Enhancing E-commerce Product Classification with BERT: Leveraging Amazon Dataset for Precise Categorization

Akansha Malviya, Maral Amiri, Sakshi Mukkirwar, Yamini Muthyala
San José State University
Department of Applied Data Science

***Abstract*—Product categorization is critical in e-commerce, enhancing operations, optimizing inventory management, and enabling data-driven decisions. However, challenges persist due to complex product descriptions, inconsistent annotations, and inaccurate labels, often leading to suboptimal classification results. This project advances product categorization by employing BERT, a sophisticated language model, fine-tuned on Amazon's dataset to tackle these issues as a multi-label classification challenge. By encoding product titles and descriptions into a comprehensive semantic framework and experimenting with various hyperparameter tuning strategies, we have tailored BERT to better capture the unique attributes and specialized knowledge inherent in e-commerce products, significantly improving categorization accuracy.**

***Keywords*— E-commerce, BERT, Categorization, Customer satisfaction, Product classification, Text classification.**

I. Motivation

In the rapidly evolving world of e-commerce landscape, accurately categorizing products is essential, especially for small businesses aiming to rival larger competitors. Misclassified products disrupt the shopping experience and hinder sales opportunities. Traditional methods of product classification often struggle due to the diverse range of items and inconsistencies in how they are described across various platforms.

Our drive stems from a desire to completely transform the way products are classified, providing businesses of various scales with the means to optimize their operations, enhance customer satisfaction, and explore new opportunities for expansion. By utilizing BERT, an advanced natural language processing model, we seek to create a system capable of accurately classifying products despite inherent difficulties. Our aim is to offer a comprehensive understanding of customer preferences, facilitate personalized recommendations.

II. Background

The task of product classification is complicated by noisy labels and inconsistent product annotations across different websites. Conventional approaches, such as rule-based systems and conventional machine learning algorithms, often struggle to capture the nuances and semantics embedded in product descriptions, leading to sub-optimal classification performance.

Recent advancements in natural language processing, particularly with transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), have opened new possibilities for tackling the product classification problem. BERT has achieved state-of-the-art results in various text classification tasks due to its ability to learn contextualized word representations by pre-training on large-scale corpora. Leveraging BERT for e-commerce product classification allows for a deeper understanding of the semantic information in product metadata, such as names and descriptions. By treating product classification as a multi-label problem and fine-tuning BERT, we can effectively handle the complexity and diversity of product data, leading to more accurate and robust classification

outcomes. Through extensive experimentation and hyperparameter tuning, we aim to demonstrate the potential of BERT in advancing e-commerce technology by significantly improving product classification accuracy.

III. Literature Survey

Recent research has advanced machine learning models for e-commerce product categorization. One approach automates the validation of textual attribute values using stochastic neural networks with Transformer-based encoder-decoder architectures optimized via ELBO [1]. Another explores multi-level and multi-class deep learning tree methods, utilizing models like FastText, TextCNN, TextRNN, VDCNN, and hierarchical search trees for label prediction [2]. Additionally, BERT-based models for multi-label product classification leverage pre-trained BERT models with fully-connected layers and binary cross-entropy loss [3], while ensemble BERT models with dynamic masked softmax and pseudo-labeling address hierarchical classification challenges [4].

Efforts to enhance BERT for e-commerce have included incorporating phrase-level and product-level knowledge through techniques like Adaptive Hybrid Masking (AHM) and Neighbor Product Reconstruction (NPR) [5]. BERT has also been pre-trained on large unlabeled text corpora, enabling its application to natural language understanding tasks in e-commerce [6]. Complementary approaches integrate consumer-centric analysis and psychological factors into traditional machine learning models like CNNs for product classification [7]. Our project builds on these efforts by utilizing a BERT-based model fine-tuned for e-commerce product classification. Extensive hyperparameter tuning and fine-tuning of BERT enable our model to accurately represent product metadata, handle diverse categories, and manage complex, noisy data, ensuring a robust and thorough understanding of product descriptions.

Vandic's [8] hierarchical product classification (HPC) framework achieved average accuracy of 76.80 % and also a maximum accuracy of 83.52% on the top level categories. In contrast, our BERT-based approach achieved an accuracy of 85.36 %, demonstrating superior performance.

IV. Methodology

The methodology for this research incorporates comprehensive techniques for data collection, preprocessing, modeling, and training. This section elaborates on the methods employed to address the challenge of categorizing e-commerce products based on textual attributes. Figure 1 shows the methodology for our project.

