Multi-level Deep Learning based E-commerce Product Categorization

Wenhu Yu
JD.com
China
yuwenhu@jd.com

Zhiqiang Sun
JD.com
China
sunzhiqiang3@jd.com

Haifeng Liu
JD.com
China
bjliuhaifeng@jd.com

Zhipeng Li
JD.com
China
lizhipeng@jd.com

Zhitong Zheng
JD.com
China
zhengzhitong@affiliation.org

ABSTRACT

E-commerce product categorization is an important topic, and its quality directly affects subsequent search, recommendations and related personalized services. E-commerce product classification is challenging due to the large scale and complexity of the product information and categories. In the E-Commerce Text Classification Challenge, we combine machine learning, deep learning, and natural language processing to propose a multi-level and multi-class deep learning tree method. Our method constructs multiple models based on single-label and multi-level label predictions as well as the characteristics of the product tree structure and combines the multiple models to generate a new classification model. The proposed classification model is tested on the online test dataset. The accuracy, recall, and F1 score are 0.8552, 0.8389, 0.8404 in leaderboard(Stage 1) and 0.8397, 0.8428, 0.8379 in leaderboard(Stage 2) respectively, ranking among top 3 scorers.

CCS CONCEPTS

•**Applied computing organization** → **E-commerce infrastructure**; Supervised learning by classification; •**Computing methodologies** → **Natural language processing**

KEYWORDS

Deep learning, Hierarchical search tree, Text classification

1 INTRODUCTION

The task of the big data challenge is to predict the e-commerce product category according to the given product title. The challenges of this task are as follows.

(1) The distribution of product categories is extremely unbalanced. For example, there are thousands of product titles related to some merchandise categories, but some product categories only have 1 to 2 titles in the training dataset.

(2) Offline training data and online test data are very different. The analysis shows that the training dataset and the test dataset involve about 220,000 and 580,0000 different words respectively, which means that the online test dataset contains a large number of new words.

(3) The levels of category to be predicted is complex. If a category label such as "A>B>C" is defined as the third category level, the product category level in the training dataset can reach 8.

(4) The large number of product categories to be predicted greatly increases the complexity of the classification problem. If we consider each label as a category in the training data set, there are 3008 categories in total. If we look at categories at different levels, we can reach 1600 categories at the most levels.

To address these challenges, we develop classification strategies based on the characteristics of the data. We merge the training dataset and the test dataset to construct word vectors for textual expressions so that semantic similarity can be used to process new words in the test dataset. Sampling and data enhancement techniques are used to address the unbalanced category issue. To deal with the complexity of product classification, we construct eight sample datasets according to the category hierarchy and develop two classification algorithms to build classification models for different levels and search paths using category trees.

2 RELATED WORK

Text classification is an important topic in natural language processing. It is widely used in information retrieval, search recommendation, news classification, anti-spam, public opinion analysis and other fields. A large number of text classification methods have already been pro-posed in the past. Conventional feature processing constructs TF-IDF and other features[2] . Word vector based methods can improve the performance compared with traditional methods in terms of expressing the semantic information between words[9], getting more dense vectors and reducing the complexity of extracting text features. Classification

algorithms and methods from Naive Bayes, KNN classification to more recent Fasttext and deep learning models such as TextCNN, TextRNN, VDCNN, AbLSTM, etc. have achieved better performance and accuracy.

Compared with the prior work, we evaluate the effectiveness of different feature extraction methods and classification algorithms, and then combine multiple models to develop a new product category classification method.

3 EXPERIMENTAL AND COMPUTATIONAL DETAILS

All experiments use Tensorflow and Fasttext. In the classification experiment, we divide the data into three sets: training set, verification set, and test set. The data volume are 720,000, 80,000 and 200,000 respectively. In order to evaluate the impact of each step on the final test, we first analyze the distributions of the word frequency, product label hierarchy and word count, etc. The effectiveness of different text preprocessing methods, including noun extraction, stemming and excluding stop words, and different word vector models is evaluated. We also evaluate the impact of Fasttext, TextCNN, TextRNN , directory tree and AbLSTM models. Based on the results, we choose the best models and combine them to classify the online test dataset.

