A Hierarchical Approach to E-commerce Product Classification Using FastText

ZHENGRI CUI
School of Computing
Dublin City University
Email: zhengri.cui2@mail.dcu.ie

***Abstract*—This study introduces a hierarchical classification framework for e-commerce product categorization using descriptive textual data from the Etsy platform. FastText, a model proposed by Facebook, is employed to predict 15 top-level product categories. Building on these results, a custom architecture is designed to classify 2,609 bottom-level categories. To address the performance limitations of conventional flat classification methods (61.8% F1-score) in large-scale, fine-grained tasks, The proposed hierarchical approach incorporates multiple intermediate classification layers, which are carefully constructed through feature engineering to progressively pass information from higher to lower levels. This strategy significantly enhances bottom-level performance to an F1-score of 87.3%.**

I. Introduction

Nowdays, product category classification in e-commerce platforms significantly influences search efficiency, recommendation system development, and conversion rates. However, as product listings and data volumes increase, the complexity of classification tasks also grows. This project addresses the product classification challenge using real world data provided by the Etsy platform. The dataset includes descriptive text about products, including various product descriptions, titles, tags and so on. The training dataset additionally contains `top_category_text`, `top_category_id`, `bottom_category_text`, and `bottom_category_id`. The ultimate goal is to predict `top_category_id` and `bottom_category_id` for products in the test set based on descriptive text data. The main challenges faced in this project include:

1) The vast number of products and extensive textual data in product descriptions render training with conventional natural language processing models both time- and resource-intensive.
2) Product categories exhibit severe imbalance, with disproportionate representation across classes. For example, among the 15 top-level categories:
   - **`home_and_living`**: 54,600 products, which is 23.8% of dataset *(most represented category)*
   - **`pet_supplies`**: 5,744 products, which is 2.5% of dataset *(least represented category)*
3) Category counts grow exponentially across hierarchy levels.
   - Top-level: 15 categories
   - Bottom-level: 2,609 categories
4) The absence of middle-level category IDs. When a product belongs to a specific `bottom_category`, information from the intermediate levels is necessary to accurately determine the product's category path.

To address these challenges, I completed the project based on the fast and powerful FastText model. I expanded the intermediate levels through feature engineering using `bottom_category_text`, built models for each hierarchical level, and designed an architecture that links the model from the top-level category to the bottom-level category.

II. Related Work

Category classification using textual data has been actively explored in both academia and industry, with applications spanning diverse areas such as content management and product organization.

This study [1] classifies Vietnamese documents into 15 categories using TF-IDF, SVD, and FastText at three different levels. Among various machine learning models tested, Random Forest combined with FastText achieved the best performance with an 82% success rate in accuracy, precision, and F1 score. In study [2], a CNN-based HFT-CNN model, which utilizes a hierarchical category structure (HFS), was proposed to address the multi-label classification problem for short texts. This approach effectively propagates the abundant data from higher levels to lower levels, mitigating the data sparsity issue and contributing to improved classification performance.

The study proposed by Amazon introduces a dual-expert classification system that leverages large language models (LLMs), combining domain-specific knowledge with general knowledge to enhance classification performance [3]. According to a paper published by researchers from the e-commerce platform Jingdong, the team conducted product category classification based on text using models such as FastText, Text-CNN, Text-RNN, VDCNN, and AbLSTM. The results show that FastText and AbLSTM outperformed the other models. Additionally, the study introduces an architecture that transfers information from higher-level categories to lower-level categories by combining these models with a tree search method [4].

III. Methodology and Experiments

The ultimate objective of this study is to predict the category ID. However, since both bottom and top categories are associated with corresponding text for each ID, the model was first trained to predict the category text. Based on the predicted category, the final ID was then derived using a predefined key-value mapping. The whole project was conducted in a Colab environment utilizing high RAM, with 229,624 training samples provided by Etsy. After completing all feature engineering steps, the data was split into training and validation sets at an 80:20 ratio, and the final model was tested. The metrics provided in this report are based on the results from the validation set. Additionally, based on this final model, predictions were made on the 25,514 samples provided by Etsy.

