

Fine-Grained Product Categorization in E-commerce

Hongshen Chen
Data Science Lab,
JD.COM
Beijing, China
ac@chenhongshen.com

Jiashu Zhao
Department of Physics and Computer
Science,
Wilfrid Laurier University
Waterloo, Canada
jzhao@wlu.ca

Dawei Yin*
Data Science Lab,
JD.COM
Beijing, China
yindawei@acm.org

ABSTRACT

E-commerce sites usually leverage taxonomies for better organizing products. The fine-grained categories, regarding the leaf categories in taxonomies, are defined by the most descriptive and specific words of products. Fine-grained product categorization remains challenging, due to blurred concepts of fine grained categories (i.e. multiple equivalent or synonymous categories), instable category vocabulary (i.e. the emerging new products and the evolving language habits), and lack of labelled data. To address these issues, we propose a novel Neural Product Categorization model—NPC to identify fine-grained categories from the product content. NPC is equipped with a character-level convolutional embedding layer to learn the compositional word representations, and a spiral residual layer to extract the word context annotations capturing complex long range dependencies and structural information. To perform categorization beyond predefined categories, NPC categorizes a product by jointly recognizing categories from the product content and predicting categories from predefined category vocabularies. Furthermore, to avoid extensive human labors, NPC is able to adapt to weak labels, generated by mining the search logs, where the customers’ behaviors naturally connect products with categories. Extensive experiments performed on a real e-commerce platform datasets illustrate the effectiveness of the proposed models.

CCS CONCEPTS

• **Information systems** → **Information systems applications**;
Web searching and information discovery; **Web log analysis**;
Data mining.

KEYWORDS

Neural Network; Product Categorization

ACM Reference Format:

Hongshen Chen, Jiashu Zhao, and Dawei Yin. 2019. Fine-Grained Product Categorization in E-commerce. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*, November 3–7, 2019,

*Corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11...\$15.00
<https://doi.org/10.1145/3357384.3358170>

Product Title	Categories
TomCare Cube Storage 6-Cube	• Cube Storage
Metal Wire Cube Storage Cube	• Closet Organizer
Closet Organizer DIY Storage	• Storage Grids
Grids Stackable Storage Bins	• Storage Bins
Wire Cubes Bookcase Modular	• Bookcase
for Home Office, Black	• Bookshelf

Table 1: An example of a product and its fine-grained categories, category in bold font doesn't appear in the title.

Beijing, China. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3357384.3358170>

1 INTRODUCTION

Nowadays, online shopping has greatly changed the way people shop. It enables customers to browse numerous products in a faster and more convenient way through searching and recommendation. For major e-commerce platforms, billions of products are in stock, and accurate product understanding has become crucial towards better online shopping experience. Among different aspects of product understanding, product categorization is a fundamental and essential way to manage products. Automatic product categorization [11] draws many interests for alleviating human labors.

An e-commerce system usually leverages product taxonomies [9] to hierarchically organize products. The upper level categories in product taxonomies are relatively coarse and manually predefined, while the lower level categories are usually more specific and fine-grained. Particularly, the fine-grained categories, typically regarding the leaf categories in taxonomies, are defined by the most descriptive and specific words of products, rather than general categories. For example in Table 1, “closet organizer” is a more fine-grained category than “organizer” or “furniture”, which brings a more precise understanding about the product. Not only for elaborated product organization, such fine-grained categories play more important roles in both query understanding and user profiling than coarse categories.

However, comparing with coarse grained categorization, the fine-grained product categorization remains more challenging: 1) *Blurred concepts*: To precisely define the fine-grained categories is non-trivial, since the fine-grained categories often include multiple partially overlapping equivalents. 2) *Instable categories*: Accompanying with emerging new product launches in the e-commerce platform, new fine-grained categories have to be introduced. In addition, some fine-grained product categories might also keep

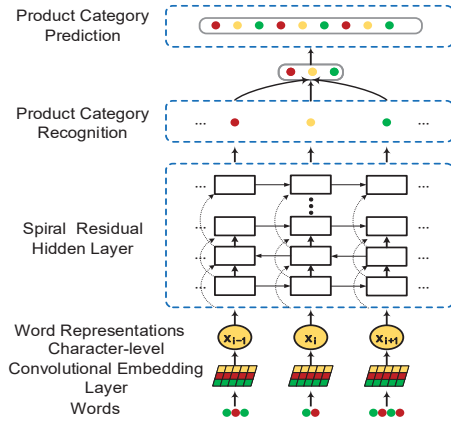


Figure 1: A sketch of neural product categorization model

evolving (i.e. new added, updated, deleted). 3) *Insufficient labelled data*: As is known to all, it is prohibitively expensive for human experts to manually label millions of products with tens of thousands of categories. Moreover, fine-grained categories are particularly difficult to label due to its difficult in definition.

