as text embeddings, image features, or other pertinent characteristics. To assess the similarity between the query element and each element in the dataset, a distance metric is employed. Commonly used metrics include Euclidean distance, cosine similarity, or Manhattan distance. The selection of the metric depends on the data and the specific retrieval task. Subsequently, the algorithm identifies the 'k' elements in the dataset that have the smallest distance (or highest similarity) to the query element. These 'k' elements are deemed the closest neighbors. Ultimately, the closest neighbors are returned as the most pertinent elements for the given query. For instance, in a content-based image retrieval system, this would involve retrieving the images that bear the closest resemblance to the query image.

4. EXPERIMENTS

4.1 Dataset

The dataset is derived from industrial e-commerce and has been published online [24]. Originally, it was prepared for product duplicate detection. Despite having different objectives, the structure and model of the data can be applied to various outputs. The dataset consists of 34,250 rows of text and image data, including 32,413 images and 11,014 labels. Each row contains an ID, image, text, and label, in Table 1 shown example of dataset.

Prior to analysis, the dataset undergoes preprocessing. The text is converted to lowercase and stop words are removed. Additionally, the images are resized to 512×512 pixels. However, it is important to note that the distribution of the dataset is imbalanced.

Table 1. Example of product image and name

ID	Image	Name	Label
train_33431845		T-shirt Currently Kaos Wanita	4254526477

Figure 4 illustrates that the most frequent label occurs in 50 rows, while some labels only appear in 2 rows. To address this issue and achieve a more balanced dataset, data augmentation methods are utilized. Labels with excessive data are down sampled, with a maximum of 10 rows per label. For smaller datasets, as suggested Mazhar et al. [7], alternative product pair is performed by generated the minority or smaller datasets by combining each alternative text and image.

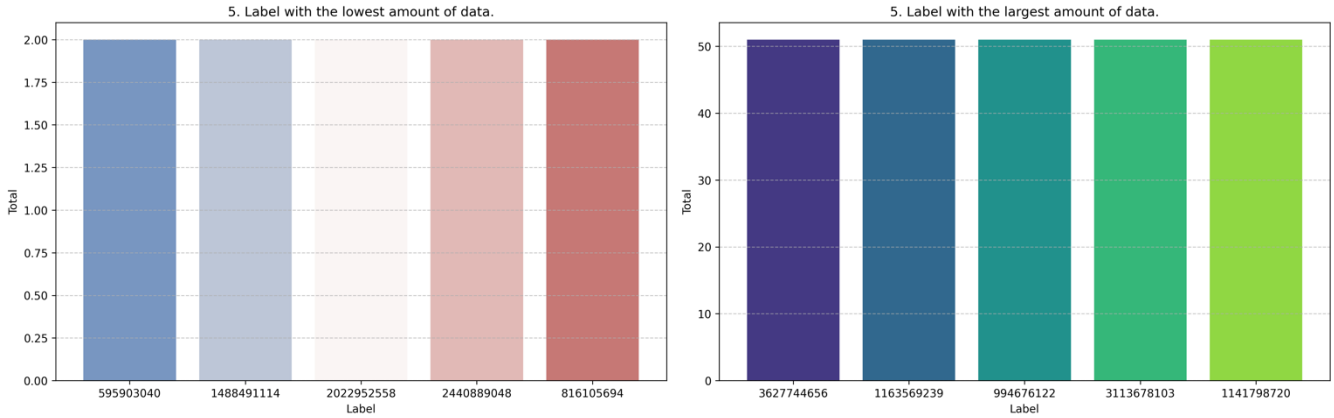

Figure 4. Distribution of dataset label

Table 2. A representative example of a noisy dataset

Image	Name	Label
	Packing Tambahan Bubble Wrap/Kardus Bekas	1960893869
	PACKING TAMBAHAN BUBBLE WRAP	4198148727
	PACKING TAMBAHAN BUBBLE WRAP	4198148727
	BUBBLE WARP	2403374241

