Finding Camouflaged Needle in a Haystack? Pornographic Products Detection via Berrypicking Tree Model

Guoxiu He*
Wuhan University
Wuhan, China
guoxiu.he@whu.edu.cn

Yangyang Kang
Alibaba Group
Hangzhou, China
yangyang.kangyy@alibaba-inc.com

Zhe Gao
Alibaba Group
Hangzhou, China
gaozhe.gz@alibaba-inc.com

Zhuoren Jiang
Sun Yat-sen University
Guangzhou, China
jiangzhr3@mail.sysu.edu.cn

Changlong Sun
Alibaba Group
Hangzhou, China
changlong.scl@taobao.com

Xiaozhong Liu†
Indiana University Bloomington
Bloomington, United States
liu237@indiana.edu

Wei Lu†
Wuhan University
Wuhan, China
weilu@whu.edu.cn

Qiong Zhang
Alibaba Group
Sunnyvale, United States
qz.zhang@alibaba-inc.com

Luo Si
Alibaba Group
Seattle, United States
luo.si@alibaba-inc.com

ABSTRACT

It is an important and urgent research problem for decentralized eCommerce services, e.g., eBay, eBid, and Taobao, to detect illegal products, e.g., unclassified pornographic products. However, it is a challenging task as some sellers may utilize and change camouflaged text to deceive the current detection algorithms. In this study, we propose a novel task to dynamically locate the pornographic products from very large product collections. Unlike prior product classification efforts focusing on textual information, the proposed model, **B**erryP**I**cking T**R**ee Mo**D**el (**BIRD**), utilizes both product textual content and buyers' seeking behavior information as berrypicking trees. In particular, the BIRD encodes both semantic information with respect to all branches sequence and the overall latent buyer intent during the whole seeking process. An extensive set of experiments have been conducted to demonstrate the advantage of the proposed model against alternative solutions. To facilitate further research of this practical and important problem, the codes and buyers' seeking behavior data have been made publicly available[1].

CCS CONCEPTS

• **Information systems** → **Spam detection**; **Query log analysis**; • **Computing methodologies** → *Neural networks*;

KEYWORDS

spam detection, information seeking, user behavior, query log analysis, deep neural network

ACM Reference Format:
Guoxiu He, Yangyang Kang, Zhe Gao, Zhuoren Jiang, Changlong Sun, Xiaozhong Liu, Wei Lu, Qiong Zhang, and Luo Si. 2019. Finding Camouflaged Needle in a Haystack? Pornographic Products Detection via Berrypicking Tree Model. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331197

1 INTRODUCTION

In the past decade, decentralized eCommerce services, e.g., eBay, eBid, and Taobao, are challenging traditional monopolistic intermediaries. Through these eCommerce ecosystems, everyone could easily become an e-merchant, and eCommerce platforms provide extra incentives to sellers with convenient marketing and buyer-access channels and resources. Unfortunately, like other online environments, illegal contents, e.g., unclassified erotica or pornographic products, can pollute the cybermarkets. While most of these eCommerce platforms don't have their own inventory, the illegal products, uploaded by some problematic sellers, can spread more easily than ever before[2]. Such risk can be quite harmful to both buyers and cybermarkets.

With the local training dataset, pornographic product detection can be a straightforward binary classification problem, i.e., machine learning or deep learning model [27] along with text features extracted from product contents such as titles or descriptions. However, this strategy doesn't work online well because sellers could easily hack the detection system (shown in Figure 1). For instance, when the current learning algorithm finds a seller is listing a pornographic product, the seller could easily change the product title or description and release it again with a new seller/product ID, which means pornographic products and their sellers hide like

[*] Work done as an intern at Alibaba Group.
[†] Corresponding authors.
[1] https://github.com/GuoxiuHe/BIRD

[2] https://www.cbsnews.com/news/ebay-selling-recalled-products-illegal/

SIGIR '19, July 21–25, 2019, Paris, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331197

chameleons in the eCommerce ecosystem while traditional learning algorithms can hardly detect them effectively. On the other hand, as eCommerce providers cannot save enough new training data in a short time window, the learning algorithm can hardly capture this dynamic for efficient illegal product detection. More specifically, the word distribution of pornographic and normal products in local (historical) dataset can be quite different which is good for training, while the gap is significantly shortened in the real (current) online testing environment (also depicted in Figure 4). It is clear that sellers are trying to change the product content to deceive the eCommerce platform.

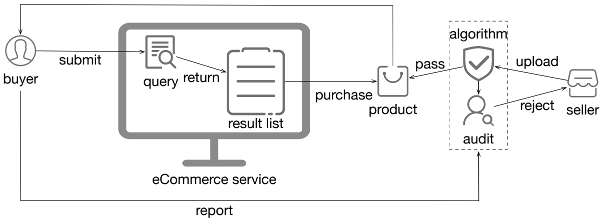


Figure 1: The Detection System in an eCommerce Service.