3.1 Data Exploration

We first analyze the given data. The result shows in Fig1.

We find that the distributions of the title length are centered around 10 words and the length of the titles in the training dataset shows a longer-tail distribution from Fig1.b and Fig1.c. Fig1.d shows that the number of products of each category in the training set varies from 1 to 70,000, an extremely unbalanced distribution. We further study the distribution that 3rd, 4th and 5th categories account for the largest proportions of the data both in titles and subcategories from Fig1.e and Fig1.f.

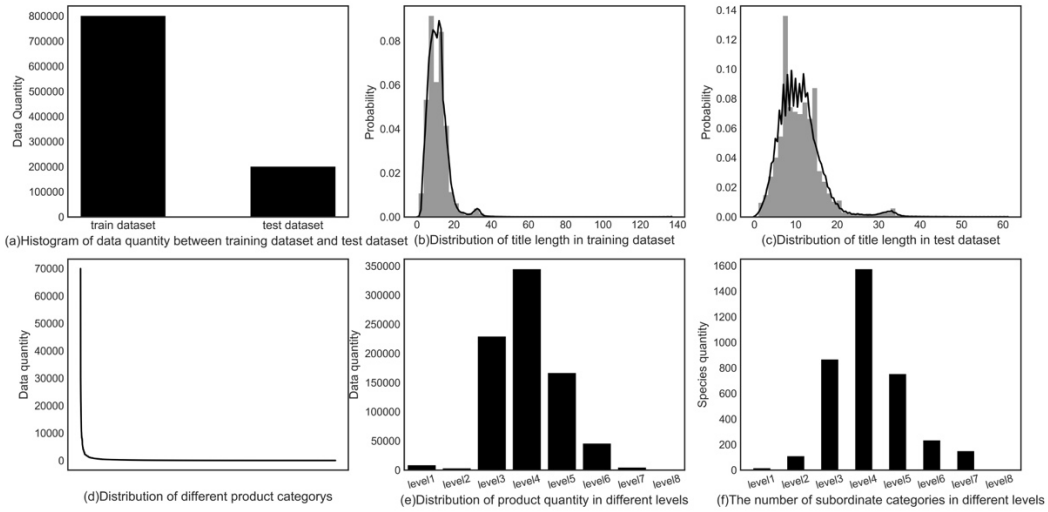

Figure 1: Fig1.a compares the data volume of the training set and the test set. The distributions of the title length in the training dataset and the testing dataset are shown in Fig1.b and Fig1.c respectively. Fig1.d depicts the number of products of each category in the training dataset. Fig1.e shows that the 3rd, 4th and 5th categories account for the largest proportions of the titles. Fig1.f shows the number of subcategories in each category level.

3.2 Data Analysis

We analyze the category labels obtained by the classification model and investigate why some categories are misclassified on the validation dataset.

The offline training dataset is divided into a standard training dataset with a size of 720,000 and a validation dataset with a size of 80,000. We set the ngram parameter to 2 and the word window length to 100. Using the Fasttext model with the standard training dataset, the model's prediction performance on the validation dataset is shown in Table1. The interval indicates the predicted probability interval. Inconsistent and Consistent indicate the number of prediction category labels that are consistent and inconsistent with the real category labels respectively. Total is the total number of titles included in the prediction probability interval.

Table 1: Probability Distribution of Predict Results

Interval	Inconsistent	Consistent	total
0~0.1	152	11	163
0.1~0.2	496	52	548
.....	.....	.....	.....
0.92~0.93	209	259	468
0.93~1	4804	60570	65374

The number of titles with a prediction probability higher than 0.93 is 65,374, accounting for 81.7% of all 80,000 titles in the validation dataset. The result is consistent basically with the calculated precision as Table2 shows. If the predicted probability is greater than 0.93, there is a high probability that the predicted label is consistent with the reference label. Otherwise, the prediction is erroneous and we need to study it further.