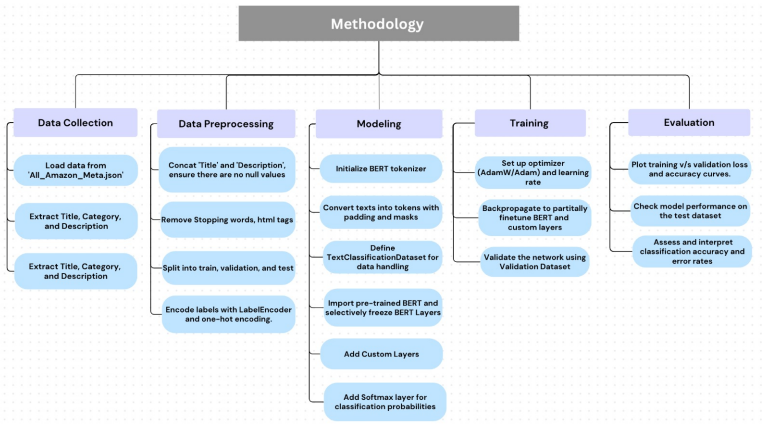

Fig. 1. Methodology

A. Data Collection

The very initial stage of our research was gathering information from the 'All Amazon Meta.json' file, which is a large file containing detailed product metadata from Amazon. Important features like ASIN, Title, Category, and Description are included in this dataset and are crucial to our study. To manage the massive volume of data efficiently, the data was processed in chunks. Figure 2 is the sample of the metadata.

B. Data Preprocessing

Data preprocessing involved several key steps to prepare the dataset for the modeling phase:

- *Text Concatenation*: The 'Title' and 'Description' fields were merged to form a unified text field for each product and ensured that there are no null values. This approach is aligned with techniques used in deep learning where comprehensive text data improves model performance.

- *Text Cleaning*: Non-relevant characters, HTML tags, and stop words were removed from the text. This cleaning process ensures that the models focus on meaningful words, improving

Sample metadata:


```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "feature": ["Botiquecutie Trademark exclusive Brand",
              "Hot Pink Layered Zebra Print Tutu",
              "Fits girls up to a size 4T",
              "Hand wash / Line Dry",
              "Includes a Botiquecutie TM Exclusive hair flower
bow"],
  "description": "This tutu is great for dress up play for your
little ballerina. Botiquecute Trade Mark exclusive brand. Hot Pink
Zebra print tutu.",
  "price": 3.17,
  "imageURL": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "imageURLHighRes": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL.jpg",
  "also_buy": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O",
"0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S", "0000031895",
"B003AVKOP2", "B003AVEU6G", "B003IEDM9Q", "B002R0FA24", "B00D23MC6W",
"B00D2K0PA0", "B00538F5OK", "B00CEV86I6", "B002R0FABA", "B00D10CLVW",
"B003AVNY6I", "B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C",
"B008UBQZKU", "B00D103F8U", "B007R2RM8W"],
  "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y",
"B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E", "B003AVKOP2",
"B00D9C1WBM", "B00CEV8366", "B00CEUX0D8", "B0079ME3KU", "B00CEUWY8K",
"B004FOEEHC", "0000031895", "B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2",
"B00EM7KAG6", "B00AMQ17JA", "B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y",
"B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ", "B00538F5OK",
"B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U", "B00CEUWUZC", "B00IJVASUE",
"B00GOR07RE", "B00J2GTM0W", "B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G",
"B008VV8NSQ", "B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S",
"B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M", "B00EHAGZNA",
"B0046W9T8C", "B00E79VW6Q", "B00D10CLVW", "B00B0AVO54", "B00E95LC8Q",
"B00GOR92SO", "B007ZN5Y56", "B00AL2569W", "B00B608000", "B008F0SMUC",
"B00BFXLZ8M"],
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports", "Dance"]]
}
```


Fig. 2. Sample Metadata

the accuracy of categorization.

- *Data Encoding and Splitting*: The categorical labels were encoded using LabelEncoder and converted to one-hot vectors to facilitate machine learning processes. The dataset was then split into training, validation, and test sets to ensure a robust evaluation of the model.

- *Category Selection*: From the initial broad set of categories, we narrowed down our focus to 16 specific categories, as shown in figure 3. These categories now serve as our target labels for the classification model.

