Journal of Data, Information and Management (2022) 4:71–85
https://doi.org/10.1007/s42488-022-00066-6

ORIGINAL ARTICLE

Classifying multi-level product categories using dynamic masking and transformer models

Ozan Ozyegen[1] · Hadi Jahanshahi[1] · Mucahit Cevik[1] · Beste Bulut[2] · Deniz Yigit[2] · Fahrettin F. Gonen[2] · Ayşe Başar[1]

Received: 18 October 2021 / Accepted: 31 January 2022 / Published online: 20 April 2022

Abstract
In an online shopping platform, a detailed categorization of the products greatly enhances user navigation. Online retailers also benefit from well-defined product categories as various sales and marketing operations such as special discounts and promotions can be easily done over a set of product categories. Furthermore, incorrect and subjective product categories suggested by an operator can be more easily identified thanks to an automated classification system. In this study, we investigate the task of classifying grocery product categories using product titles. We employ a wide variety of text classification models for this task, including traditional machine learning and deep learning models as well as state-of-the-art transformer models. In our analysis, we specifically focus on the generalizability of the trained classification models to the products of other online retailers, the dynamic masking of infeasible subcategories for pretrained language models, and the impact of incorporating different word embeddings. We observe that the deep learning models and the transformers significantly outperform traditional text classification methods such as XGBoost and SVM, and achieve excellent prediction performance exceeding 90% accuracy and F1-score values. We lastly explore the failure cases where a product is misclassified, and make recommendations for future studies to improve the prediction performance.

Keywords Multi-level classification · Machine learning · Supervised learning · Product category classification

1 Introduction

E-commerce platforms have become more and more popular over the years. The COVID-19 epidemic further stimulated public interest in e-commerce, resulting in a boom of e-commerce businesses (Gao et al. 2020). As a result, the e-commerce industry has become more competitive, driving firms to make considerable expenditures to improve their platforms.

E-commerce platforms organize their products into categories to make it easier for users to find what they intend to buy. Proposing the right category for a product based on its description can be difficult and time-consuming for e-businesses. The incorporation of multi-level categorization increases the complexity of this task. For example, "mega size original chips" can be classified under snacks category, salty snacks subcategory, and chips segment. As the number of items offered on an e-commerce platform grows, keeping track of a plethora of categories in different levels becomes increasingly challenging. In this regard, the goal of the product category classification models is to automate the process of determining the appropriate category for a given product using the available information such as product title and description. These models can be used for various purposes in practice.

Instead of painstakingly reviewing all categories to identify the appropriate one for each item, the retailers may utilize product category classification algorithms to automate the item classification. This can be particularly helpful when categorizing the new products that are introduced to the retail store. These models can also be used to revise the biased or repeated category levels currently existing in the system. Additionally, product classification models can be helpful in enhancing the marketing strategies and analytics, e.g., by efficiently processing the collected product information from the market place.

✉ Mucahit Cevik
mcevik@ryerson.ca

[1] Mechanical Industrial Engineering Department, Ryerson University, Toronto, Ontario, M5B 2K3, Canada

[2] Getir Perakende Lojistik A.S., Istanbul, Turkey

The product title classification problem appears to be a variant of the well-studied text classification problems at first glance. Although there are some commonalities, text length distribution, and the structure of the short titles distinguish the product category classification problem from the others (Yu et al. 2012).

Accordingly, different techniques have been developed for this problem to extract the most information from available brief textual information. Employing word-level and character-level features to obtain fine-grained subword information (Wang et al. 2017), context-relevant concept word embeddings (Xu et al. 2020), data augmentation (Rosario 2017), and word-cluster embeddings (Shen et al. 2018) are examples of such techniques.

An online store can benefit from product classification in three ways. First of all, automatic classification of items can aid customers in navigating online retail platforms. A high-quality product classification leads to a more effective and satisfying user experience. Secondly, it enables online merchants to better manage their sales and marketing activities. They can quickly introduce new products to their system and monitor aggregated data of different product categories instantaneously. Lastly, online merchants may identify and follow the products offered by other online stores. They can monitor collected data on the status of various product categories of their competitors using the recommendations of product title classification algorithms.

The problem of multi-level product category classification based on textual product information can be formally defined as follows. Let $\mathbb{X}$ be the document space comprising textual product information, and $\mathbb{C}^{\ell} = \{c_1, c_2, \ldots, c_K\}$ be the fixed set of K classes for category level $\ell \in \mathbb{L} = \{1, 2, \ldots, L\}$. A text classification model γ maps the documents (i.e., products identified by textual data) to classes (i.e., product category values), that is, $\gamma : \mathbb{X} \to \mathbb{C}^{\ell}$ for any given $\ell \in \mathbb{L}$. As such, our classification problem can be described as determining γ such that $\gamma(\mathbf{x}) \in \mathbb{C}^{\ell}$ is the product category for level ℓ that is most appropriate for $\mathbf{x} \in \mathbb{X}$. Note that this problem can also be posed as a multilabel classification task where γ classifies L category values simultaneously. Table 1 shows sample data instances for a product classification task, which contain product titles and their related category, subcategory, segment, and subsegment tags. For instance, accurate classification models would be able to classify the product with title "Lipton Yellow Label Bulk Tea" as Beverages (Category), Tea & Hot Drinks (Subcategory), Bulk Tea (Segment) and Black Tea (Subsegment).

1.1 Research goals

We investigate the problem of multilevel categorization of the grocery products based on textual product information mainly consisting of product titles.

We explore the limits of text classification methods for grocery product title classification tasks through our detailed numerical study. We examine a variety of machine learning approaches based on Natural Language Processing (NLP) techniques. The methods investigated in this study may help better classification/categorization of new items, the consolidated monitoring of other vendors' products, and the diagnosis of improperly labeled products in the system.

1.2 Contributions

We summarize the contributions of our study as follows.

- To the best of our knowledge, our work constitutes the first study on multi-level classification for predicting grocery products categories.
- We compare and contrast six different text classification algorithms in this task. We examine different techniques such as utilizing product titles and applying three dynamic masking approaches for infeasible sub-levels for transformer models.

  Different from the previous works, we leverage bi-lingual models to improve prediction performance based on Turkish and English product titles. By investigating the conventional and state-of-the-art NLP techniques, our analysis offer a robust baseline for the product title classification task.
- We conduct a detailed numerical study using six distinct datasets acquired from different online retail stores to test the generalizability of the prediction models. Furthermore, we identify the challenges involved in grocery product title classification through a detailed analysis of the model predictions. This thorough analysis showcase the capabilities and limitations of

Table 1 Sample product titles and four-level category labeling (brand names are italicized)

Product Title	Category	Subcategory	Segment	Subsegment
Lipton Yellow Label Bulk Tea	Beverages	Tea & Hot Drinks	Bulk Tea	Black Tea
Dalan Lilac 4 lt Liquid Soap	Nature's Signature	Personal Care	Hand, Body & Face Products	Liquid Soap
Vivident Storming Gum Mint Flavor 33 g	Snacks	Sweet Snacks & Candy	Chewing Gum	Mint-flavored
Lean Ground Beef, Value Pack	Meat & Poultry	Red Meat	Beef & Veal	Ground

various NLP techniques for product title classification, and contribute to further methodological and empirical work in this field.

1.3 Structure of the paper

We review the literature on hierarchical product category classification in Section 2, followed by the methodology and dataset discussion in Section 3. We report our findings for within- and cross-platform product category prediction in Section 4. This section also provides a discussion on incorrectly classified instances and underlying causes for such misclassifications. Lastly, we summarize our work, acknowledge and address the study limitations and discuss future research directions in Section 5.

2 Literature review

Hierarchical product category classification is a challenging task as it requires product instances to be accurately assigned to multiple levels of categories based on limited information such as short product titles and brand names. Over the past years, the interest in this problem has increased with the growth of online shopping and the availability of large datasets.