In this paper, we undertake a comprehensive examination of the presence of noise in image datasets, with a specific focus on the occurrence of multiple labels assigned to a single image. Our meticulous analysis uncovers that approximately 0.51% of the images within the dataset demonstrate label ambiguity, as they are assigned with more than two labels. It is noteworthy that, despite similarities in the product names, the labels themselves vary, thereby emphasizing the intricate nature of the noise. To enhance understanding, we present illustrative examples of noisy data in Table 2, offering valuable insights into the complexities of this issue. Additionally, we deliberate on the potential repercussions of eliminating noisy datasets on model performance, advocating for a nuanced approach to data preprocessing.

4.2 Evaluation metrics

The proposed model will be evaluated using well-established information retrieval metrics, including precision, recall, and F1-score. Its performance will be assessed on test data, and the results will be analyzed using these metrics, as summarized in Table 3, which presents the recall and precision metrics.

Table 3. Recall and precision metrics

	Relevant	Not Relevant
Retrieved	a	b
Not Retrieved	c	d

Precision is comparing total number of products retrieved that are relevant with total number of products that are retrieved.

$$Precision = \frac{a}{a+b} \tag{2}$$

Recall is comparing total number of products retrieved that are relevant with total number of relevant products.

$$Recall = \frac{a}{a+c} \tag{3}$$

F1-score is the evaluation result calculated by combining the precision and recall values.

$$F1-Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

4.3 Result

The model was built using pre-trained models available in

TensorFlow and Keras, except for CharacterBERT, for which we used a pre-trained model from Huggingface [25]. For the text modality, we applied identical parameters to both BERT and CharacterBERT, including the Adam optimizer with a learning rate of 1e-5, 25 epochs, sparse categorical cross-entropy loss, softmax activation, and an additional ArcMargin layer. The network was trained and tested on both the original and augmented datasets, with TF-IDF, a method commonly used in query retrieval [26], serving as a baseline comparison. For the image modality, we use EfficientNetB3, and similar parameters were used, except for the learning rate, which was adjusted using a learning rate scheduler. The learning rate initially increased linearly over the first few epochs to start training with a low learning rate, reducing the risk of instability, and then underwent exponential decay. The weight of each model is utilized for generating text and image embeddings, which are subsequently integrated into the K-NN algorithm [27].

Table 4 depicts the evaluation results obtained from the original dataset. BERT and CharacterBERT score well, but CharacterBERT has a higher recall, resulting in a higher F1-Score. Although the precision scores for the text modalities are praiseworthy, the image modalities exhibit a noticeable decline in performance. This incongruity could potentially be attributed to factors such as dataset imbalance and the existence of noisy data. The integration of both image and text modalities leads to an enhancement in the recall score, albeit accompanied by a slight reduction in precision. This implies that the fusion of information from both modalities augments the model's capability to retrieve pertinent instances, thereby mitigating, to some extent, the impact of imbalanced and noisy data.

The evaluations on the augmented dataset are delineated in Table 5. To address the imbalance within the dataset, down sampling was employed, leading to the generation of alternative product instances for the minority class. This augmentation strategy aimed to bolster the representation of underrepresented classes, thereby fostering a more balanced dataset for evaluation.

Table 4. Evaluation result on original dataset

Model	Precision	Recall	F1-Score
TF-IDF	0.9182	0.5273	0.6111
BERT	0.9000	0.7221	0.7587
CharacterBERT	0.8830	0.7760	0.7922
EfficientNetB3	0.6667	0.8580	0.6600
BERT + EfficientNetB3	0.8410	0.8985	0.8327
CharacterBERT + EfficientNetB3	0.8290	0.9135	0.8373

Table 5. Evaluation result on augmented dataset

Model	Precision	Recall	F1-Score
TF-IDF	0.9294	0.5392	0.6333
BERT	0.9325	0.8458	0.8622
CharacterBERT	0.9527	0.9000	0.9106
EfficientNetB3	0.9317	0.8721	0.8726
BERT + EfficientNetB3	0.8934	0.9597	0.9056
CharacterBERT + EfficientNetB3	0.9052	0.9676	0.9186