To analyze the erroneous results, we first obtain the data that does not match the actual prediction labels in the validation dataset, and then use the similarity method to perform an approximate match detection.

For example, for the title "Fuel Pressure Regulator Carter 404-500HP" , the Fasttext model's prediction is 2199> 661> 4498> 343 while the actual result is 2199> 4592>12. We search the most similar product titles as 2199>661>4498>343 and 2199>4592>12 in the training dataset. The most similar title as the former and the latter are "Fuel Pressure Regulator: Belt/Hex Drive Pump EFI Regulator" and "Edelbrock 1727 Fuel Pressure Regulator" respectively. This may be the cause of misclassification.

3.3 Feature Engineering

In the field of natural language processing, feature engineering includes text preprocessing, feature extraction and text expression. Feature extraction process plays a very important role in text classification. The classification task is mainly composed of two parts, converting data to features and features to classification. The former process determines the upper limit of the classification model performance.

3.3.1 Text preprocessing. We try different text preprocessing methods including removing stop words, stemming and nouns extraction to evaluate their accuracy using the Fasttext algorithm. The result on the validation dataset is shown in Table 2. We can see that the more we modify the title, the worse the resulting accuracy is. We hence decide to use the original title as input in the experiments later.

Table 2: Probability distribution of forecast results

Text Preprocessing Methods	Precision	Recall	F1 score
Origin	**0.823**	**0.823**	**0.823**
Exclude numbers	0.816	0.816	0.816
Stopwords	0.806	0.806	0.806
Stopwords +Stemming	0.797	0.797	0.797
Stopwords +Stemming +Extract nouns	0.807	0.807	0.807

3.3.2 Feature extraction ant text extraction. Text representation is a way to convert text information into machine understandable language. The traditional methods for text representation are word-bag model or vector space model. The word-bag model is characterized by high dimensionality and sparsity. It cannot express the semantic information very well. In order to perform text representation better, we extract some features to enhance text information. Common feature extraction methods include mutual information, information gain, and TF-IDF method. However, these feature extraction methods can only reflect the features of specific words, but can not express the context and semantic similarity. Semantics-based text representation transforms text into word vector. Word2vec, glove and Fasttext can perform text representation based on semantics. We use Fasttext to calculate the word vector on the 1 million dataset in our work.

3.4 Modeling

Traditional text classification methods such as naive Bayes and nearest neighbor classification do not take into account contextual correlation. Therefore, the performance is poor when used in large-scale multi-classification problems. Deep neural networks may be better because it calculates the local correlation[2-4]. In our work, we use the word vector method to do text representation and build the model using the CNN, RNN, and several other network structures.

If we consider "3292>114>123" a single label, it's a single-label classification problem. If we consider "3292>114>123" a combination of "3292", "114" and "123", it becomes a multi-label classification. We discuss the two different models next.

3.4.1 Single-lable prediction model. We initialize the word embedding matrix using the Fasttext model. The word embedding dimension is 100 and N-gram is 2. We also use TextCNN, TextRNN, AbLSTM and other models to conduct the text classification experiments [5-11]. The performance of each model is shown in Table 3. Among all the models, Fasttext and AbLSTM perform better than the others.

Table 3: Test performance of five different single models on online datasets

Methods	Precision	Recall	F1 score
Fasttext	**0.82**	**0.82**	**0.82**
TextCNN	0.76	0.75	0.74
VDCNN	0.74	0.75	0.74
TextRNN	0.73	0.74	0.73
AbLSTM	**0.83**	**0.83**	**0.82**

3.4.2 Multi-level label prediction model. If the problem is defined as a multi-level label prediction, we need to predict the category label in each level and then combine different levels of category numbers according to the category tree.

We analyze the category labels and find that a category number belongs to a single level at a time. For example, number 11 never appears under the first-level and second-level categories at the same time. Based on this observation, we generate a multi-level tree using the product category labels for the offline training data. A multi-level tree example is shown in Fig2.