The size of the data was significantly reduced from 115 GB to 23.5 MB and 79 MB for different parts of the dataset, making computational processing more feasible while retaining essential information. The reduction involved filtering unnecessary columns and balancing the number of entries across selected categories to prevent model bias toward any single category.

C. Modeling

The modeling phase in our study utilizes the BERT tokenizer, which employs word piece tokenization to efficiently handle subword units, allowing for more nuanced understanding and processing of text data. This tokenizer breaks down words into manageable tokens, which are then transformed into token embeddings. These comprehensive embeddings are then input into BERT's deep neural network architecture.

```
[2]: df.Category.value_counts()
     Last executed at 2024-05-08 18:21:13 in 14ms
```

```
[2]: Appliances                  9832
     Sports & Outdoors           9786
     Clothing, Shoes & Jewelry   9773
     Grocery & Gourmet Food      9770
     Musical Instruments         9742
     Arts, Crafts & Sewing       9735
     Patio, Lawn & Garden        9698
     Toys & Games                9696
     Tools & Home Improvement    9680
     Home & Kitchen              9673
     Office Products             9639
     Pet Supplies                9637
     Automotive                  9613
     Electronics                 9450
     Cell Phones & Accessories   9409
     Books                       6240
```

Fig. 3. Categories

This design is grounded in the foundational work by [6], who originally developed BERT and detailed its architecture and capabilities for processing complex language patterns. The output from BERT's final layer, specifically the 1x768 dimensional vector associated with the CLS (classification) token, is then used as an input to a downstream deep neural network (DNN).

The use of BERT tokenizer is inspired by its success in various Natural Language Processing tasks as discussed by [3], who fine-tuned BERT for product data classification.

Our custom model architecture builds upon the foundational BERT model, which is pre-trained on a large corpus of text and then adapted for our specific e-commerce categorization task:

- *Custom Layers*: Following the architecture proposed by [5], we added custom layers to the pre-trained BERT model, batch normalization for faster convergence, and additional dense layers to tailor the model to our classification needs. To further optimize training, we implemented a learning rate scheduler. This scheduler adjusts the learning rate dynamically throughout the training process, which helps in decreasing the learning rate as the model

approaches convergence.

- *Output Layer*: A softmax layer was employed to convert the logits from the final dense layer into probabilities, following the principles discussed by Dunne and Campbell [9]. These scores represent the model's confidence levels across the various categories, effectively allowing the model to make categorical distinctions by assigning higher probabilities to the most likely category for each input sample.

  Softmax equation is given by the equation 1.

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1}$$

  Where:

  - z_i represents the logits (input values) for class i,
  - e is the base of the natural logarithm,
  - K is the total number of classes in the classification problem,
  - $\sum_{j=1}^{K} e^{z_j}$ is the sum of the exponentials of all the logits in the vector.

D. Training

The model was trained using the AdamW optimizer, a choice driven by its ability to handle sparse gradients on noisy problems like text classification as suggested by [2]. We specifically experimented with two distinct methods of handling the BERT layers, which are essential for understanding how the pre-trained model adapts to our specific e-commerce dataset.

- Freezing All Layers of BERT: In this method, all layers of the pre-trained BERT model were frozen, using it solely as a feature extractor without updating its weights during training. This approach speeds up training and reduces computational demands by limiting trainable parameters to just the custom layers we added.
- Not Freezing Any Layers of BERT: The second approach involved not freezing any layers of the BERT model, allowing all layers to update their weights during training. This method is expected to enhance the model's ability to fine-tune the pre-trained embeddings specifically for our task of product categorization.

During the training process, we utilized cross entropy loss as our primary loss function, aiming to minimize it to improve the model's accuracy in classifying product categories, as discussed in the context of multi-label learning algorithms by [10].

Cross Entropy Loss is given by the equation 2.

$$L = -\sum_{i=1}^{C} y_i \log(p_i) \tag{2}$$

Validation was periodically performed to tune hyperparameters and prevent overfitting, ensuring the model generalized well on unseen data.

E. Testing and Evaluation

In the testing phase, the model's performance was critically evaluated by plotting training versus validation loss and accuracy curves. This step is crucial to assess the effectiveness of the model in real-world scenarios, ensuring that it performs well across diverse sets of product data.

Test accuracy was specifically measured to quantify the model's performance. This metric provides a direct indicator of how well the model predicts the correct categories against the actual labels in the test dataset.

This comprehensive methodology leveraging advanced machine learning techniques and deep learning models ensures a robust approach to automatically categorize e-commerce products based on textual attributes, facilitating more effective product management and recommendation systems.