Yu et al. (2012) provided one of the first studies on this research problem. They conducted an extensive numerical study to illustrate how linear SVMs could be used for large-scale multi-class title classification, and identified the differences between product title classification and generic text classification problems. They used a dataset from a large internet company, which contains 29 classes.

They also compared the effectiveness of different feature representations. Their numerical study showed that stemming and stop word removal are harmful, and bigrams are effective for the product title classification task.

There have been significant advances in NLP models over the past decade. For word representations, methods such as Glove (Pennington et al. 2014) and Word2Vec (Mikolov et al. 2013) became increasingly popular. More recently, advanced NLP techniques such as BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019) and XLM (Ma et al. 2020) have been shown to achieve state-of-art performance for many language tasks. These models, also known as pretrained language models (PTMs), are used slightly differently compared to the previous machine learning-based NLP models. They are first trained on large-scale unlabeled corpora to leverage a good understanding of natural language. Then, depending on the task, a few layers are attached to the end of the "pretrained" base model. Afterwards, the full network is fine-tuned end-to-end on a smaller task-specific corpus. There are multiple advantages of using PTMs over the traditional methods. Firstly, the PTMs can generate contextual embeddings which are more representative of the input sentence. These embeddings not only represent the semantic meaning, but also capture the contextualized meaning of the inputs. Secondly, since the PTMs have a large number of parameters (exceeding 100 million), they can be trained on very large datasets without overfitting. Thirdly, the same PTMs can be adapted for many NLP tasks with computationally inexpensive task-specific training. For instance, after training a PTM for a text classification task on a biomedical domain, the same model can be used for a question answering task on the same domain by only changing the last layers of the model and retraining on a small task-specific dataset. Finally, PTMs require less hyperparameter tuning as they are less sensitive to the changes in hyperparameter values and random initial weights compared to most deep learning architectures. Note that training PTMs from scratch is generally a computationally expensive process due to their large number of parameters. However, many PTMs are open-sourced and publicly available.

Accordingly, it is possible to load weights of a PTM, and fine-tune the model on a smaller task-specific dataset to lessen the computational burden (Devlin et al. 2018).

The adoption of pretrained language models can also be seen in the most recent work on product category classification (Tagliabue et al. 2021; Zhang et al. 2020). Most of the recent literature on product category classification problems can be found in the "Semantic Web Challenge" competition and case studies published by the competing teams (Zhang et al. 2020). The second part of the challenge focuses on multi-level product category classification. The considered dataset in the competition contains more than 15,000 product instances randomly sampled from 702 vendors' websites. The products are labeled in GPC hierarchy[1]. As baseline models, teams tested the same configuration proposed by Tagliabue et al. (2021), which uses the FastText algorithm. For evaluating the results, standard metrics such as Precision, Recall, and F1-score are used, and to measure the overall performance, Weighted-Average macro-F1 (WAF1) scores are reported by each of the participating teams. All the top submissions ended up using variants of the BERT (Devlin et al. 2018) architecture. For instance, Zahera and Sherif (2020) (the Team DICE) proposed a multi-label BERT architecture called ProBERT for the multi-label product category classification. ProBERT contains fully-connected neural layers with Sigmoid activations for each classification task. The winner of the competition, Yang et al. (2020) (Team Rhinobird) proposed a slightly more complex method, which uses BERT as the base model. To obtain a semantically rich

[1] https://www.gs1.org/standards/gpc

representation, they used hidden states from the last hidden layers of BERT, resulting in 17 different BERT models. These models were then combined using a two-level ensemble strategy. In the first level, they applied 5-fold cross-validation (CV) by splitting the training data into training and validation sets. Then, they trained the same model five times, each time using a different fold as the validation set and the remaining folds as the training set. Afterwards, they averaged the probability outputs of these five models with the same model architecture but trained on a different dataset. In the second level, an ensemble of 17 different BERT models was created where each model votes for the prediction, and the most voted class is selected as the final prediction. Moreover, they proposed a Dynamic Masked Softmax function that explicitly considers the dependencies among different category levels (Zhang et al. 2020). The dynamic masking of the subcategories based on the predicted category reduces the complexity of the optimization problem by filtering out the child categories unrelated to the predicted parent category.

3 Methodology

We provide details on the datasets and the methods employed for our multilevel product categorization task in this section. We also briefly discuss the experimental setup, evaluation metrics and hyper-parameter selection.

3.1 Datasets

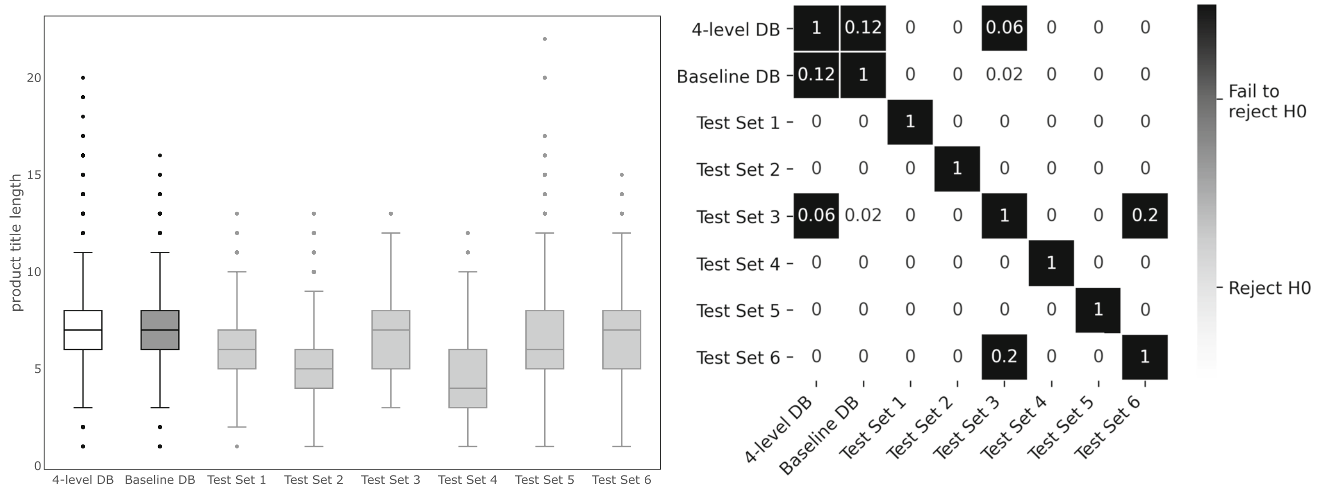
Our datasets are mined from Turkish online grocery marketplaces. We extracted product information from seven online grocery retailers as can be seen in Table 2. As the labels were inconsistent, a team of researchers manually explored the cases where the categories can be unified. Note that, to perform cross-platform classification, we employ one platform as the training set (i.e., Baseline DB), while the others serve as the test sets. This approach is mainly adopted to conform with the practical use cases for the product category classification task. For instance, an online retailer can use their own dataset to train a classification model, which can then be used to classify the products of other retailers. This way, the company can gather information about the products sold in the marketplace by type/category. Moreover, using the dataset from a single platform as the training set helps assessing the generalizability of the classification models. Most of the datasets have two category levels (category and subcategory); however, we also consider a 4-level categorization for the products, which additionally includes segment and subsegment information. We limit products in the testing sets to the ones whose categories and subcategories are present in the training set. Table 2 lists the number of items, categories, subcategories, segments, and subsegments before and after the data cleaning process. Only one dataset (4-level DB) has all four labels, and we use it for a more in-depth analysis of the product category classification models.

The distribution of product title length, employed as the independent attribute in our classification task, is depicted in Fig. 1a. Titles are usually short, with an average length of 6.6 words, and follow a similar distribution across different datasets.