To address the aforementioned issues, in this paper, we propose a Neural Product Categorization model (NPC) specialized in fine-grained product categorization. The proposed NPC aims to generate the product categories from the product contents, e.g. titles, attributes, descriptions etc.. In order to better understand the word patterns, NPC firstly transforms a word into its corresponding vector representation by utilizing a character-level convolutional layer and then maps the sequences of word representations into the product context annotations back and forth through a spiral residual hidden layer. Finally, to perform categorization beyond predefined categories, NPC identifies product fine-grained categories by jointly recognizing categories from the product content and predicting categories from predefined category vocabulary together. To avoid extensive human efforts, the training data is generated by mining the search logs, where the customers' behaviors naturally connect products with categories. Extensive experiments show that the proposed NPC is able to outperform the state-of-the-arts noticeably.

2 NEURAL PRODUCT CATEGORIZATION

The neural product categorization model takes in the product content and output the fine-grained product categories. The overall framework of the model is shown in Figure 1.

Character-level Convolutional Embedding Layer. To enable the model distilling the lexical compositional features, we employ a character-level convolutional embedding layer to map a word. Given a word consisting of a sequence of characters, n -gram feature sets are generated by applying the convolutional filters over the character embedding sequence, which is then followed by a *high-way network* [14] to simulate more complex compositional features. For a product content $T = w_1, w_2, \dots, w_l$ consisting of l words, we apply the character-level convolutional embedding layer for each word, resulting with a word representation sequence.

Model	Top 5			Top 3			Top 1		
	P@5	R@5	F@5	P@3	R@3	F@3	P@1	R@1	F@1
NPC-word embedding	49.54	42.66	45.84	59.58	32.36	41.94	74.6	13.96	23.52
NPC-BiLSTM	55.63	45.71	50.18	63.75	33.97	44.32	75.11	14.05	23.67
NPC-Reco	58.14	37.45	45.55	64.16	31.39	42.16	77.44	14.49	24.41
NPC-Pred	50.91	44.84	47.68	61.1	33.06	42.9	75.36	14.1	23.75
NPC	57.18	45.42	50.62	64.95	33.93	44.58	75.8	14.18	23.9

Table 2: Comparisons of the model variants (%).

Spiral Residual Hidden Layer. We employ a spiral residual LSTM, which is a multi-layer deep bi-directional stacking LSTM with residual connections [6] connecting each layer, to compose back and forth sophisticated features. It sequentially transforms the word representations into the product content annotations h^t .

2.1 Categorization

Product Category Recognition. We notice the phenomenon that some fine-grained categories are already listed in the product content, especially for new emerging out-of-vocabulary categories. Therefore, a straightforward way to categorize the product is to recognizing them from the product content. Recognition probability measures the probability of a word in the product content acting as a category, and is defined as: $p_{\Gamma}(w) = \begin{cases} \frac{1}{Z_1} \sum_{w_i=w} e^{\Psi_{\Gamma}(w_i)}, & w \in T \\ 0, & w \notin T \end{cases}$,

where Ψ_{Γ} is scoring function and w_i stands for the i -th word in the content T , $Z_1 = \sum_{w_i \in T} e^{\Psi_{\Gamma}(w_i)}$ is the normalization term. The scoring function $\Psi_{\Gamma}(w_i)$ is defined as: $\Psi_{\Gamma}(w_i) = \mathbf{w}^T \rho_{\Gamma}(\mathbf{h}_i^t)$, where ρ_{Γ} is non-linear transformation function, \mathbf{h}_i^t is the i -th product content representation, and \mathbf{w} is a one-hot indicator vector of word w_i . Note that, We can also enumerate phrases as candidate product categories.