From the seller viewpoint, however, employing camouflaged content can be a double-edged sword. While they can successfully escape from the traditional detection algorithm, buyers will have to spend extra time and efforts to locate the pornographic products from very large collections by using more sophisticated information seeking strategies. For instance, when buyers search for *porn video USB*, which is an illegal query, via Taobao, they won't get any result. In order to locate what they are looking for, buyers will have to update the query content a few times and also check/consume the retrieved products carefully. From the information seeking perspective, this kind of *'buyer search pornographic products behavior'* can be interpreted via the classical *berrypicking* model from Marcia Bates [3], which is depicted as a tree structure shown in Figure 2.

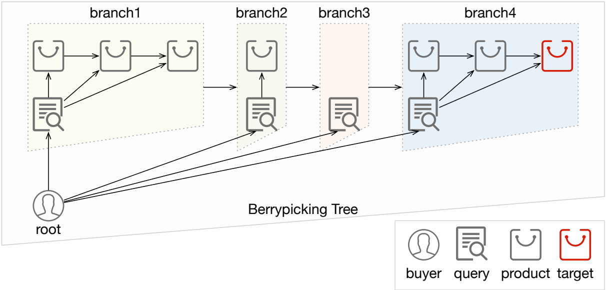


Figure 2: Berrypicking Tree is consist of a root, which denotes the buyer, as well as several branches. Each branch is composed of a query and clicked products sequence. All branches happen one by one.

In this study, we propose a novel method for Pornographic Products Detection by leveraging deep-information seeking behavior mining. More specifically, the Berrypicking Tree is proposed to encapsulate various kinds of seeking information, e.g., queries sequence and clicked products sequence. Furthermore, unlike prior studies, we propose an innovative **B**erryP**I**cking T**R**ee Mo**D**el (**BIRD**), which captures the hidden semantics and latent seeking intent of the tree by utilizing a new recurrent model and a pruning mechanism. Experiments based on a large real-world dataset indicate that the proposed BIRD outperforms all the baseline solutions. In particular, while sellers can use various strategies to escape from classical detection methods, they CANNOT directly change buyers' seeking behavior.

Briefly, the main contributions of this work can be summarized as follows:

- We raise the question of automatically detecting pornographic products in an eCommerce ecosystem, which, to the best of our knowledge, is the first inquiry effort to this problem.
- We propose an innovative algorithm, BIRD, to locate the pornographic products by leveraging the massive buyers' information seeking data. In particular, the berrypicking tree with pruning is used to encapsulate the buyers' seeking behavior, and the hidden semantics and latent buyer intent are encoded for effective detection.
- In order to prove the hypothesis, we collect a large product plus buyers' seeking behavior dataset from one of the world largest eCommerce sites. Extensive online experimental results show that the proposed model can successfully identify the pornographic products and outperform a number of alternative baselines. And we make the codes and dataset publicly available.

2 BERRYPICKING TREE MODEL

In this section, we propose a novel model, BerrypIcking TRee MoDel (BIRD), depicted in Figure 3, which enables automatic illegal products detection in an eCommerce ecosystem by exploring in-depth buyers' information seeking behavior.

2.1 Overview

In this work, we locate the illegal product from very large collections by utilizing its associated buyers' seeking behavior sessions. A product $\hat{p}$ in the collection is represented as a set $(Sess(\hat{p}), y)$, where $Sess(\hat{p})$ is the session log and $y \in Y = \{0, 1\}$ represents the product label of whether this product is illegal. More specifically, the session log $Sess(\hat{p})$ encapsulates the berrypicking tree structure, as shown in Figure 2, which is represented as a sequence:

$$Sess(\hat{p}) = [(q_1, P_1), \cdots, (q_t, P_t), \cdots, (q_{n_s}, P_{n_s})], \tag{1}$$

where q_t is the $t-$th query that buyer submits before purchasing $\hat{p}$, and P_t denotes the sequence of products that buyer clicked one by one after browsing the results of q_t. In each session, $\hat{p}$ is the last product in P_{n_s}, and n_s is the total number of queries that buyer tries. In addition, each query and product (content) is consist of a sequence of words and each word w is mapped to a d_e dimensional word embedding $\boldsymbol{w} \in \mathbb{R}^{d_e}$ by looking up the embedding matrix $\boldsymbol{E} \in \mathbb{R}^{n_w \times d_e}$, where n_w is the vocabulary size.