The detailed workflow is as follows:

A. Data Exploration

The original training dataset includes the following product-related text columns: `title`, `description`, `tags`, `type`, `room`, `craft_type`, `recipient`, `material`, `occasion`, `holiday`, `art_subject`, `style`, and `pattern`. The target-related columns are `top_category_text`, `top_category_id`, `bottom_category_text`, and `bottom_category_id`.

Upon exploration, the `top_category` consists of 15 distinct classes, with their distribution shown in Figure 1. The `bottom_category` contains a total of 2,609 classes, with the number of products per class ranging from a minimum of 42 to a maximum of 98.

Although intermediate-level categories are not explicitly provided in the training data, they can be inferred from the `bottom_category_text` by using the period (".") as a delimiter. For example, in the category `home_and_living.lighting.light_accessories`, `home_and_living` represents the top-level category (matching `top_category_text`), `lighting` indicates a mid-level category, and `light_accessories` refers to the lowest-level category.

An analysis of the data revealed that the number of periods in `bottom_category_text` ranges from 0 to 6, indicating that the category hierarchy can extend up to 7 levels deep.

B. Text Preprocessing

According to the study in [4], text preprocessing techniques such as stop word removal, stemming, and noun extraction were attempted. However, the FastText model showed better accuracy when the original text was preserved. Based on this observation, the experiment focused not on modifying the original text, but rather on identifying which combination of text columns yields the best performance.

In the first experiment, the Fasttext model predicted the top-level category using a combined text consisting of only three columns: `title`, `description`, and `tags`. In the second experiment, all text-related columns were concatenated into a single column and used for top-category prediction.

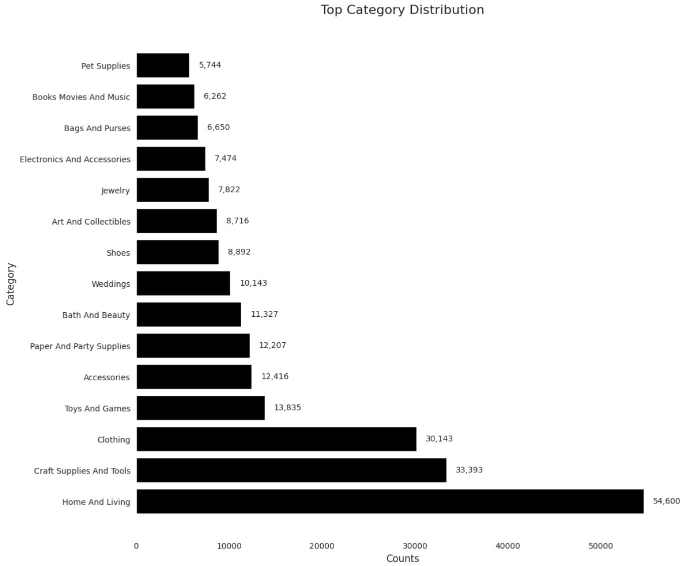

Fig. 1. Distribution of Top Categories

The results showed that, using the same dataset and default parameters, the FastText model achieved an overall precision and recall of 85.1% when using only the three columns. In contrast, combining all text columns improved the performance to 86.1%.

Based on this outcome, the following experiments were conducted using a method that combines all text columns into a single unified column. Further preprocessing was applied by removing HTML entities, URLs containing "www", special characters, and unnecessary whitespace, resulting in a normalized plain text for training.

C. Feature Engineering

The primary objective of this study is to predict both `top_category_id` and `bottom_category_id` based on given text. However, there is a significant disparity in the number of classes between these two categories. While `top_category_id` consists of only 15 classes, `bottom_category_id` includes as many as 2,609 classes, representing an extremely large and complex label space. This introduces a structural limitation, where an increase in the number of classes—particularly in the case of bottom-level categories—typically results in reduced classification accuracy under the same training conditions.

To address this issue, the textual information in `bottom_category_text` was split to construct a hierarchical structure consisting of seven levels, from Level 1 (which corresponds to `top_category_text`) to Level 7. The resulting taxonomy, which was not explicitly present in the original dataset, was systematically derived from this information. The structure of the hierarchy is illustrated in Figure 2, and the distribution of the dataset across the constructed hierarchy is presented in Figure 3.