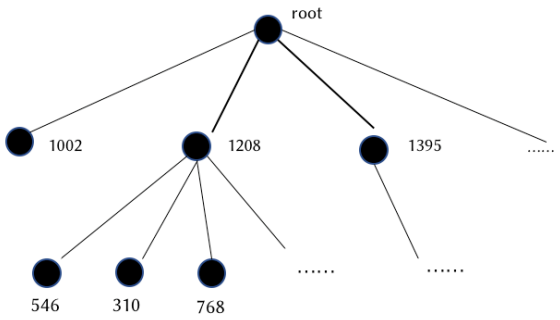

Figure 2: Diagram of category tree

Different data training sets for different category levels are generated. Since the levels are up to 8, 8 models are generated. Each model predicts a label, and searches the tree in a top-down order. The probability of a multi-level category label is the average of the probability of the labels on the search path.

Fig.3 shows the multi-level tree classification model training process. First, the original data is divided into eight sets according to different hierarchical category labels, and a classification model corresponding to per level is trained on each data set.

Fig.4 depicts the multi-level tree classification prediction process. The multi-level tree classification model consists of an input layer, a word vector layer, a classification model layer, and a tree search layer. For a new product title, according to the pre-training word vector model, the word vector layer is converted into a vector form that can be calculated by the computer, and then according to different hierarchical models, the top three most probable labels are calculated, and finally the categories are combined. In the tree, all possible paths are calculated, and the path with the highest average probability is selected as the output value.