V. Experiments and Results

In this study, various hyperparameter tuning strategies were experimented to optimize the performance of BERT model.Firstly, experimentation with different learning rates, numbers of epochs, and the inclusion or exclusion of dropout layers is performed to identify the optimal combination of hyper parameters to achieve the highest accuracy on our validation dataset. Then through systematic testing, it is found that the choice of dataset size and batch size significantly influenced the model's performance.

The first experiment gave accuracy of 83.47 percent for included a batch size of 32, a learning rate of le-5, and a learning scheduler with a factor of 0.8. Adam optimizer for efficient gradient descent and

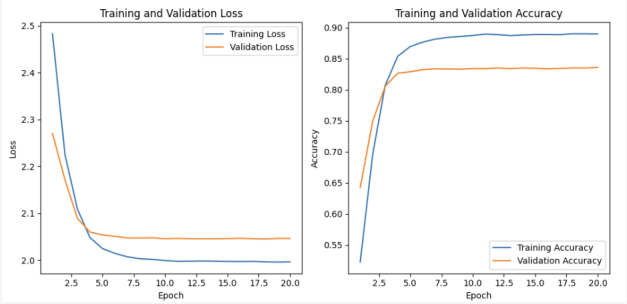

Fig. 4. Experiment 1

```
Epoch 20/20, Average Training Loss: 1.996705814611384
Time taken for epoch 20: 272.78 seconds
Epoch 20/20, Average Validation Loss: 2.04652243984051
Test Accuracy: 0.8347
```

Fig. 5. Experiment 1 Accuracy

employed Leaky ReLU and Softmax as activation functions.The results is shown in figure 4 and 5.

The second experiment gave accuracy of 75.57 percent for included a batch size of 64, a learning rate of le-5, and a learning scheduler with a factor of 0.8. Adam optimizer for efficient gradient descent and employed Leaky ReLU and Softmax as activation functions.This experiment implies keeping the batch size to 32 gives better results as shown in figure 6 and 7.

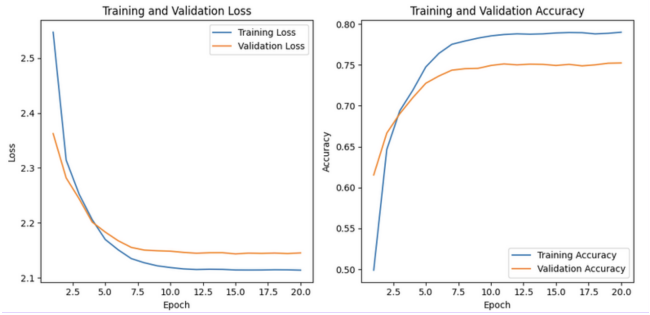

Fig. 6. Experiment 2

```
Epoch 20/20, Average Training Loss: 2.113920298979859
Time taken for epoch 20: 268.15 seconds
Epoch 20/20, Average Validation Loss: 2.1452926844157587
Test Accuracy: 0.7557
```

Fig. 7. Experiment 2 Accuracy

The third experiment performed by changing the size of dataset from 10k to 3k gave us accuracy of 83.47, which implies that adding more data to the model gave better accuracy.The results is shown in figure 8 and 9.

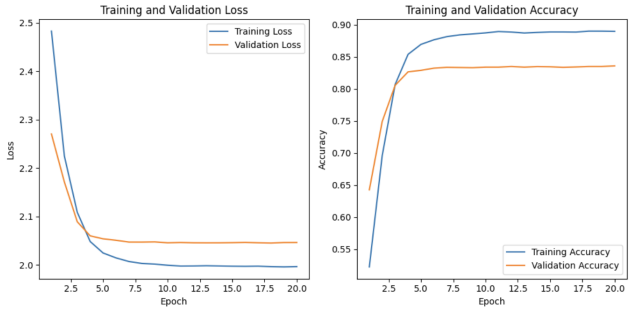

Fig. 8. Experiment 3

```
Epoch 20/20, Average Training Loss: 1.996705814611384
Time taken for epoch 20: 272.78 seconds
Epoch 20/20, Average Validation Loss: 2.04652243984051
Test Accuracy: 0.8347
```

Fig. 9. Experiment 3 Accuracy

After extensive experimentation, the results shows the configuration that achieved the best validation accuracy of 85.36 percent included a batch size of 32, a learning rate of le-5, and a learning scheduler with a factor of 0.65.Adam optimizer for efficient gradient descent and employed Leaky ReLU and Softmax as activation functions.After so many experimentation it has been discovered that excluding dropout layers led to better performance, which is contrary to the common practice of using dropout to prevent overfitting. Additionally, using Cross Entropy Loss as the loss function and trained the model for 20 epochs with batch normalization applied.The training and validation loss along with training and validation result of best model is shown in figure 10 and 11 .