D. Modeling

During the modeling phase, separate models were trained for each hierarchical level by adjusting hyperparameters based

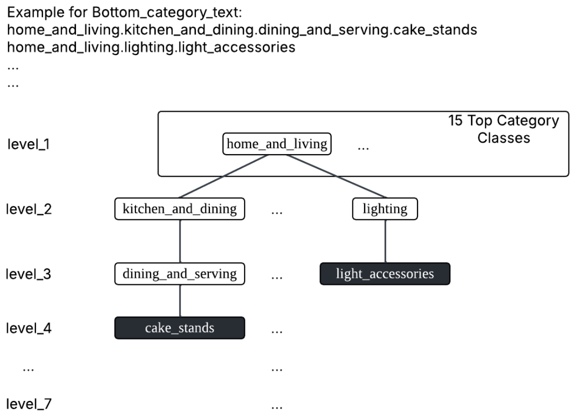

Fig. 2. The reconstructed hierarchical category structure ranges from Level 1 to Level 7. Level 1 contains 15 classes, which correspond to the top category text. The black areas represent classes that belong to both that level and the bottom category.

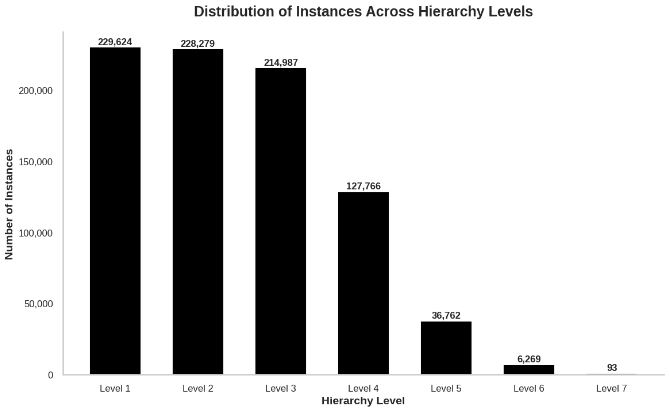

Fig. 3. Distribution of samples across the hierarchical category levels.

on level-specific datasets. The ranges of hyperparameters used throughout the experiments were as follows: learning rate ranged from 0.05 to 1.5, embedding dimension from 50 to 120, and the number of epochs from 6 up to 20. The window size was set to either 5 or 6, and the word n-gram to 2 or 3. Two types of loss functions were tested: ”ova” and ”softmax”.

1) Top Category: Predicting the top category is equivalent to predicting the Level 1 category defined in this study. Initially, the model was trained using FastText with its default settings, achieving a precision and recall of 86.1% on the validation set. After tuning the hyperparameters, the performance improved significantly. With the following configuration: lr = 1.5, dim = 70, epoch = 6, ws = 6, wordNgrams = 3, loss = ’ova’, the precision and recall increased to 89.96%. Based on the model, the embedding data set to 70 dimensions was dimensionality-reduced using PCA, and the results of visualizing the actual labels and predicted labels on the validation set are shown in Figure 4

2) Bottom Category: In the bottom class, the large number of categories leads to a sparse representation problem.

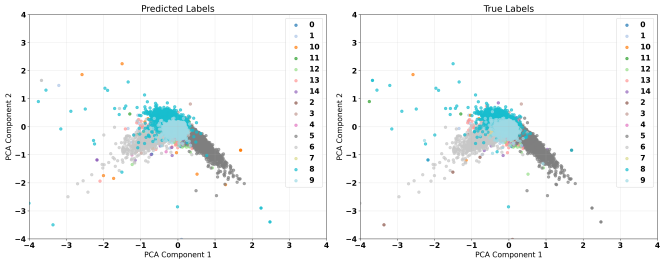

Fig. 4. Predicted Labels VS Actual Labels

To address this, incorporating prediction information from the upper-level hierarchy into the lower-level classification is identified as a key strategy. Based on this approach, the architecture shown in Fig 5 was designed, and the equation used for bottom_category_text prediction is presented below:

Input Features and Preprocessing:

Let the input features be represented as:

$$X = [X_1, X_2, \ldots, X_n] \tag{1}$$

A preprocessing function $\phi(\cdot)$ is applied to the input to produce the unified feature representation:

$$\tilde{X} = \phi(X) \tag{2}$$

Hierarchical Prediction:

Suppose there are L hierarchical levels ($\ell = 1, 2, \ldots, L$), and each level has a classifier f_ℓ.