We note that this short textual information may make the learning process more challenging. We also investigate the similarity of the product title length distributions for all the datasets. Specifically, we employ the Kruskal-Wallis H test, a rank-based nonparametric test with the null hypothesis that the medians of all groups are equal. We test it for the significance level of 0.05, and with the p-value approximating zero, we reject the null hypothesis. We further pinpoint which specific groups significantly differ from the others using Dunn's post hoc test. Figure 1b

Table 2 Dataset descriptions including number of items as well as number of category (Cat), subcategory (Subcat), segment (Seg), and subsegment (Subseg) values

Datasets	Before cleaning					After cleaning				
	Item	Cat	Subcat	Seg	Subseg	Item	Cat	Subcat	Seg	Subseg
Baseline DB	3,119	18	113	-	-	-	-	-	-	-
Test Set - 1	2,039	15	81	-	-	284	3	7	-	-
Test Set - 2	3,094	27	142	-	-	1,891	16	90	-	-
Test Set - 3	659	17	84	-	-	656	15	81	-	-
Test Set - 4	2,435	22	120	-	-	1,235	13	50	-	-
Test Set - 5	6,981	26	366	-	-	378	2	7	-	-
Test Set - 6	3,892	20	114	-	-	995	8	27	-	-
4-level DB	24,996	15	101	145	749	24,338	14	94	114	493



(a) The distribution of product title lengths

(b) Dunn's test for pairwise comparison

Fig. 1 Comparison of product title lengths of the datasets

shows the p-values of pairwise comparison of product title lengths for different sets. We reject the null hypothesis that there is no difference between groups at the significance level of 0.05. The white grids indicate the significant difference between the distributions of the datasets. We note that, in many cases, the product title lengths differ between products, making cross-platform category prediction more difficult.

3.2 Baseline text classification models

We employ standard text classification models, namely, XGBoost, Support Vector Machine (SVM), and Long Short-Term Memory networks (LSTMs). XGBoost is a scalable tree boosting approach, which generates a collection of weak trees by incorporating instances that have the most contribution to the model's learning process. On the other hand, SVM, a popular approach in text categorization (Goudjil et al. 2018; Wang and Qu 2017; WibowoHaryanto et al. 2018), performs the classification by identifying the support vectors (i.e., data points closer to the separating hyperplane) that separates the data instances. These points are chosen such that the margin of the classifier is maximized. These two models cannot handle textual information directly and require a numeric representation of it. Accordingly, we employ two approaches for transforming product titles to numeric vectors. Firstly, we utilize word embeddings to convert each word to a numeric vector. Specifically, we use TF-IDF values of each word as its weight, and apply a weighted average on the whole title using TF-IDF and the word embeddings (Stein et al. 2019). Secondly, we use Sentence-BERT to convert each title directly to its numeric representation (Reimers and Gurevych 2019). BERT has the advantage of considering the semantics of the vocabularies around the targeted word.

LSTMs are specific recurrent neural network architectures. They can model both long and short-term impacts of any textual information by utilizing the input, output, and forget gates in LSTM cells. Because of their capacity to maintain new or relevant information when needed and forget old or irrelevant information, LSTMs are appropriate models for text classification problems. We employ a Bi-directional LSTM (BiLSTM) model in our analysis, which has the ability to learn textual input from both directions, and then integrate it using convolutional neural networks to create a single expression (Li et al. 2018). We train separate networks for the four prediction tasks. That is, we feed in the product titles to these four networks, and the corresponding network returns the associated prediction for category, subcategory, segment or subsegment.

3.3 Pretrained language models

We fine-tune the large pretrained language models after attaching fully-connected layers to the output layer for the category, subcategory, segment and subsegment labels. Before the fully-connected layers, the hidden layers of the pre-trained language model is concatenated to obtain a detailed contextual representation of the model input (e.g., product title). Softmax activations are used after the fully-connected layers to obtain the class probabilities. We use the Dynamic Masked Softmax (Yang et al. 2020) for the subcategorical labels to enhance the classification performance. Three large pretrained language models with multi-lingual embeddings are considered in our study, namely, BERT, XLM, and XLM-RoBERTa.

3.3.1 BERT

BERT is a well-known pretrained language model. It obtains a detailed contextual representation by jointly conditioning the input on both left and right context in all layers. After unsupervised pretraining on a large dataset, the BERT model can be fine-tuned by adding an output layer to generate state-of-art NLP models for a variety of tasks. For our experiments, we use two versions of this architecture. First architecture is the multi-lingual version of BERT pretrained on 102 languages with the largest Wikipedia using a masked language modelling (MLM) objective (Devlin et al. 2018). Second architecture is the Turkish language version (Oluk 2020), pretrained on 200 GB of Turkish text.

3.3.2 XLM

Conneau and Lample (2019) use a novel training objective that allows unsupervised training for learning high-quality cross-lingual representations. Specifically, XLM uses the Translation Language Modelling (TLM) objective in addition to the BERT's MLM objective. In TLM objective, the same sentence in two different languages are concatenated. Then random masking is applied in the concatenated sentence in the training phase. The model uses the surrounding context in the same language as well as the context given by the second language to predict a masked token. Accordingly, the XLM model can learn superior cross-lingual word embeddings by using both the MLM and the TLM objectives. We employ the multi-lingual version of this architecture for in our analysis, which has been pretrained on 102 languages with the largest Wikipedia dataset (Devlin et al. 2018).

3.3.3 XLM-RoBERTa

Liu et al. (2019) propose RoBERTa architecture which involves a number of modifications to the original BERT architecture that improves the prediction performance on a variety of NLP benchmarks. The main changes include altering certain initial hyperparameters, removing the next-sentence prediction objective, and training with higher batch sizes. We employ the multi-lingual version of this architecture pretrained on 2.5TB of CommonCrawl data in 100 languages using a masked language modelling (MLM) objective.

3.4 Dynamic masking

In the standard classification approach for product categorization using pretrained language models, the last hidden states of the base model are concatenated to obtain an encoded representation of the inputs. Then, feed-forward network layers with Softmax activations are applied to compute the probabilities for the categories of each level. However, the standard Softmax layer does not take into account the dependencies between the parent (e.g., category) and child (e.g., subcategory) categories.

An alternative method proposed by Yang et al. (2020) uses Dynamic Masked Softmax to dynamically filter out the child categories that are unrelated to the current parent category (see Fig. 2). For instance, if the category c_i is predicted by the model, the model should only recommend subcategories that fall in the same category. Hence, infeasible subcategories should be filtered out using a binary mask. In this method, we define a binary mask matrix $M \in \{0,1\}^{C \times S}$, where C is the number of categories and S is the number of subcategories. We then compute the Dynamic Masked Softmax instead of regular Softmax for computing the subcategory predictions as

$$P(y_s|c,\theta) = \frac{\exp(O_s)M_{c,s} + \exp(-8)}{\sum_{s'=1}^{S} \exp(O_{s'})M_{c,s'} + \exp(-8)} \tag{1}$$

where c and s correspond to category and subcategory labels, respectively, θ denotes the model parameters, and y_s denotes

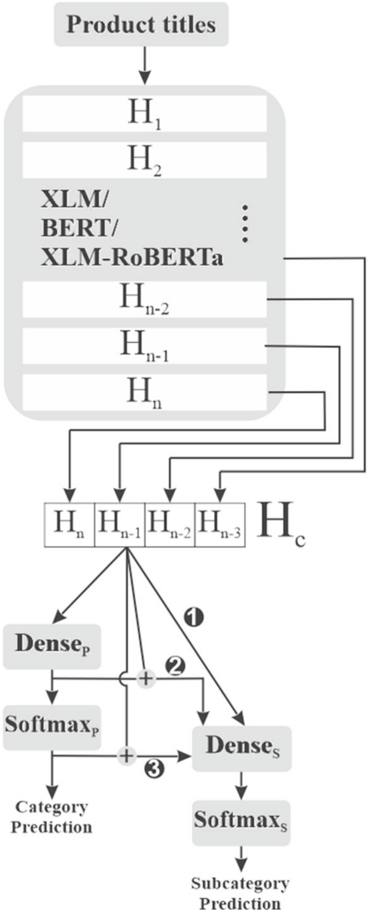


Fig. 2 Masked XLM, XLM-RoBERTa and BERT (Yang et al. 2020)

the predicted probability for the subcategory s. Note that $\exp(-8)$ terms are included for numerical study. This design can also be extended to more than two levels as needed.