The outcome is indeed promising, showcasing notable advancements in model performance when evaluated on the augmented dataset. Through meticulous down sampling and the generation of alternative product instances for underrepresented classes, the dataset underwent a transformation aimed at rectifying its inherent imbalance. This augmentation strategy not only bolstered the representation of minority classes but also fostered a more robust and comprehensive training environment for the model. Consequently, the model's capacity to generalize and effectively capture the clearer, more distinct patterns in the data was markedly enhanced. This improvement underscores the success in mitigating the challenges posed by label ambiguity, leading to a more robust learning process.

These results also suggest that combining models can leverage their strengths to achieve better overall performance, with CharacterBERT + EfficientNetB3 achieving the best overall performance, balancing precision and recall effectively. We emphasize that the recall performance improved substantially, highlighting a limitation of CharacterBERT, which struggled to capture similar products when the textual descriptions were slightly different, even though they conveyed the same contextual meaning. In contrast, the image-based comparisons were able to fully identify similar products, demonstrating the effectiveness of visual analysis in scenarios where textual descriptions vary but the underlying products remain the same, as shown in Figure 5. Conversely, Figure 6 illustrates instances where the image model failed to capture product similarities that the text model successfully identified.

(a) [LOGU] Tempelan kulkas magnet angka, tempelan angka magnet

(b) 10Pcs Magnet Kulkas Model Angka 0-9, Bahan Kayu, Warna-Warni

Figure 5. Product match found by image model

(a) Headphone MDR XB 450 / XB450 / XB-450 / EXTRA BASS KABEL

(b) HEADPHONE BANDO HANDSFREE EXTRA BASS XB450 - RELAXING

Figure 6. Product match found by text model

5. CONCLUSION

In conclusion, this paper presents a novel approach to product matching in e-commerce through the integration of

text and image modalities using deep learning techniques. The proposed model leverages pre-trained BERT and CharacterBERT models for text embedding, EfficientNet for image embedding, and the ArcFace loss function for enhanced feature discrimination. By employing K-nearest neighbors (KNN) algorithm, the model measures product similarity based on the distances between embeddings in a unified embedding space. Through experimentation on an industrial e-commerce dataset, the model demonstrates promising results in product matching, even in the presence of noise such as label ambiguity. Evaluation metrics including precision, recall, and F1-score are utilized to assess the model's performance, which showcases its effectiveness in retrieving relevant products. Overall, this research contributes to the advancement of product matching techniques in e-commerce, offering insights into the integration of text and image modalities for more accurate and efficient product recommendations. Future work, applying image augmentation to generate variations of images could help the model learn more robust features while reducing overfitting to noisy examples. Another approach is multi-label learning, where the model is trained to recognize all possible labels that might apply to images or text.

REFERENCES

[1] Coppola, D. (2024). E-commerce as percentage of total retail sales worldwide from 2021 to 2027. Statista. https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide.

[2] Lehrer, C., Trenz, M. (2022). Omnichannel business. Electron Markets 32(2): 687-699. https://doi.org/10.1007/s12525-021-00511-1

[3] Chiu, J. (2023). Retrieval-enhanced dual encoder training for product matching. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 216-222. https://doi.org/10.18653/v1/2023.emnlp-industry.22

[4] Shah, K., Kopru, S., Ruvini, J.D. (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), pp. 8-15. https://doi.org/10.18653/v1%2FN18-3002

[5] Ristoski, P., Petrovski, P., Mika, P., Paulheim, H. (2018). A machine learning approach for product matching and categorization. Semantic Web, 9(5): 707-728. https://doi.org/10.3233/SW-180300

[6] Tracz, J., Wójcik, P.I., Jasinska-Kobus, K., Belluzzo, R., Mroczkowski, R., Gawlik, I. (2020). BERT-based similarity learning for product matching. In Proceedings of Workshop on Natural Language Processing in E-Commerce, Barcelona, Spain, pp. 66-75.