At level ℓ, the probability of class c_ℓ is given by:

$$P_\ell(c_\ell \mid \tilde{X}) = f_\ell(\tilde{X}) \tag{3}$$

The joint probability across the full classification path is computed as:

$$P_{\text{path}}(c_1, \ldots, c_L) = \prod_{\ell=1}^{L} P_\ell(c_\ell \mid \tilde{X}) \tag{4}$$

Path Validation Condition:

Let c_L^* denote the class predicted by the bottom-level classifier. We define the path consistency check as:

$$\text{Check}(c_L^*) = \begin{cases} \text{True}, & \text{if } \forall \ell \in \{1, \ldots, L{-}1\},\ c_\ell^* \in \text{Parent}(c_{\ell+1}^*) \\ \text{False}, & \text{otherwise} \end{cases} \tag{5}$$

Here, Parent$(c_{\ell+1}^*)$ represents the parent class(es) of $c_{\ell+1}^*$ in the hierarchy.

Final Prediction Rule:

$$\text{Final Class} = \begin{cases} c_L^*, & \text{if Check}(c_L^*) = \text{True} \\ \arg\max_{c_1,\ldots,c_L} P_{\text{path}}(c_1, \ldots, c_L), & \text{otherwise} \end{cases} \tag{6}$$

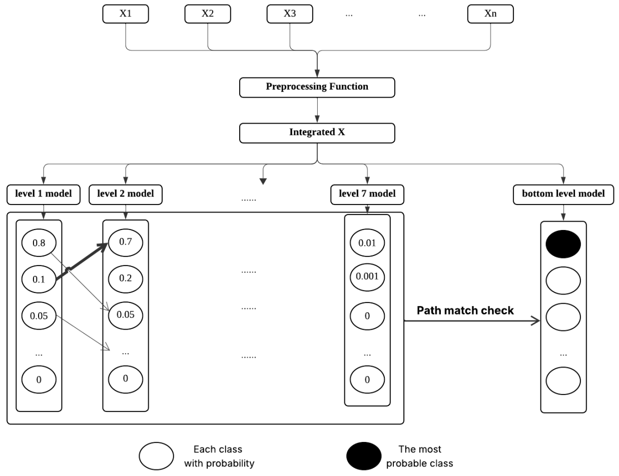

Fig. 5. Architecture of Searching Bottom Category

E. Results

The results obtained in this experiment are shown in the table I. For the top category, the adjustments refer to the comparison between the default parameter setting and the best-found parameters. For the bottom category, the adjustments refer to the application of the search architecture and the case without applying it.

TABLE I
F1 SCORE COMPARISON BY CATEGORY LEVEL

Category Level	F1 Score	
	Before Adjustment	After Adjustment
Top Category	0.861	0.899
Bottom Category	0.618	0.873

IV. CONCLUSION

Predicting categories based on product texts is not an easy task, especially when it comes to the bottom category, where the number of classes increases exponentially. However, by applying FastText hierarchically and incorporating a search architecture, we have shown that this approach is both fast and effective.

REFERENCES

[1] H. H. Luong, L. T. T. Le, and H. T. Nguyen, "An approach for web content classification with fasttext," in *Computational Data and Social Networks*, M. H. Hà, X. Zhu, and M. T. Thai, Eds. Singapore: Springer Nature Singapore, 2024, pp. 138–146.

[2] K. Shimura, J. Li, and F. Fukumoto, "HFT-CNN: Learning hierarchical category structure for multi-label short text categorization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 811–816. [Online]. Available: https://aclanthology.org/D18-1093/

[3] Z. Cheng, W. Zhang, C.-C. Chou, Y.-Y. Jau, A. Pathak, P. Gao, and U. Batur, "E-commerce product categorization with LLM-based dual-expert classification paradigm," in *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, S. Kumar, V. Balachandran, C. Y. Park, W. Shi, S. A. Hayati, Y. Tsvetkov, N. Smith, H. Hajishirzi, D. Kang, and D. Jurgens, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 294–304. [Online]. Available: https://aclanthology.org/2024.customnlp4u-1.22/

[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.