In the dynamic masking approach, the prediction of a sub-level category depends on the prediction of its parent level category. For instance, predicting the subcategory "Tea" depends on predicting the associated main category "Beverage" correctly. In our numerical analysis, we experiment with three different configurations of dynamic masking and compare these variants with the standard approaches. For all three configurations, we first extract the hidden layers that make up the transformer architecture, and concatenate them. Then, we apply feed-forward (Dense) layers and Softmax activation functions for each multi-class classification task.

In the vanilla configuration, the information about the parent level prediction is not carried to the child level prediction. These two prediction tasks are executed independently. Specifically, the input to the feed-forward network layers of each category level is the hidden state of the base model (H_c), as shown in configuration 1 in Fig. 2. As an alternative, it is possible to provide additional information about the parent level's prediction for the child level's prediction. For this purpose, we can concatenate the hidden state of the transformer model with the parent level's output before predicting the child level. This information can be provided in two ways. In the first configuration, the output of the parent level's feed-forward layer (Dense_p) before the activation is provided to the child (Dense_s). In the second configuration, the output after the softmax activation ($\mathrm{Softmax}_p$) is provided.

3.5 Experimental setup

Figure 3 depicts our experimental setup consisting of two parts: within- and cross-platform grocery product category prediction. We first investigate the performance of the models in predicting the product categories (Part I). Accordingly, Baseline DB described in Section 2 is used as the training set, and it is subjected to 5-fold CV. Then, we conduct experiments to explore the performance of different models and word embeddings. Furthermore, for a more comprehensive evaluation of the dynamic subcategory masking for the pre-trained language models, we train these models again on a larger dataset.

For this section, we use 4-level DB dataset, which contains more than 24,000 products categorized to four levels. Secondly, we use the best performing models in Part I to determine their generalizability as indicated by their performance for predicting the products of other online platforms.

The analysis with the cross-platform prediction also helps understanding the capabilities of the models in predicting new possible products in the system.

3.5.1 Evaluation metrics

We utilise Brier score, accuracy and the weighted-average macro F1-score (WAF1) to evaluate the classification performance. Brier-score is a common accuracy measure for probabilistic predictions. It is calculated for each level by measuring the mean squared distance between the predictions and the binary truth vectors, with larger differences

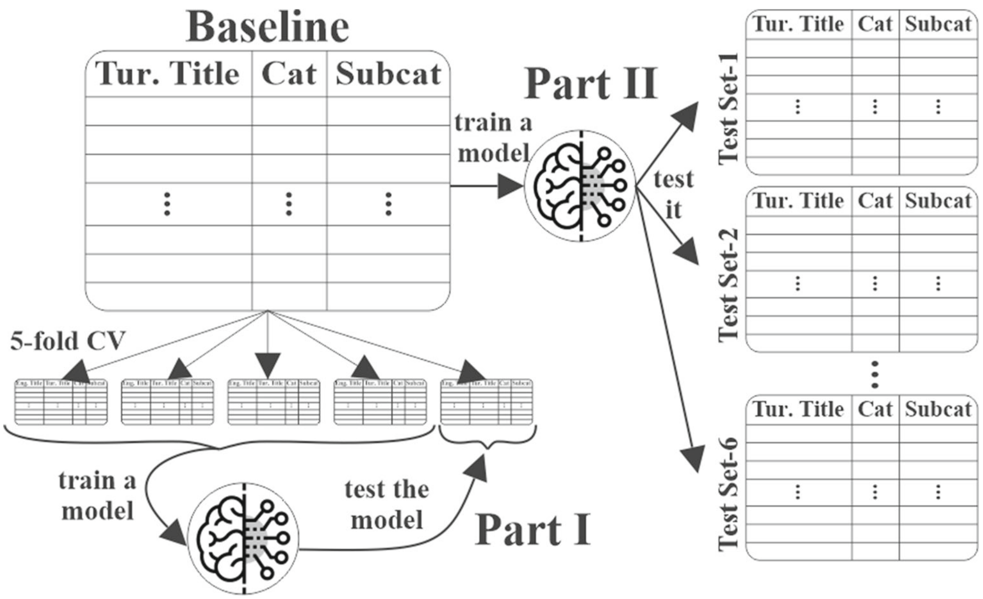


Fig. 3 Experimental setup

reflecting higher error in the prediction. Thus, a lower score indicates better performance. Brier score is calculated as

$$\text{Brier-score} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}(f_{ik} - o_{ik})^2$$

where N is the number of samples, and K is the number of classes. For a given instance $i \in \{1, \ldots, N\}$, f_{ik} represents the predicted probability for class k, and o_{ik} values encode the ground truth vector (kth value being one and others being zero). On the other hand, accuracy is a simple statistic that indicates how often a model makes a correct prediction. Lastly,

F1-score is calculated as the harmonic mean of precision and recall, that is, F1-score $= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$.

These values are determined separately for each class, and WAF1-scores are derived using a weighted average. Similar to Zhang et al. (2020), we rank the models by taking the average WAF1 for all category levels.

3.5.2 Parameter settings

Tensorflow and Transformers libraries are used to implement the BiLSTM and pretrained language models (Wolf et al. 2020). We fine-tune the pretrained transformers *bert-base-turkish-uncased*, *bert-base-multilingual-uncased*, *xlm-mlm-100-1280*, and *jplu/tf-xlm-roberta-base* for the classification tasks. We utilize Adam optimizer and a batch size of 16 for all pretrained language models. Early-stopping is used during the training phase to minimize over-fitting.

The weights of the best-performing model on the validation set are saved and used for testing.

We also perform a grid search to fine-tune the hyperparameters. As mentioned in Sections 4.1 and 4.3, this procedure is carried out via a distinct validation set. Table 3 shows the parameter search space for each model. The best combinations of hyperparameters are in bold. We observe a high sensitivity in the SVM and XGBoost to their parameters, whereas LSTM and transformers are less affected by the change in their network-related parameters. We use scikit-learn and xgboost packages in Python for implementing the SVM and XGBoost models, respectively.

4 Numerical results

In our numerical study, we first focus on identifying the best performing models on the baseline dataset. Next, we assess the impact of dynamic masking approaches for the multilevel product categorization task. We examine the generalizability of the model performances to different datasets extracted from various online grocery retailers. In this experiment, classification models are trained with the two-level baseline dataset (i.e., with category and subcategory labels), and tested on six two-level test sets. Lastly, we demonstrate misclassifications for sample data instances, and discuss the underlying causes and remedies to improve classification performance.

4.1 Performance comparison

We examine alternative word embeddings and a wide range of multi-class classification methods, including classic machine learning algorithms (e.g., SVM and XGBoost), as well as more contemporary deep learning methods (e.g., BiLSTM and BERT) for the multilevel product categorization task. Using a 90-10 division over the training set, we create the validation set and optimize the model parameters accordingly. Then, using the stratified CV, we split the dataset into five folds, where one fold is used as the test set and

Table 3 Model parameters and search space

Model	Hyperparameters
SVM	kernel = [**linear**, poly, rbf], degree of polynomial kernel= [**1**,2,3,4,5] C = [1, **10**, 100, 1000]
XGBoost	number of trees = [**100**, 150, 200], max depth = [**2**, 3, 5], learning rate = [0.01, 0.1, **0.5**]
LSTM	Embedding layer, BiLSTM layer = [100, 200, **300**], Dense layer = ([50, 100, 150, **200**], relu activeation), Dense layer = (n of cat/subcat, softmax activation) optimizer= adam, early stopping patience=10, #epochs=[**20**,100]
BERT-ML	learning rate=[**5e-5**, 4e-5, 3e-5, 2e-5], n_last_hidden_layer=[**1**, 2, 3] batch size=16, early stopping patience=10, #epochs=100
BERT-TR	learning rate=[5e-5, 4e-5, 3e-5, **2e-5**], n_last_hidden_layer=[1, **2**, 3] batch size=16, early stopping patience=10, #epochs=100
XLM-ML	learning rate=[5e-5, 4e-5, 3e-5, **2e-5**], n_last_hidden_layer=[**1**, 2, 3] batch size=16, early stopping patience=10, #epochs=100
XLM-RoBERTa-ML	learning rate=[5e-5, 4e-5, **3e-5**, 2e-5], n_last_hidden_layer=[**1**, 2, 3] batch size=16, early stopping patience=10, #epochs=100

Identified hyperparameter values are bolded the remaining as the training set. This process is repeated five times for all folds. This approach enables identifying appropriate parameters and minimize overfitting.