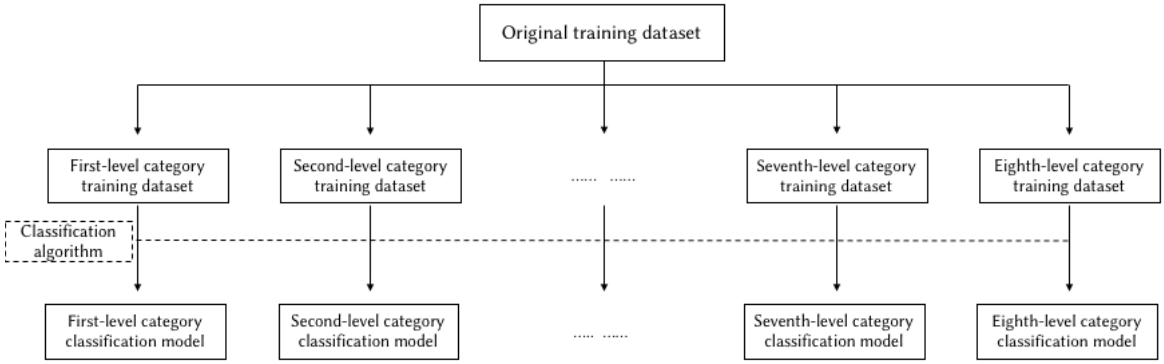

Figure 3: Training process of multi-class tree classification model

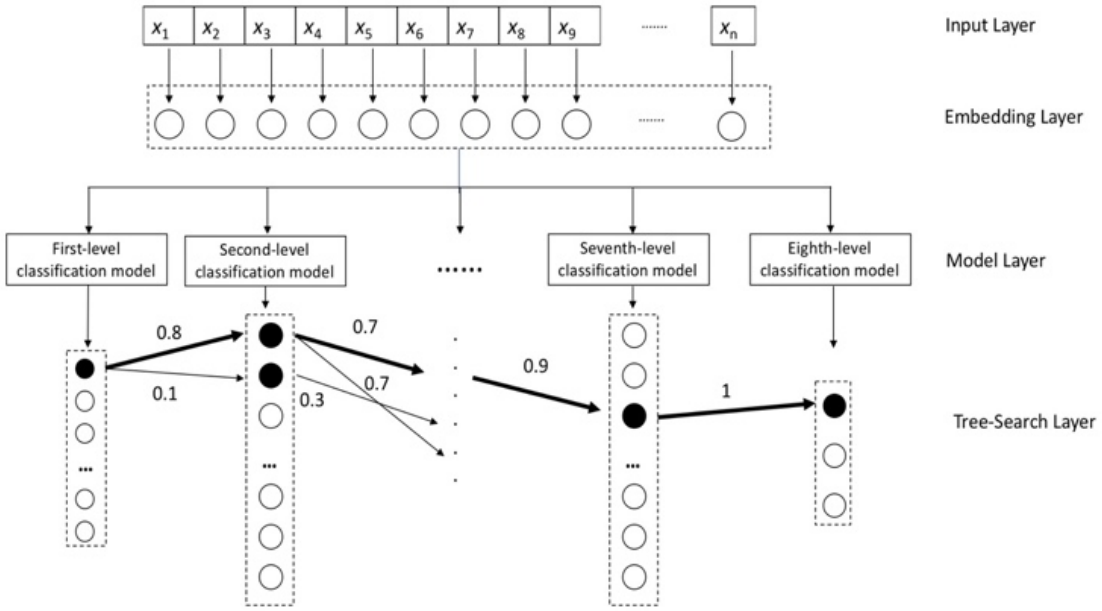

Figure 4: Predict process of multi-class tree classification model

For each test sample, assuming that the maximum level of the hierarchical tree is k , the maximum possible path is calculated as follows :

$$Path_h = [C^1_{m_1}, C^1_{m_2}, \dots C^1_{m_{k-1}}, C^1_{m_k}] \quad (1)$$

$$P_h = \prod_{m=2}^{k} P_{m-1_i} P_{m_j} \quad (2)$$

$$Path_{max} = Path_{argmax(P_h)} \quad (3)$$

where $C^1_{m_k}$ represents a random label in the k[th] category and $Path_h$ is a collection of all paths made up of different levels of category elements. P_{m-1_i} and P_{m_j} represent the probability corresponding to the i[th] label at the m-1[th] level , and the j[th] label at the m[th] level respectively. P_h is the product of the probability of each node on a path and $Path_{max}$ is the path with the highest probability .

With the previous experience of the hierarchical tree model, we try a new sample tagging method, which replaces tag with a combination of the tags of current hierarchical level and all of its superior. For example, if the predicted target is '3292>114>123', we will get '3292', '3292>114' and '3292>114>123'. In order to distinguish it from the previous hierarchical tree, we name the new tree model SP tree. With different classification algorithms, different models including Fasttext+SP TREE (FST) and AbLSTM+ SP TREE (AST) are constructed. We choose the deepest level of the label set containing the hierarchical prediction label from top to bottom, and take the label with the highest probability in the set as the prediction result. Equation (4) and (5) describe the above calculation process, where $Index_k$ represents the sequence number corresponding to the prediction result with the highest probability at the k[th] level.$Path_{argmax(P(Path_{k-1}))}$ represents the most probable prediction in the k-1[th] level. $Path_k$ denotes all predictions for the k[th] level. $Path_{kmax}$ indicates the maximum probability label of the k[th] level which contains its superior labels.

$$Index_k = argmax P(Path_{argmax(P(Path_{k-1}))} \in Path_k) \quad (4)$$

$$Path_{kmax} = Path_{Index_k} \quad (5)$$

The Fasttext and AbLSTM methods are used for classification according to the results of the single model. The models are submitted online for evaluating the test dataset. The result is shown in Table 4.

The model based on the single-label prediction achieves higher recall rates and F1 scores while the model based on multi-label prediction achieves higher precision.

Table 4: Test performance of multi-level category models on online dataset

Methods	Precision	Recall	F1 score
Fasttext+Tree(FT)	0.85	0.77	0.80
AbLSTM+Tree(AT)	0.83	0.83	0.82
Fasttext+SPtree(FST)	0.84	0.80	0.81
AbLstm+SPtree(AST)	0.84	0.81	0.81

3.5 Model fusion

For model fusion, we use both the simple voting method and the weighted voting method. The simple voting method refers to voting on the results according to multiple models, and determining the category based on the voting results. The weighted voting method refers to adding up the predicted probability values of multiple models, and choosing the prediction with the highest probability value. The results of online data test using the simple voting method and the weighted voting method are shown in Table 5.