[7] Mazhar, K.A., Brodtbeck, M., Gühring, G. (2023). Similarity learning of product descriptions and images using multimodal neural networks. Natural Language Processing Journal, 4: 100029. https://doi.org/10.1016/j.nlp.2023.100029

[8] Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., Zafeiriou, S. (2018). ArcFace: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4690-4699. https://doi.org/10.1109/CVPR.2019.00482

[9] Arroyo, R., Jiménez-Cabello, D., Martínez-Cebrián, J. (2020). Multi-label classification of promotions in digital leaflets using textual and visual information. arXiv preprint arXiv:2010.03331. https://doi.org/10.48550/arXiv.2010.03331

[10] Li, M., Kok, S., Tan, L. (2018). Don't classify, translate: Multi-level e-commerce product categorization via machine translation. arXiv preprint arXiv:1812.05774. https://doi.org/10.48550/arXiv.1812.05774

[11] Gupte, K., Pang, L., Pasumarty, H.V. (2021). Multimodal product matching and category mapping: Text+image based deep neural network. IEEE International Conference on Big Data, Orlando, FL, USA, pp. 4500-4505. https://doi.org/10.1109/BigData52589.2021.9671384

[12] Ko, E. (2021). Product matching through multimodal image and text combined similarity matching. Stockholm, Sweden.

[13] Wang, L., Li, Y., Huang, J., Lazebnik, S. (2018). Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(2): 394-407. https://doi.org/10.1109/TPAMI.2018.2797921

[14] Gao, D., Jin, L., Chen, B., Qiu, M., Li, P., Wei, Y., Wang, H. (2020). FashionBERT: Text and image matching with adaptive loss for cross-modal retrieval. arXiv:2005.09801. https://doi.org/10.48550/arXiv.2005.09801

[15] Zhuge, M., Gao, D., Fan, D., Jin, L., Chen, B., Zhou, H., Shao, L. (2021). Kaleido-BERT: Vision-language pre-training on fashion domain. arXiv Preprint arXiv:2103.16110. https://doi.org/10.48550/arXiv.2103.16110

[16] Peeters, R., Bizer, C., Glavaš, G. (2020). Intermediate training of BERT for product matching. Small, 745(722): 2-112.

[17] Falzone, S., Münster, T., Gühring, G. (2022). Measuring similarity for technical product descriptions with a character-level siamese neural network. In Tagungsband zum Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg, Technische Hochschule Ulm.

[18] Zhang, X., Zhao, J., LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in Neural Information Processing Systems, 28. https://doi.org/10.48550/arXiv.1509.01626

[19] Ma, W., Cui, Y., Si, C., Liu, T., Wang, S., Hu, G. (2020). CharBERT: Character-aware pre-trained language model. arXiv preprint arXiv:2011.01513. https://doi.org/10.18653/v1/2020.coling-main.4

[20] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (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, pp. 4171-4186. https://doi.org/10.18653/v1/N19-1423

[21] Boukkouri, H.E., Ferret, O., Lavergne, T., Noji, H., Zweigenbaum, P., Tsujii, J. (2020). CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In Proceedings of the 28th International Conference on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.609

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Pollosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762. https://doi.org/10.48550/arXiv.1706.03762

[23] Tan, M., Le, Q.V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946. https://doi.org/10.48550/arXiv.1905.11946

[24] Howard, A., Liew, C., Wong, M. (2021). Shopee-Price Match Guarantee. Kaggle. https://kaggle.com/competitions/shopee-product-matching.

[25] Eiji, R. (2023). BERT Character. Huggingface. https://huggingface.co/RafaelEiji/bert_character.

[26] Ramos, J.E. (2003). Using TF-IDF to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning, pp. 29-48.

[27] Deotte, C., Buvinic, M.K. (2021). RAPIDS cuML TfidfVectorizer and KNN. Kaggle. https://www.kaggle.com/code/cdeotte/rapids-cuml-tfidfvectorizer-and-knn.