Table 4 illustrates the performance of each model and word embedding settings as obtained by 5-fold CV.

We find that Turkish Glove embedding provides a superior numeric representation for the vocabularies utilized in our dataset when compared to Turkish FastText, Sentence-BERT, and Turkish Word2Vec embeddings. BiLSTM with Turkish Glove embedding leads to best overall performance among the standard text classification methods as indicated by Brier-score (avg. Cat: 0.06, avg. Sub: 0.14), accuracy (avg. Cat: 96.6%, avg. Sub: 90.6%) and F1-score (avg. Cat: 96.5%, avg. Sub: 90.1%). Pretrained language models mostly perform similarly with BERT-based models performing slightly better than others, followed by XLM and XLM-RoBERTa models as indicated by the average F1-scores. Surprisingly, the vanilla pretrained language models outperformed the masked configurations of the models for the two-level product classification task.

Overall, these results show that the vanilla BERT architecture with multilingual embedding performed the best among all considered classification model-word embedding pairings, closely followed by other transformer architectures.

4.2 Impact of dynamic masking

We investigate the effects of dynamic masking on multi-level product category classification in a more detailed manner considering the four-level classification task.

Table 5 shows the results for the pretrained language models with three masking strategies obtained by using 4-level DB.

We find that the BERT with Turkish Embedding performed the best, with a marginal improvement over the BERT with multilingual embedding. We also see that the rank between the masking configurations are consistent across all transformer architectures in terms of the average F1-score. The non-masked configuration performed best, followed by the "Mask", "Mask After Activation", and "Mask Before Activation" configurations.

The results shown in Tables 4 and 5 indicate that the masking mechanisms do not necessarily improve the classification performance of the model. This result can be due to the availability of semantically similar child categories

Table 4 Comparison of different models and word embeddings using Baseline DB (performance values reported as avg. ± stdev, which are obtained using 5-fold CV results)

Model	Word embedding	Brier-score		Accuracy (%)		F1-score (%)		
		Cat	Sub	Cat	Sub	Cat	Sub	Avg.
SVM	Turkish Word2Vec	0.25±0.01	0.57±0.02	85.0±1.0	74.0±1.5	84.7±1.1	72.2±1.8	78.4
XGBoost		0.28±0.01	0.45±0.01	80.6±0.6	68.2±0.4	80.3±0.6	67.0±0.5	73.6
Bi-LSTM		0.10±0.01	0.21±0.03	93.8±0.8	86.4±2.5	93.8±0.8	85.9±2.2	89.9
SVM	Turkish FastText	0.32±0.02	0.70±0.02	78.5±1.0	62.0±1.8	78.8±0.9	57.1±2.0	67.9
XGBoost		0.19±0.02	0.37±0.02	87.2±1.4	74.9±1.8	86.8±1.5	73.8±1.7	80.3
Bi-LSTM		0.08±0.01	0.16±0.01	95.2±0.9	89.5±0.6	95.2±0.8	89.2±0.6	92.2
SVM	Turkish Glove	0.11±0.02	0.39±0.03	93.5±1.2	88.9±2.1	93.4±1.1	88.1±2.3	90.8
XGBoost		0.14±0.02	0.29±0.02	90.4±1.1	80.7±1.6	90.2±1.2	79.7±2.0	84.9
Bi-LSTM		0.06±0.02	0.14±0.01	96.6±0.9	90.6±0.8	96.5±0.9	90.1±0.7	93.3
SVM	Multilingual BERT	0.18±0.02	0.31±0.02	86.0±1.8	79.6±2.1	86.6±1.8	81.9±1.7	84.2
XGBoost		0.20±0.02	0.32±0.02	83.4±1.4	77.0±1.7	84.1±1.5	78.4±1.9	81.3
BERT-ML	Multilingual	0.06±0.01	0.12±0.01	96.5±0.9	93.1±1.0	96.4±0.9	92.9±1.1	94.6
BERT-TR	Turkish	0.06±0.01	0.13±0.02	96.6±0.2	92.5±0.8	96.6±0.2	92.3±0.9	94.4
XLM-ML	Multilingual	0.06±0.01	0.13±0.02	96.6±0.9	92.2±1.4	96.5±0.9	91.9±1.6	94.2
XLM-RoBERTa-ML	Multilingual	0.06±0.01	0.13±0.02	96.4±0.4	92.0±1.4	96.3±0.4	91.6±1.3	93.9
BERT-ML$^{\mathcal{M}}$	Multilingual	0.07±0.02	0.14±0.01	96.0±1.3	91.6±1.0	96.0±1.3	91.4±1.0	93.7
BERT-TR$^{\mathcal{M}}$	Turkish	0.06±0.01	0.14±0.03	96.6±0.7	92.5±1.5	96.5±0.7	92.1±1.5	94.3
XLM-ML$^{\mathcal{M}}$	Multilingual	0.05±0.01	0.14±0.00	96.9±0.3	91.6±0.3	96.9±0.3	91.2±0.2	94.1
XLM-RoBERTa-ML$^{\mathcal{M}}$	Multilingual	0.07±0.01	0.16±0.02	95.7±0.5	90.4±1.3	95.6±0.5	89.8±1.2	92.7

$^{\mathcal{M}}$: Mask, ML: Multilingual Embedding, TR: Turkish Embedding

Table 5 Comparison of dynamic masking approaches using 4-level DB (performance values reported as avg. ± stdev, which are obtained using 5-fold CV results)