Product Category Prediction. Similar to previous product classification approaches [1], we also categorize the products through product classification on predefined categories. It first performs a thorough understanding of the product content through an attention mechanism, and then predicts the product categories in the predefined category vocabulary. The product categories prediction probability can be formalized as: $p_{\Omega}(w) = \begin{cases} \frac{1}{Z_2} e^{\Psi_{\Omega_V}(w)}, & w \in \Omega_V \\ 0, & w \notin \Omega_V \end{cases}$,

where Ω_V is the product category vocabulary, $Z_2 = \sum_{w \in \Omega_V} e^{\Psi_{\Omega_V}(w)}$ is the normalization term. The scoring function $\Psi_{\Omega_V}(w)$ is defined as: $\Psi_{\Omega_V}(w) = \mathbf{w}^T \rho_V(s)$, where ρ_V is non-linear transformation function, and s is the context vector. The context vector s depends on the sequence of the context annotations \mathbf{h}^t , and is computed through a product content attention: $s = \sum_{i=1}^l \alpha_i \mathbf{h}_i^t$, the weight α_i of each annotation equals to the corresponding product category prediction probability.

Joint Product Categorization by Recognition and Prediction. In our model, we identify the product categories by jointly recognizing categories from the product content and predicting categories from predefined category vocabulary. The joint product categorization probability is computed as: $p(w) = p_{\Omega_V}(w) + p_{\Gamma}(w)$.

To train model parameters, a negative log-likelihood objective is exploited, which is defined as: $L(\Theta) = -\frac{1}{N} \sum_n \log(p(w^{(n)}|T^{(n)}))$, where the superscript (n) indicates the index of one product content-category pair.

Model	Failure Rate	#	P	R	F
Literal Matching	0.85%	1.0	55.38	10.36	17.46
BiLSTM-CRF	18.47%	1.0	75.7	11.68	20.24
			P@1	R@1	F@1
DeepCN	-	8.34	65.53	12.26	20.65
NPC-Reco	0.48%	3.75	77.44	14.49	24.41
NPC-Pred	-	8.95	75.36	14.1	23.75
NPC	-	6.72	75.8	14.18	23.9

Table 3: Comparison with previous works. # denotes the average number of specialised categories.

3 DATA PREPARATION

Mining Search Logs For Training. Manually annotating the product categories for training is extremely labor-intensive, since a product may be associated with multiple categories, and the existing categories typically amount to tens of thousands, let alone the newly emerging categories. More troubling, figuring out the exact concept scope of the fine-grained categories is extraordinarily difficult. Therefore, we propose to generate weak labels by mining the search logs, where the customers’ behaviors naturally connect products and categories. Intuitively, in search, when a customer issuing a “category query”, he/she may click the most relevant products to that category. It implies that the frequently clicked products are more likely to belong to the querying category. In detail, we collect 6 months search logs where the queries match the category names in a given category dictionary consisting of 62,000 items. For each query, the top most 10,000 frequently clicked products are collected to obtain the (product, category) pairs as training data. The total records in the training set is amount to 20,214,661.

Validation and Testing Data. We manually annotate a validation set, and a testing set to evaluate the performance of the approaches. The validation/test set is generated from the entire instock products of the e-commerce site. To guarantee the products with different exposure rates fairly evaluated, we sample the test sets at random by stratified sampling based on search engine traffics [7]. We divide all the stocking products into ten buckets, according to the exposure rates in search traffics, where each bucket is of the approximately same number of clicks, and draw about 170 products from each bucket resulting with 1651 products. For each product, the categories generated from all approaches are mixed and presented to domain experts in a random order for fair manual annotation.

4 EXPERIMENTS

4.1 Experimental Details

We evaluate the effectiveness of the fine-grained neural product categorization model with respect to the following evaluation metrics: Accuracy, Precision, Recall, and F-measure. The dimensions of both the character embeddings and the word embeddings are set to 100. The size of LSTM units is also set to 100. The maximum length of each product content is set to 28. For the character-level convolutional embedding layer, the maximum length of a word set to 4, which is determined by the existing fine-grained category set. We stack 6 layers of bi-directional residual LSTM units for spiral residual hidden layer. Training is done on every sample through Adam optimization algorithm [10]. Given a product, the neural