Table 5: Test performance of different fusion strategy category prediction models on online dataset

Methods	Fusion Strategy	P	R	F1
Baseline (Fasttext)	--	0.83	0.82	0.82
Baseline+ FT+AbLSTM	Simple voting	0.83	0.81	0.82
Baseline + FT+AbLSTM+ AT	Simple voting	**0.86**	0.80	0.82
Baseline+ FT+AbLSTM	Weighted voting	**0.86**	0.82	**0.83**
Baseline + FT+AbLSTM+ AT	Weighted voting	0.85	**0.83**	**0.84**
Baseline + FT+AbLSTM+ AT+FST+AST (STAGE 1)	Weighted voting	**0.8552**	**0.8389**	**0.8404**
Baseline + FT+AbLSTM+ AT+FST+AST (STAGE 2)	Weighted voting	**0.8397**	**0.8428**	**0.8379**

4 CONCLUSION

The classification of product categories based on the product titles is an important and challenging problem. We propose a method that combines different classification models.

1. More data preprocessing such as excluding stop words, stemming, and extracting nouns can lead to worse performance. This observation suggests the use of the original titles as the input.
2. We build two types of models using single-label prediction and multi-level label prediction respectively. For single-label prediction, we use Fasttext, Text-CNN, Text-RNN, VDCNN and AbLSTM. The results show that Fasttext and AbLSTM perform better than the others. For multi-level label prediction, according to the different processing methods of sample tags, we construct hierarchical search tree model and short path tree model . For hierarchical search tree, we first extract the category tree structure in the training dataset and use different classification algorithms to predict the top three labels with the highest probability in each level. We then choose the path with the highest probability as the prediction result according to the

category tree. Compared with the hierarchical tree, the short path tree appends all parent tags to the current level tag when processing the sample tag. The model based on single-label prediction achieves higher recall rates and F1 scores while the model based on multi-label prediction gets higher precision. These results imply that the classification model can be improved by combing different models.

3. Our approach combines the results from Fasttext, AbLSTM, Fasttext-Tree(FT) , AbLSTM-Tree(AT) , Fasttext-SP-Tree(FST) and AbLSTM-SP-Tree(AST) with weighting strategy and achieves a precision, recall and F1 score of 0.8552, 0.8389, 0.8404 in leaderboard(Stage 1) and 0.8397 , 0.8428 , 0.8379 in leaderboard (Stage 2) respectively . The proposed approach could also be used for other text classification tasks such as movie, music, fresh, etc.

ACKNOWLEDGMENTS

This work was partially supported by the SIGIR eCom`18 Project "Taxonomy Classification for eCommerce-scale Product Catalogs" and by Rakuten Institute of Technology Boston (RIT-Boston).

REFERENCES

[1] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing:deep neural networks with multitask learning. *Internet Control Message Protocol (ICML)*, 160-167.

[2] Cicero Nogueria Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, 69-78.

[3] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. In *Neural computation*, 1735-1780.

[4] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP).*

[5] Pedro HO Pinheiro and Ronan Collobert. 2014. Recurrent convolutional neural networks for scene labeling. In *Internet Control Message Protocol* (*ICML*), 82-90.

[6] Karen Simonyan and Andrew Zisserma. 2015. Very deep convolutional networks for large-scare image recognition. In *ICLR* .

[7] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola , and Eduard Hovy.2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, 1480-1489.

[8] Geoffrey E.Hinton, Simon Osindero and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. In *Neural Computation*, 1527-1554.

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jegou and Tomas Mikolov.2017. FASTTEXT.ZIP:COMPRESSING TEXT CLASSIFICATION MODELS. In *ICLR*.

[10] Zichao Yang,Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola and Eduard Hovy. 2016.Hierarchical attention networks for document classification. In *Proceeding of NAACL-HLT*, 1480-1489.

[11] Alexis Conneau , Holger Schwenk, Yann Le Cun and Loïc Barrault. 2017. Very Deep Convolutional Networks for Text Classification.

[12] Takeru Miyato, Andrew M. Dai and Ian Goodfellow. 2016. Virtual Adversarial Training for Semi-Supervised Text Classification.