Model	Brier-score				Accuracy (%)				F1-score (%)
	Cat	Subcat	Seg	Subseg	Cat	Subcat	Seg	Subseg	Avg.
BERT-ML	0.06±0.01	0.11±0.01	0.13±0.02	0.30±0.05	96.6±0.8	93.2±0.8	92.3±1.1	81.1±3.2	90.3
BERT-ML$^{\mathcal{M}}$	0.05±0.01	0.12±0.02	0.17±0.02	0.33±0.04	96.9±0.9	92.7±1.1	91.4±1.1	81.2±2.4	90.1
BERT-ML$^{\mathcal{MA}}$	0.06±0.01	0.13±0.01	0.17±0.02	0.34±0.03	96.7±0.6	92.4±0.8	91.3±1.0	80.8±2.0	89.8
BERT-ML$^{\mathcal{MB}}$	0.06±0.01	0.14±0.02	0.18±0.02	0.36±0.04	96.3±0.9	92.0±1.1	90.7±1.2	79.5±2.5	89.0
BERT-TR	0.06±0.01	0.11±0.02	0.12±0.01	0.32±0.05	96.4±0.8	93.8±1.0	92.9±0.9	81.3±2.9	90.6
BERT-TR$^{\mathcal{M}}$	0.05±0.01	0.13±0.02	0.16±0.02	0.32±0.05	96.8±0.7	93.0±1.1	91.8±1.1	82.3±2.8	90.5
BERT-TR$^{\mathcal{MA}}$	0.06±0.01	0.13±0.02	0.16±0.02	0.34±0.04	96.7±0.5	93.0±1.0	91.7±1.0	81.1±2.3	90.1
BERT-TR$^{\mathcal{MB}}$	0.06±0.01	0.13±0.02	0.17±0.02	0.35±0.04	96.4±0.8	92.5±1.1	91.1±1.2	80.6±2.2	89.6
XLM-ML	0.05±0.01	0.11±0.01	0.13±0.01	0.32±0.04	96.7±0.7	93.4±0.6	91.9±0.7	79.9±2.4	89.8
XLM-ML$^{\mathcal{M}}$	0.06±0.01	0.14±0.02	0.19±0.02	0.37±0.04	96.3±0.7	91.6±1.1	90.4±1.0	77.9±2.0	88.4
XLM-ML$^{\mathcal{MA}}$	0.06±0.02	0.14±0.02	0.19±0.03	0.38±0.05	96.3±1.1	91.5±1.3	90.2±1.3	77.8±2.9	88.3
XLM-ML$^{\mathcal{MB}}$	0.06±0.01	0.14±0.02	0.19±0.02	0.38±0.04	96.1±0.9	91.6±0.9	90.1±1.2	77.4±2.7	88.1
XLM-RoBERTa-ML	0.06±0.01	0.12±0.01	0.13±0.02	0.35±0.05	96.3±0.6	92.8±0.8	91.6±0.9	77.9±2.8	89.0
XLM-RoBERTa-ML$^{\mathcal{M}}$	0.07±0.01	0.15±0.01	0.21±0.02	0.41±0.03	95.7±0.4	90.7±1.0	89.1±1.1	76.0±1.8	87.2
XLM-RoBERTa-ML$^{\mathcal{MA}}$	0.07±0.01	0.15±0.02	0.20±0.03	0.41±0.04	95.9±1.0	90.8±1.4	89.2±1.4	75.7±2.7	87.1
XLM-RoBERTa-ML$^{\mathcal{MB}}$	0.07±0.01	0.16±0.02	0.22±0.02	0.42±0.05	95.7±0.8	90.4±0.8	88.8±1.3	75.1±3.0	86.7

$^{\mathcal{M}}$: Mask, $^{\mathcal{MA}}$: Mask After Activation, $^{\mathcal{MB}}$: Mask Before Activation,
ML: Multilingual Embedding, TR: Turkish Embedding

under different parent categories. In such cases, when the masked models are used, if the parent category is incorrectly predicted, the child categories will be misclassified as well. This is because all the subcategories that are under a different category are masked. However, in the non-masked models, the model can still predict the child category correctly, even though the parent category is predicted incorrectly. We present mispredictions from the validation dataset using the non-masked BERT model (BERT-TR) in Table 6 to better illustrate this phenomenon. In these examples, we observe that the child category prediction can be correct even though the parent category prediction is incorrect. For example, in the first instance, a subcategory for a liquid soap product is misclassified as “Cosmetics” instead of “Soap”. However, the remaining category levels are predicted correctly.

Table 5 provides other evidence on why masked versions might not perform as well as the vanilla models.

For instance, we observe that the average reduction in the child-level category accuracies are higher when masked configurations are used, indicating that masking leads to additional performance deterioration when predicting subcategorical labels. It is worth noting that there are some exceptions to this such as subsegment of the BERT-ML model. We believe these exceptions can be explained by the randomness introduced by the mixed training objective.

Overall, we find that the use of dynamic masking has certain benefits and drawbacks. It reduces the complexity of the prediction task by creating a mask over the sub-levels which do not belong to the predicted parent category. This helps achieving higher prediction performance when the number of classes are high for that category. On the other hand, when the boundaries/distinction between the categories are not clear in the dataset, the use of dynamic masking can have a detrimental affect on the model performance. This mainly stems from the fact that when the dynamic masks are used, a wrong prediction would mean all the sub-level predictions will be incorrect. In cases where the parent category predictions are not accurate, the misclassifications can compound at each level, leading to a lower sub-level prediction performance. In brief, the dynamic masking is most useful when the classes are strictly separated, the higher-level categories are accurately predicted, and the number of categories at the sub-levels are high in the dataset. Accordingly, the model design needs to take into account these factors before implementing a dynamic masking configuration.

4.3 Cross-platform multilevel classification

We next discuss our findings with the cross-platform classification task.

Note that certain categories and subcategories in the extracted datasets may not exist in the training set (i.e., Baseline DB). Accordingly, we only take into account the items whose categories and subcategories are available in the training set. We use the models with the best hyperparameters described in Section 3.5.2 to perform the predictions. We employ the models trained in each of the 5-fold CV process to predict the labels in the test sets, which helps achieve a robust evaluation. Therefore, we report the average and standard deviation over the performance values for each test set obtained by five prediction models (see Table 7).

Our analysis in Section 4.1 shows the best performing models for within-platform product categorization as BERT-ML, BERT-TR, and XLM-ML. In cross-platform prediction, surprisingly, BiLSTM outperforms or performs

Table 6 Examples of the non-masked model mispredictions obtained by BERT-TR (texts are translated from Turkish to English, wrong predictions are italicized)

Product	Ground truth				Predicted Category			
	Cat	Subcat	Seg	Subseg	Cat	Subcat	Seg	Subseg
Dove liquid soap avocado oil 500 ml	Personal Care	Soap	Hand, Body and Face Products	Liquid Soap	Personal Care	*Cosmetics*	Hand, Body and Face Products	Liquid Soap
Milango moments cherry particled chocolate milk 76 gr	Snacks	Chocolate	Chocolate	Chocolate Milk	Snacks	*Wafer*	*Wafer*	Chocolate Milk
Dogadan white tea plain 18-pack	Beverage	Tea	Herbal Tea	Tea Bag	Beverage	Tea	*Cold Tea & Coffee*	Tea Bag
Dentac surgical mask colored mix 10-pack	Personal Care	General Health	Health	Surgical Mask	Personal Care	General Health	*Oral Care*	Surgical Mask
Nivea moisturizer sunscreen spray 200 ml	Personal Care	Body and Hand Care	Hand, Body and Face Products	Sun Protection	Personal Care	*Cosmetics*	Hand, Body and Face Products	Sun Protection

Table 7 Models' performance of cross-platform category prediction for competitor analysis