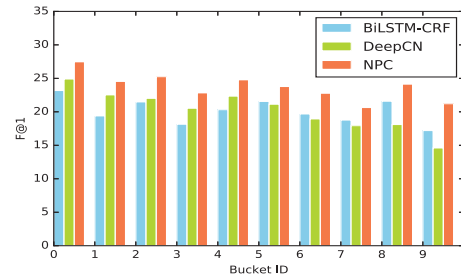


Figure 2: F@1 on different buckets. A larger bucket ID indicates a lower product exposure rate.

product categorization model predicts a score for each of the candidate fine-grained product categories. The categories are ranked by the predicted scores, and we keep the categories with scores higher than the given threshold as the output of the model. For all the approaches discussed in this paper, the validation set is utilized to set a threshold that can achieve the best F-measure at top 5.

4.2 Model Discussions

Effects of Character-level Convolutional Embedding Layer. The words in the product content are encoded to the embeddings through the character-level convolutional embedding layer. To validate its effectiveness, we replace character-level convolutional embedding layer with the conventional word embedding matrix (NPC-word embedding). As shown in table 2, we observe that the performance for NPC-word embedding is much lower than NPC with respect to all the metrics. It illustrates that the superiority in exploiting the characters, especially for words with similar prefix or suffix.

Effects of Spiral Residual Hidden Layer. The spiral residual hidden layer composes the context annotations back and forth by stacking multiple forward and backward LSTM units. It captures the long range dependencies and structural information in the product content. When comparing a variant utilizing a single bi-directional LSTM (NPC-BiLSTM in table 2) with NPC, we observe that the spiral residual hidden layer effectively improves the performance which demonstrates the advances of the deep residual structure.

Product Categorization Recognition vs Prediction. We further analyze the performance of recognizing the fine-grained product categories from titles (NPC-Reco) and predicting the product categories according to the content context vector (NPC-Pred). In table 2, we can see when only recognizing the fine-grained product categories from product content, the precision are relatively higher than product category prediction, whereas, the recalls of the product category prediction is superior to only recognizing the categories from content. When jointly incorporating both approaches, we achieve the best performance in terms of the F-measures. This indicates that recognizing the product categories from content ensures the precision, meanwhile, predicting the product categories can help the model discover the categories that are not present in the product content.

4.3 Results

The fine-grained product categorization aims to excavate all the equivalent categories of a product, while most previous works are

focused on predefined category classification. We here compare our model with three representative methods:

Literal Matching, where the product category is recognized from the product content by matching the exact words. If a word in the content matches an item in the given product category dictionary, then it is designated as a category of the product.

BiLSTM-CRF is favored in many content understanding tasks [12]. **DeepCN** [5] is our implementation of the state-of-the-art in large scale item categorization by product classification. The results are presented in Table 3.

When comparing the failure rate, we find that, when only recognizing the product categories from contents (as in Literal Matching, BiLSTM-CRF, and NPC-*Reco*), the models fails to complete the product categorization occasionally. They tend to not recognize any word as product category in some cases. Whereas, product categorizing by classification (DeepCN, NPC-*Pred*, NPC) would not suffer from similar problem.

In terms of the number of generated categories, Literal Matching and BiLSTM-CRF output approximately only one category for each product, while product category prediction models (DeepCN and NPC-*Pred*) generate more categories. The number of categories generated by NPC is larger than product category recognition models and fewer than product category prediction models. This illustrates that product category prediction tends to produce more fine-grained categories than product category recognition.

With respect to the evaluation metrics, we list the top one performance of the product category prediction models to perform a fair comparison. BiLSTM-CRF model performs better than Literal Matching. Literal Matching is prone to the fake categories. The precision of DeepCN is slightly lower than BiLSTM-CRF, however, the recall of DeepCN is much higher. All three variants of our model perform much better than all the baselines, while NPC-*Reco* achieves the best performance. This shows that our model is more adept at recognizing the product categories from product content.

Performance on Different Buckets. We further study the performance of the model with respect to different product exposure rates. From Figure 2, we notice that, our model consistently performs better than the baselines on all buckets. Products in bucket 0 are of relatively higher exposure rates than all other buckets. Both DeepCN and NPC surpasses BiLSTM-CRF by a large margin in bucket 0. As the exposure rates decreases, the performance of DeepCN gradually declines and BiLSTM-CRF overtakes DeepCN. The products in bucket 9 are long-tail products. NPC still performs much better than DeepCN and BiLSTM-CRF. We speculate that for high exposure rates product, product category prediction is a better choice, while for long-tail products, recognizing category from the product content is a superior option.

