Predicting Product Attributes on Etsy Marketplace using Deep Learning Techniques

Vaibhav Mudgal (22263496)
vaibhav.mudgal2@mail.dcu.ie

***Abstract*—Product classification is essential in e-commerce, where large amounts of products are sold through online marketplaces. It involves categorizing products into categories and subcategories based on their attributes, such as type, top category, color, etc. Accurate product classification can improve the customer experience by making it easier for them to find the products they are looking for and help online retailers to manage their inventory better. This project uses deep learning techniques to predict product attributes, namely top category, bottom category, and color, for items listed on the Etsy marketplace. The goal is to maximize the F1 score for each predicted attribute, improving product categorization and identification on the platform.**

I. Introduction

Accurately predicting product attributes such as top and bottom category IDs and color IDs is crucial for improving user experience and aiding in product discovery. However, a large amount of data and the complexity of the product attributes make this task challenging. Deep learning techniques have shown the potential to address these challenges and improve prediction accuracy. In this project, we aim to use deep learning techniques to predict product attributes for unseen items in the test dataset. The proposed model will be benchmarked against a hidden test dataset.

Additionally, we will explore the possibility of predicting all three attributes simultaneously and visualizing learned representations or embeddings to identify similar items that cluster together. The dataset contains textual and image data stored in Parquet files and tfrecords formats, respectively. The proposed architecture utilizes bidirectional layers, a concatenation layer, and dense layers for multi-output classification. The results showed promising performance for all three predictions.

II. Related Work

Product classification in e-commerce using deep learning has been a subject of interest in recent years, with various studies proposing new methods and techniques to improve the performance of these systems.

Zahavy et al. [1] proposed a fusion approach utilizing a multi-modal architecture for product classification in e-commerce. The authors reported that this approach, which combines image and text classification results, outperforms when image or text classification results are used separately.

Verma et al. [2] investigated the domain adaptability of text and image-based architectures and found that a Deep Multi-Modal, Multi-Level Fusion framework that uses both modalities simultaneously enhances the overall performance of the classification system.

Zhao et al. [3] proposed a method to enhance product identification and classification accuracy by improving the dense connection block. To accomplish this, the authors created a fusion feature pyramid convolution network to define the feature graph of the DenseNet layers and then employed a Dual-Path Feature Fusion (DPFM) module to fuse the feature maps of the three layers.

Yu et al. [4] proposed a product categorization solution using different classification models, including Fasttext [5], Text-CNN [6], Text-RNN [7], VDCNN [8], and bi-directional LSTM modules [9], for both single and multi-label prediction. The authors used a simple voting and weight voting method to fuse the different classification models, yielding satisfactory results in terms of accuracy, recall, and F1 score.

E-commerce product description methodologies present distinct challenges apart from product classification. Zhu and colleagues [10] have introduced a multi-modal pretraining methodology that tackles the modality-missing and modality-noise problems, which are common issues in product-based application tasks. Overall, the use of deep-learning models in product classification has resulted in some of the best models. Hence, deep learning models have potential in the domain of product classification.

III. Methodology

A. Data Analysis

In this study, a total of 245,485 training data points and 27,119 testing data points were used. The data was stored in two different formats, parquet files for text data and tfrecords for image data.

The parquet data consisted of both textual and categorical data. For categorical data, one-hot encoding was used to transform the data. For textual data, tokenization was applied and an embedding layer by keras was used to convert the text into a numeric format. The bottom category of the data contained more than 2700 classes, which made it difficult to predict accurately. The accuracy achieved for this category was less than 1%. It was observed that all the classes in the bottom category were within the range of 40 to 90.

To improve the quality of the data, columns with more than 50% of NaN values were dropped from the dataset. In the categorical section, only 1 column is taken into consideration as seen in the table below.

Overall, the data analysis showed that the dataset was complex and required preprocessing steps to achieve better

Categorical Column	No. of NaN Values (Out Of 245,485)	Percent NaN Values
Room	236758	96.4%
Craft Type	212965	86.7%
Recipient	231732	94.4%
material	224609	91.5%
occasion	192256	78.3%
holiday	204466	84.3%
art_subject	242712	98.8%
style	228453	93.1%
shape	243127	99%
pattern	234807	95.6%