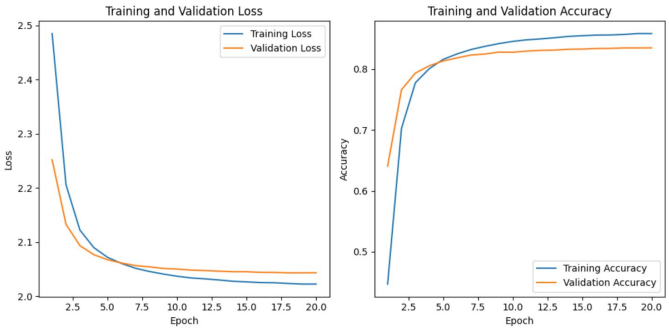

Fig. 10. Best model

```
Epoch 20/20, Average Training Loss: 1.9860749946196299
Epoch 20/20, Average Validation Loss: 2.017783838762364
Test Accuracy: 0.8536
```

Fig. 11. Best model accuracy

TABLE I
SUMMARY OF THE RESULTS

Exp.	Batch	Scheduler	Dataset	Accuracy
1	32	0.8	10,000	83.47%
2	64	0.8	10,000	75.57%
3	32	0.8	3,000	83.47%
4	32	0.65	10,000	85.36%

The results show how important it is to carefully adjust the settings (hyperparameters) in deep learning models, the summary of results is shown in Table I. Not using dropout layers, along with a small learning rate and a learning rate scheduler, helped the model learn steadily and adjust its weights properly. Adding batch normalization made the model more stable, which helped achieve high accuracy.These findings suggest that while dropout layers can be useful, their effectiveness can depend on the specific task and dataset. This means it's crucial to customize hyperparameters for each unique situation to get the best results.

VI. Future Improvements and Discussion

In the future, new methods that use images, descriptions, and titles to categorize products on Amazon can be explored. Focus can be on developing "late fusion models," which combine different types of information at the decision stage. By doing this, the strengths of both visual (images) and textual (descriptions and titles) data to improve the accuracy and reliability of product categorization can be used.

Recent studies suggest that using multiple sources of data can significantly enhance how well models perform in classification tasks. By combining both visual and textual information, late fusion models can better handle unclear or complex product attributes. This approach not only aims to make our current model more accurate but also lays the groundwork for creating smarter and more context-aware e-commerce solutions.

Overall, the BERT model with all layers frozen, when used for product classification based on text, has demonstrated superior accuracy. It achieved an accuracy of 85.36 percent, which surpasses the performance of the hierarchical product classification (HPC) framework developed by [8], which reported an average accuracy of 76.80 percent and a maximum accuracy of 83.52 percent on the top-level categories.

References

[1] Y. Wang, Y. E. Xu, X. Li, X. L. Dong, and J. Gao, "Automatic validation of textual attribute values in e-commerce catalog by learning with limited labeled data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2533–2541.

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

[3] H. M. Zahera and M. A. Sherif, "Probert: Product data classification with fine-tuning bert model," in *MWPD@ISWC*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222304838

[4] L. Yang, E. Shijia, S. Xu, and Y. Xiang, "Bert with dynamic masked softmax and pseudo labeling for hierarchical product classification." in *MWPD@ ISWC*, 2020.

[5] D. Zhang, Z. Yuan, Y. Liu, F. Zhuang, and H. Xiong, "E-bert: Adapting bert to e-commerce with adaptive hybrid masking and neighbor product reconstruction," 2009.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] F. Sun, D.-B. Luh, Y. Zhao, and Y. Sun, "Product classification with the motivation of target consumers by deep learning," *IEEE Access*, vol. 10, pp. 62 258–62 267, 2022.

[8] D. Vandic, F. Frasincar, and U. Kaymak, "A framework for product description classification in e-commerce," *J. Web Eng.*, vol. 17, no. 1–2, p. 1–27, mar 2018.

[9] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, vol. 181. Citeseer, 1997, p. 185.

[10] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.