5 RELATED WORK

Product categorization problem in E-commerce is usually defined as classification of products into an existing list tens to thousands of categories [2, 3, 8]. Since the training data, usually consist of product titles and descriptions, are too sparse to be directly adopted as features, standard classification approaches utilized various types of manual features. [13] employed SVM in item categorization with unigram words, morphological features, quantity features and

pattern features. A linear model is used to learn the multi-class classification problem with LDA and embedding features for Yahoo! products in [11]. [4] developed a two-level ensemble approach utilizing path-wise, node-wise and depth-wise classifiers based on document embedded vectors. [1] ensembled deep belief nets and KNN learned from semantic hashed product title and description to classify products. [15] illustrated that an attentional CNN with both word and character encodings increases the training efficiency, while [5] utilized a RNN. These approaches are only suitable for well-defined product classification with a limited number of categories. In fine-grained product categorization, each product belongs to multiple equivalent categories. We generate the product categories by jointly recognizing and predicting the product categories.

6 CONCLUSION

In this paper, we propose a neural product categorization model to generate the fine-grained categories from product contents by jointly recognizing from the product content and predicting from a predefined product category vocabulary. Moreover, to alleviate human labors, the proposed model is able to effectively adapt weak labels which is generated by mining search logs. Experiments on large-scale datasets from a real e-commerce platform demonstrate the effectiveness of the model. We'd like to perform more detailed comparisons and analysis in the future work.

REFERENCES

- [1] Ali Cevahir and Koji Murakami. 2016. Large-scale Multi-class and Hierarchical Product Categorization for an E-commerce Giant. In *COLING*. 525–535.
- [2] Eli Cortez, Mauro Rojas Herrera, Altigran S da Silva, Edleno S de Moura, and Marden Neubert. 2011. Lightweight methods for large-scale product categorization. *Journal of the American Society for Information Science and Technology* 62, 9 (2011), 1839–1848.
- [3] Ying Ding, M Korotkiy, Borys Omelayenko, V Kartseva, V Zykov, Michel Klein, Ellen Schulten, and Dieter Fensel. 2002. Goldenbullet: Automated classification of product data in e-commerce. In *BIS*.
- [4] Vivek Gupta, Harish Karnick, Ashendra Bansal, and Pradhuman Jhala. 2016. Product classification in e-commerce using distributional semantics. *arXiv* (2016).
- [5] Jung-Woo Ha, Hyuna Pyo, and Jeonghee Kim. 2016. Large-scale item categorization in e-commerce using multiple recurrent neural networks. In *SIGKDD*. ACM, 107–115.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.
- [7] Dustin Hillard, Stefan Schroedl, Eren Manavoglu, Hema Raghavan, and Chirs Leggetter. 2010. Improving ad relevance in sponsored search. In *WSDM*. ACM, 361–370.
- [8] Sheng Huang, Xinlan Liu, Xueping Peng, and Zhendong Niu. 2012. Fine-grained product features extraction and categorization in reviews opinion mining. In *ICDM Workshops*. IEEE, 680–686.
- [9] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluís Garcia-Pueyo. 2012. Supercharging recommender systems using taxonomies for learning user purchase behavior. *VLDB* 5, 10 (2012), 956–967.
- [10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv* (2014).
- [11] Zornitsa Kozareva. 2015. Everyone likes shopping! multi-class product categorization for e-commerce. In *NAACL*. 1329–1333.
- [12] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL*. Association for Computational Linguistics, Berlin, Germany, 1064–1074. <http://www.aclweb.org/anthology/P16-1101>
- [13] Dan Shen, Jean David Ruvini, Manas Somaiya, and Neel Sundaresan. 2011. Item categorization in the e-commerce domain. In *CIKM*. ACM, 1921–1924.
- [14] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv* (2015).
- [15] Yandi Xia, Aaron Levine, Pradipto Das, Giuseppe Di Fabbrizio, Keiji Shinzato, and Ankur Datta. 2017. Large-Scale Categorization of Japanese Product Titles Using Neural Attention Models. In *EACL*, Vol. 2. 663–668.