Fig. 1. Percent of NaN values

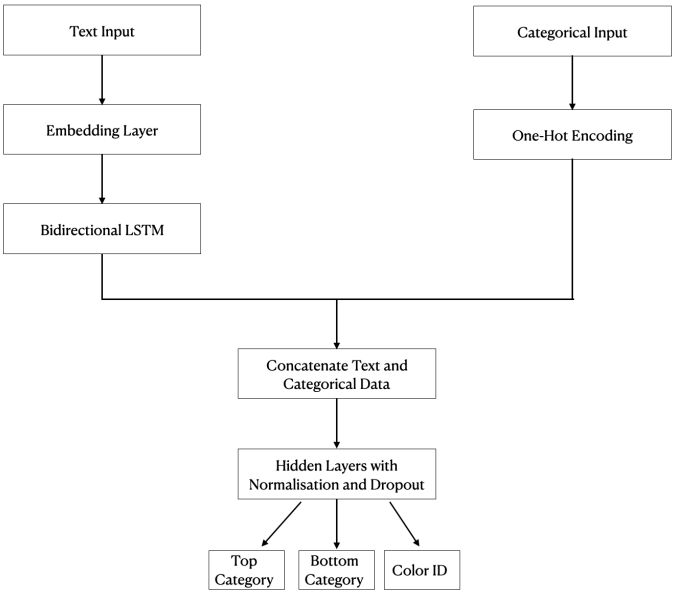

Fig. 2. Different steps in the architecture

results. The use of different data formats and transformation techniques allowed for the inclusion of both image and textual data, which increased the complexity of the analysis.

B. Data Pre-processing

In this study, data pre-processing played an essential role in preparing the data for the deep learning model. The first step was removing columns with more than 50% missing data. This was done to reduce the impact of missing values on the model's accuracy and to reduce the dimensions of the data. As a result of this step, some features were dropped from the dataset.

After the removal of missing data, categorical data were one-hot encoded. This technique was used because deep learning models cannot handle categorical data directly. One-hot encoding is a process that converts categorical data into binary values. Each category is represented as a binary vector where all values are zero except for the index representing the category, which is set to one. This process increases the number of features but ensures the model can handle categorical data.

Textual data is pre-processed using tokenization and embedding. Tokenization is the process of splitting the text into individual words or tokens. This process was done using the Keras tokenizer. After tokenizing the data, it is passed through the embedding and bidirectional layers.

Overall, the pre-processing steps were critical in ensuring the data was ready for the deep learning model. The removal of missing data, one-hot encoding of categorical data, and tokenization and embedding of textual data helped ensure that the model could handle the different data types in the dataset.

C. Architecture

The steps of the architecture can be seen in Fig. 2. The proposed model is designed to perform multi-output classification on a textual dataset. The architecture consists of five main parts: an embedding layer, three bidirectional layers, a concatenation layer, two dense layers with batch normalization and dropout, and three output layers.

The embedding and bidirectional layers are critical components in the given architecture as they enable the model to learn efficiently from categorical and sequential data. They allow the model to capture important patterns and dependencies in the data, leading to more accurate predictions.

Next, the output of the bidirectional layers is concatenated into a single vector. The concatenation layer enables the model to merge the information learned from the three inputs into a single representation.

The model's output consists of three branches, each with a different number of neurons. The first branch outputs a probability distribution over 15 classes(top category), the second outputs a probability distribution over 2782 classes(bottom category), and the third outputs a probability distribution over 20 classes(color id). In summary (Fig. 3), the proposed architecture is a deep learning model that combines bidirectional layers, concatenation layer, and dense layers to learn representations from different types of inputs and classify them into multiple classes. The model has three output layers, each with a different number of neurons, to perform multi-output classification.

D. Findings And Experiments

The analysis was conducted on a dataset containing 245,485 training data points and 27,119 testing data points, where Parquet files contain textual and categorical data and tfrecords contain image data. For categorical data, one-hot encoding was used, and for textual data, tokenisation followed by an embedding layer by Keras was used.

Initially, a single-layered deep learning model was trained to predict the top and bottom categories. The training accuracy for the top category was around 70%, and for the bottom

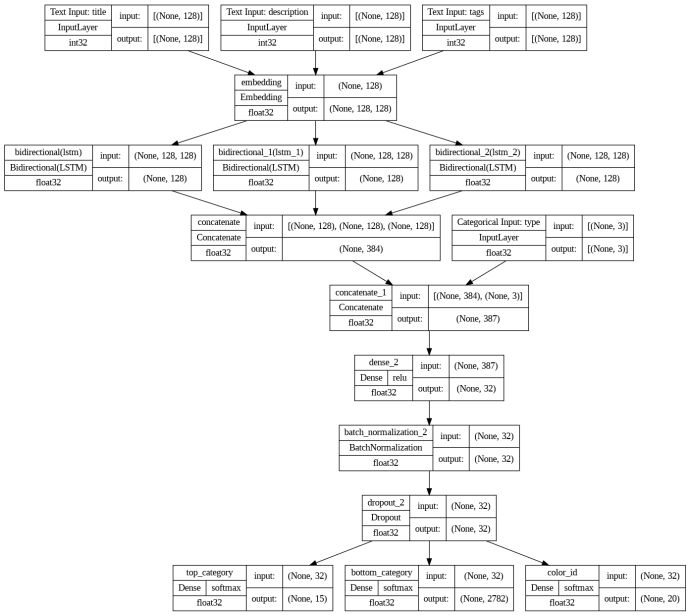

Fig. 3. Detailed Architecture

category was less than 1%. It was observed that the accuracy of the bottom category was improved significantly just by adding the color id to the output. The training accuracy improved to 45% for the bottom category and 85% for the top category. The loss of color id might have helped to learn the bottom category better.