Models	Metrics	Test Set - 1			Test Set - 2			Test Set - 3			Test Set - 4			Test Set - 5			Test Set - 6		
		Cat	Subcat	Avg.	Cat	Subcat	Avg.	Cat	Subcat	Avg.	Cat	Subcat	Avg.	Cat	Subcat	Avg.	Cat	Subcat	Avg.
BiLSTM	Brier-score	0.18±0.03	0.32±0.02	**0.25**±0.07	0.31±0.04	0.37±0.02	0.34±0.04	0.03±0.00	0.06±0.01	**0.04**±0.01	0.47±0.06	0.56±0.03	0.52±0.06	0.23±0.02	0.38±0.02	**0.31**±0.08	0.30±0.01	0.36±0.02	**0.33**±0.03
BERT-ML		0.27±0.07	0.61±0.12	0.44±0.06	0.32±0.03	0.46±0.01	0.39±0.02	0.07±0.01	0.10±0.01	0.08±0.01	0.46±0.04	0.65±0.01	0.55±0.02	0.39±0.04	0.50±0.08	0.44±0.05	0.45±0.04	0.48±0.03	0.46±0.03
BERT-TR		0.22±0.06	0.43±0.03	0.32±0.02	0.27±0.01	0.40±0.02	**0.33**±0.02	0.05±0.01	0.08±0.01	0.07±0.01	0.38±0.02	0.57±0.02	**0.48**±0.02	0.25±0.06	0.39±0.05	0.32±0.05	0.36±0.03	0.40±0.02	0.38±0.01
XLM-ML		0.21±0.05	0.61±0.06	0.41±0.06	0.32±0.03	0.44±0.02	0.38±0.02	0.07±0.01	0.10±0.02	0.09±0.01	0.44±0.04	0.61±0.02	0.53±0.03	0.39±0.05	0.47±0.03	0.43±0.03	0.46±0.03	0.45±0.02	0.46±0.01
XLM-RoBERTa-ML		0.16±0.06	0.60±0.11	0.38±0.08	0.31±0.04	0.42±0.01	0.36±0.02	0.07±0.01	0.09±0.01	0.08±0.00	0.43±0.05	0.60±0.01	0.52±0.03	0.30±0.07	0.47±0.03	0.38±0.05	0.40±0.04	0.46±0.02	0.43±0.02
BiLSTM	Accuracy	89.2±1.6	79.0±1.1	**84.1**±5.2	81.0±2.4	75.3±1.2	78.2±3.4	98.2±0.3	96.4±0.8	**97.3**±1.1	71.0±3.6	63.1±1.7	67.1±4.9	86.3±1.5	75.7±1.3	**81.0**±5.5	82.2±0.8	76.8±1.5	**79.5**±3.0
BERT-ML		83.4±4.6	61.1±5.8	72.2±3.8	81.0±2.1	71.8±1.5	76.4±1.8	95.8±0.3	94.2±0.9	95.0±0.6	73.1±3.1	59.9±1.9	66.5±2.5	77.3±2.2	68.4±4.8	72.8±3.1	73.4±1.7	69.2±0.9	71.3±1.2
BERT-TR		85.0±4.0	68.4±2.0	76.7±2.4	83.1±1.2	74.4±0.8	**78.7**±0.9	96.9±0.4	95.0±0.5	95.9±0.4	75.8±1.8	63.4±1.4	**69.6**±1.4	83.9±4.5	74.0±2.0	78.9±2.7	78.0±2.1	73.4±1.9	75.7±1.4
XLM-ML		86.5±3.0	61.3±3.2	73.9±3.0	81.3±1.9	72.3±1.2	76.8±1.6	95.7±0.6	93.4±1.1	94.6±0.9	73.6±2.7	61.1±1.3	67.4±2.0	77.2±3.1	68.0±2.5	72.6±2.5	73.4±1.6	70.0±1.8	71.7±1.2
XLM-RoBERTa-ML		90.4±3.2	62.7±6.4	76.6±4.6	81.6±2.1	73.5±0.8	77.6±1.2	96.0±0.7	94.1±0.6	95.1±0.3	73.9±2.9	62.6±1.1	68.3±1.8	82.2±4.3	67.6±2.4	74.9±3.2	76.1±1.6	68.3±1.8	72.2±1.5
BiLSTM	F1-score	93.0±1.0	85.1±1.3	**89.0**±4.1	83.7±1.9	78.8±1.0	**81.2**±2.9	98.2±0.3	96.5±0.8	**97.3**±1.0	77.6±2.9	69.8±1.5	**73.7**±4.5	90.9±1.0	84.1±1.1	**87.5**±3.6	86.6±0.7	81.8±1.4	**84.2**±2.6
BERT-ML		90.5±2.7	66.5±7.0	78.5±3.3	82.2±1.7	71.2±1.4	76.7±1.5	95.7±0.3	93.4±1.0	94.6±0.7	76.7±2.8	62.5±1.5	69.6±2.2	81.9±2.0	78.2±3.8	80.1±2.8	79.0±2.6	75.2±0.9	77.1±1.6
BERT-TR		91.7±2.4	73.4±1.7	82.6±1.7	84.1±1.2	73.0±0.9	78.6±0.9	96.8±0.4	94.2±0.6	95.5±0.4	79.4±1.6	64.9±1.2	72.1±1.2	89.1±2.8	82.7±1.5	85.9±1.9	83.8±1.2	79.3±1.2	81.6±0.7
XLM-ML		92.7±1.8	64.7±4.9	78.7±3.3	82.4±1.2	71.6±0.8	77.0±1.0	95.6±0.6	92.7±1.3	94.2±1.0	76.9±1.7	63.5±0.8	70.2±1.2	83.2±2.1	78.7±2.2	80.9±1.4	78.6±1.7	75.1±1.9	76.8±0.9
XLM-RoBERTa-ML		94.6±1.8	65.0±6.5	79.8±4.1	83.2±1.5	73.0±0.6	78.1±0.9	95.9±0.7	93.3±0.8	94.6±0.3	78.2±1.7	65.3±0.8	71.7±1.2	87.3±3.5	78.1±1.9	82.7±2.3	80.6±1.2	73.6±1.7	77.1±1.3

BiLSTM model uses Turkish Glove embeddings similarly to the BERT and XLM variants. The Brier-score and accuracy metrics rank the models on the same order. For Test Set-2 and Test Set-4, BERT-TR is the best performer, while for the remaining four datasets BiLSTM provides best performance values. F1-score metric finds BiLSTM as the best model for all six datasets. Overall, while within-platform performances of the models given in Table 4 are relatively close, there is a higher performance variation on cross-platform prediction.

Overall, the cross-platform product categorization outcomes show that the models trained on a comprehensive training dataset may be applied to the ones collected from other online grocery stores. However, we note that the product titles in each of the test sets are different, and, as the difference between the training set and the test set grows larger, the cross-platform performance deteriorates even more.

4.4 Discussion on model predictions

Lastly, we visually investigate the predicted category levels and the ground truth to find the items for which the models fail to predict the proper category and subcategory values. Table 8 shows sample data instances, the ground truth values and the predictions from BERT-TR model. We summarize our general observations on the misclassifications as follows:

- Should a product exist in the test set and not in the training set or contain different phrasing from the training set, a miscategorization may happen.
- The model's forecast is influenced by the general connotation of some brand names. For instance, "Raffaello" is a brand that produces chocolate. However, Raffaello is also a type of pasta. Accordingly, for the last instance in Table 8, the model predicts category and subcategory labels as "Pasta" and related labels, whereas the ground truth is related to "Chocolate".
- Generally, product categorization is done manually and thus it might be subjective. For example, one online store may classify a product as a dairy product, while another may categorize it as a beverage. Such a problem cannot be solved during the data cleaning process since we encounter a long list of product titles and category levels. A manual inspection may, nevertheless, result in subjectivity. Hence, we did not revise such category/subcategory labels in the data cleaning phase.
- Some product names might be associated with more than one category. A book on cooking, for example, may be classified as food rather than a book.
- In a four-level product category classification task, the finest level can be too specific and difficult to be predicted. Therefore, more data instances might be needed to have a better fit on each category level.

We observe that BERT-TR fails to predict the exact category or subcategory for some items, as shown in Table 8. Investigating failure cases can further reveal the reasons for misclassifications. "Dentiste Sensitive tube 100 gr" is a regular toothpaste that is imported to Turkey. The title does not have enough information other than "sensitive" and "tube". The model predicted the item as a shaving cream since shaving creams are presented in tube form and they are usually advertised for sensitive skins. For the product "Green Life fennel tea 150 gr Bag", the model predicted category, subcategory, and segment correctly, whereas it failed in the finest level possibly due to the keyword "Bag" in the title. "Piyale strawberry

Table 8 Examples of product category mispredictions obtained by BERT-TR (texts are translated from Turkish to English, wrong predictions are italicized)

Product	Ground Truth				Predicted Category			
	Cat	Subcat	Seg	Subseg	Cat	Subcat	Seg	Subseg
Dentiste Sensitive tube 100 gr	Personal Care	Oral Care	Toothpaste	Regular Toothpaste	Personal Care	*Shaving Supplies*	*Shaving*	*Shaving Cream & Aftershave*
Green Life fennel tea 150 gr Bag	Beverage	Tea	Herbal Tea	Tea Bag	Beverage	Tea	Herbal Tea	*Bulk Tea*
Piyale strawberry pudding 125 gr	Staple Food	Desserts	Powder Desserts	Pudding	Staple Food	Desserts	Powder Desserts	*Milk Dessert*
Begdat bay leaves	Staple Food	Herbs, Spices & Sauces	Spices	Assorted Spices	Staple Food	Herbs, Spices & Sauces	Spices	*Mixed Spices*
Raffaello 150 gr	Snacks	Sweet Snacks & Candy	Chocolate	Milk Chocolate	*Staple Food*	*Pasta, Rice & Beans*	*Pasta*	*Spaghetti Pasta*

pudding 125 gr" is categorized as milk dessert by the model while actual label is pudding. This misclassification can be justified by the fact that pudding is prepared by using milk. As such, a fresh pudding can be classified under either group. Such a delicate categorization could also be the reason why "Begdat bay leaves" are misclassified in the subcategory level. Raffaello is a brand that produces both snacks and it is also a type of pasta. Therefore, with such a short title of "Raffaello 150 gr", the model categorized it as spaghetti rather than milky chocolate.

We note that a retail company benefits from such detailed examinations by unifying similar subcategories, merging and revising misleading or subjective category levels, and expanding the product titles in their websites by adding more product information. This also benefits the search engine optimization of a company's online grocery stores. Accordingly, providing more information about the product's ingredients than a simple title may enhance prediction accuracy. Currently, many grocery websites in Turkey do not include those detailed information. Overall, the model performances are found to be reasonable considering the limited input provided to the models.

5 Conclusion

Companies rarely have complete knowledge about the items available in the marketplace while developing a marketing strategy. As a result, they typically predict the missing items and match it to their category definitions to have a better understanding of the market. Furthermore, they may seek to discover wrongly categorized items based on the products in their database to have a better sense of the current market trends. In this research, we investigated text classification techniques to automate the prediction of product category levels using product titles. We evaluated the extracted datasets of Turkey's leading online grocery platforms and used several machine learning techniques for the prediction task. We employed three different masking strategies to improve the accuracy of pretrained language models for the multi-level product classification. After conducting a detailed comparative analysis with these models, we closely examined the misclassified products. This analysis revealed many cases where the existing labels (i.e., category values) are debatable or difficult to predict even for human experts.

We observe that tuned text classification methods show high level of robustness for both within platform and cross-platform prediction tasks as evidenced by low standard deviation values around various performance metrics. In addition, our preliminary analysis indicate that deep learning-based classification methods are highly robust to model parameters as well. Low Brier-score values and higher than 90% accuracy and F1-score values indicate that these classification methods can be used in practice to avoid tedious manual categorization, automatically labeling the new products, and examining the large number of products sold by a retailer. They can also be used for enhanced marketing strategies and analytics as they enable categorizing the products in the marketplace, enabling more efficient data collection from other retailers. We note that the use of pre-trained versions of the large language models significantly reduces the computational requirements. That is, training a pre-trained model on a task-specific dataset takes a few hours on a GPU and between 24 and 48 hours on a regular CPU. As such, the product classification methods considered in this study can be employed by many retail companies.

We recognize that our research has certain limitations. While we ensure that we utilize both well-established and novel techniques in text categorization, as the NLP is a fast-changing field, continually analyzing the trends and applying alternative approaches for the prediction task are deemed to be necessary. Furthermore, because many online retailer websites do not provide additional specific information about the products, we only consider product titles for the category classification task. Future research can integrate product descriptions, specifications, and prices to enhance prediction performance.

Designing text classification models to improve the performance in certain product categories might be an interesting future research direction. The trained models, for example, exhibit low accuracy in the "Newspaper & Magazine" subcategory. Fine-tuning the models on an online book store dataset or training a second book/non-book classifier can improve the predictions for this category without compromising performance in other areas. In addition, considering that different models perform better for different cross-platform prediction tasks, an ensemble of models can enhance the prediction performance. Moreover, Stochastic Weight Averaging (SWA) may be used to further improve the performance for unbalanced product category cases. Finally, the semantic similarity of some categories might be problematic for both models and practitioners. Companies may use these models to revise their manual categorization and potential bias or overlap in their product labeling. Occasionally revisiting such information may help to better organize the products on online platforms. Therefore, we aim to expand our empirical analysis as we collect more data from the marketplace.

Acknowledgements The authors would like to thank the Getir company for supporting this study and providing data and feedback throughout.

Declarations

Data Availability Statement All the datasets are propriety and obtained from Getir Perakende Lojistik A.S.

Disclosure Statement No potential conflict of interest was reported by the authors.

References

Conneau A, Lample G (2019) Cross-lingual language model pretraining. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (eds) Advances in Neural Information Processing Systems, vol 32. Curran Associates, Inc.

Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805

Gao X, Shi X, Guo H, Liu Y (2020) To buy or not buy food online: The impact of the covid-19 epidemic on the adoption of e-commerce in china. PloS One 15(8):e0237900

Goudjil M, Koudil M, Bedda M, Ghoggali N (2018) A novel active learning method using svm for text classification. Int J Autom Comput 15(3):290–298

Li C, Zhan G, Li Z (2018) News text classification based on improved bi-lstm-cnn. In: 2018 9th International Conference on Information Technology in Medicine and Education (ITME), pp 890–893

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692

Ma S, Yang J, Huang H, Chi Z, Dong L, Zhang D, Awadalla HH, Muzio A, Eriguchi A, Singhal S et al (2020) Xlm-t: Scaling up multilingual machine translation with pretrained cross-lingual transformer encoders. arXiv:2012.15547

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:1301.3781

Oluk A (2020) Turkish language models. https://github.com/Loodos/turkish-language-models

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv:1908.10084

Rosario RR (2017) A data augmentation approach to short text classification. Ph.D. Thesis, UCLA

Shen Y, Zhang Q, Zhang J, Huang J, Lu Y, Lei K (2018) Improving medical short text classification with semantic expansion using word-cluster embedding. In: International Conference on Information Science and Applications. Springer, pp 401–411

Stein RA, Jaques PA, Valiati JF (2019) An analysis of hierarchical text classification using word embeddings. Inf Sci 471:216–232

Tagliabue J, Greco C, Roy J-F, Yu B, Chia PJ, Bianchi F, Cassani G (2021) Sigir 2021 e-commerce workshop data challenge. arXiv:2104.09423

Wang J, Wang Z, Zhang D, Yan J (2017) Combining knowledge with deep convolutional neural networks for short text classification. In: IJCAI, vol 350

Wang Z, Qu Z (2017) Research on web text classification algorithm based on improved cnn and svm. In: 2017 IEEE 17th International Conference on Communication Technology (ICCT), pp 1958–1961

WibowoHaryanto A, KholidMawardi E, Muljono (2018) Influence of word normalization and chi-squared feature selection on support vector machine (svm) text classification. In: 2018 International Seminar on Application for Technology of Information and Communication, pp 229–233

Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Scao TL, Gugger S, Drame M, Lhoest Q, Rush AM (2020) Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Online, pp 38–45

Xu J, Cai Y, Wu X, Lei X, Huang Q, Leung H, Li Q (2020) Incorporating context-relevant concepts into convolutional neural networks for short text classification. Neurocomputing 386:42–53

Yang L, Shijia E, Shiyao X, Yang X (2020) Bert with dynamic masked softmax and pseudo labeling for hierarchical product classification. In: Proceedings of Mining the Web of HTML-embedded Product Data Workshop (MWPD2020)

Yu H-F, Ho C-H, Arunachalam P, Somaiya M, Lin C-J (2012) Product title classification versus text classification. Csie Ntu Edu Tw:1–25

Zahera HM, Sherif MA (2020) Probert: Product data classification with fine-tuning bert model. In: Proceedings of Mining the Web of HTML-embedded Product Data Workshop (MWPD2020)

Zhang Z, Bizer C, Peeters R, Primpeli A (2020) Mwpd2020: Semantic web challenge on mining the web of html-embedded product data. In: CEUR Workshop Proceedings, vol 2720. RWTH, pp 2–18