Further, one more layer was added to the model, and different dropout and batch size values were tested. The best results were obtained for 0.2 drop out after the first layer and 0.4 after the second layer. The final training accuracy for the top category was 92%, for the bottom id it was 60%, and for color id it was 65%. These results suggest that the added layer and optimal dropout and batch size values have significantly improved the model's performance in predicting the top, bottom, and color categories.

However, it is essential to note that the bottom category contains 2700+ classes, and it is a complex task to predict them with high accuracy. Moreover, all the classes of the bottom category lie within the range of 40 to 100. Additionally, some columns were dropped that had more than 50% of NaN values, and it was observed that it significantly affected the training and testing of the model.

IV. Results

A. Parquet Files:

- The first model tested was a double-layer neural net with a 1D convolutional layer (instead of LSTM Bidirectional layer) with 128 filters and a kernel size of 5, followed by a max pooling layer to reduce the dimensionality. The model achieved an F1 score of 0.853 for the top category, 0.446 for the bottom category, and 0.357 for the color ID. These scores indicate that the model performed relatively well in predicting the top category but needed help predicting the bottom category and color ID.

Predicting Category	Validation F1 Score
Top Category	0.892
Bottom Category	0.509
Color ID	0.480

Fig. 4. Results

- The second model tested was a double-layer neural net with Bidirectional LSTM. After dropping all the categorical columns, the model was trained using only the "type" column. The Bidirectional LSTM layer captured the dependencies between words in the text and achieved higher scores than the first model. The validation F1 score for the top category was 0.892, for the bottom category was 0.509, and for the color ID was 0.480 as shown in Fig. 4. These results show that the model performed better than the first model in predicting all categories, with the top category having the highest F1 score.

B. TFRecords(images):

- The RESNET50 model was used to train the dataset to predict the color ID. The model achieved an F1 score of 0.4586 for the color ID, with a validation F1 score of 0.3665. These scores indicate that the model could make predictions with moderate accuracy, but there is still room for improvement. Overall, the results of this experiment suggest that further refinement and optimization of the model are necessary to improve its accuracy.

V. Conclusion

In this study, we presented a deep learning architecture for multi-output classification on a complex image and textual data set. The proposed architecture achieved promising results, achieving a validation F1 score of 0.892 for the top category, 0.509 for the bottom category, and 0.480 for the color ID category. The analysis showed that the data set was complex and required preprocessing steps to achieve better results. The use of different data formats and transformation techniques allowed for the inclusion of both image and textual data, which increased the complexity of the analysis. The final result of parquet data (text data) is far better than that of TFRecords data (image data). The proposed architecture combines bidirectional layers, a concatenation layer, and dense layers to learn representations from different types of inputs and classify them into multiple classes. Overall, the proposed architecture can be used to perform multi-output classification on complex datasets containing textual data.

Hardware was the biggest limitation while training the model, which became one of the future works. Other future work includes training on better-performing models for tfrecords image data other than RESNET50. Moreover, adding convolution layers to the text data model and using XGBoost or Random forest feature importance to extract the important features with tuned hyperparameters to achieve better results.

REFERENCES

[1] T. Zahavy, A. Magnani, A. Krishnan, and S. Mannor, "Is a picture worth a thousand words? a deep multi-modal fusion architecture for product classification in e-commerce," *arXiv preprint arXiv:1611.09534*, 2016.

[2] E. Daskalakis, K. Remoundou, N. Peppes, T. Alexakis, K. Demestichas, E. Adamopoulou, and E. Sykas, "Applications of fusion techniques in e-commerce environments: A literature review," *Sensors*, vol. 22, no. 11, p. 3998, 2022.

[3] B. Zhao, W. Li, Q. Guo, R. Song *et al.*, "E-commerce picture text recognition information system based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[4] W. Yu, Z. Sun, H. Liu, Z. Li, and Z. Zheng, "Multi-level deep learning based e-commerce product categorization." in *eCOM@ SIGIR*, 2018.

[5] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[6] I. Alshubaily and X. Zhang, "Textcnn with attention for text classification.. 10.48550," *arXiv preprint arXiv.2108.01921*, 2021.

[7] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[8] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[9] A. Mousa and B. Schuller, "Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis," 2017.

[10] Y. Zhu, H. Zhao, W. Zhang, G. Ye, H. Chen, N. Zhang, and H. Chen, "Knowledge perceived multi-modal pretraining in e-commerce," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2744–2752.