The goal of BIRD is to explore the textual semantics and the latent buyer's search intent from the session log as well as estimate the probability of whether the target product is illegal based on the

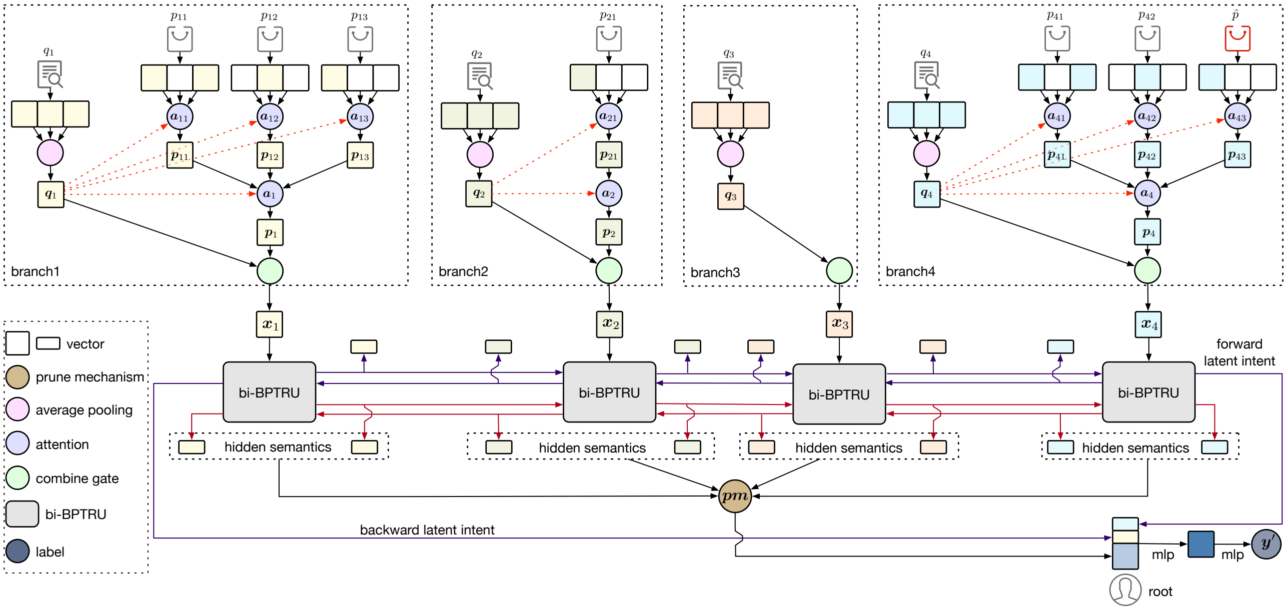


Figure 3: BerryPIcking TRee MoDel (BIRD)

seeking session by learning the parameters θ:

$$y' = argmax_{y \in Y} Pr(y|Sess(\hat{p}), \theta). \quad (2)$$

2.2 Branch Representation

In order to make the model more flexible and robust in the online testing for word distribution gap reason mentioned in Section 1, we project the word embedding lookup table $\boldsymbol{E}$ with a transfer matrix $\boldsymbol{W}_e \in \mathbb{R}^{d_e \times d_e}$ to $\tilde{\boldsymbol{E}}$:

$$\tilde{\boldsymbol{E}} = \boldsymbol{E} \cdot \boldsymbol{W}_e. \quad (3)$$

As Figure 2 depicts, each branch of the berrypicking tree, including a query and the associated sequence of the clicked products, represents a proactive search effort from a buyer. As query can be short and it is not advisable to model query with a complex model, we employ average pooling to characterize the semantics of query $\boldsymbol{q}_t \in \mathbb{R}^{d_e}$ as follows:

$$\boldsymbol{q}_t = AvgPooling(lookup(\tilde{\boldsymbol{E}}, q_t)) . \quad (4)$$

Text information of product can be noisier compared with query. Meanwhile, as aforementioned, sellers may purposely use the camouflaged content to mislead the eCommerce platform and hack the classification algorithm. In order to cope with this problem, we employ a product-query attention model to highlight the useful information in the product content, whereas the content of $t-$th product $p_{tt} \in P_t$ is represented as $\boldsymbol{p}_{tt} \in \mathbb{R}^{d_e}$ based on the attention $\boldsymbol{a}_{tt}$ by query q_t:

$$\begin{aligned} \boldsymbol{a}_{tt} &= softmax(lookup(\tilde{\boldsymbol{E}}, p_{tt}) \cdot \boldsymbol{q}_t) , \\ \boldsymbol{p}_{tt} &= \boldsymbol{a}_{tt}^T \cdot lookup(\tilde{\boldsymbol{E}}, p_{tt}) . \end{aligned} \quad (5)$$

Then, the sequence of products P_t is represented as a matrix $\boldsymbol{P}_t = [\boldsymbol{p}_{t1}, \cdots, \boldsymbol{p}_{tt}, \cdots]$.

Note that, for eCommerce service, buyers are highly likely to click the interested, curious, or suspected pornographic products following the returned order from search engine especially on mobile devices, which relies on the query-product relevance score. Hence, we investigate the products sequence that clicked after the submitted query by a query attention method as follows:

$$\begin{aligned} \boldsymbol{a}_t &= softmax(\boldsymbol{P}_t \cdot \boldsymbol{q}_t) , \\ \boldsymbol{p}_t &= \boldsymbol{a}_t^T \cdot \boldsymbol{P}_t , \end{aligned} \quad (6)$$

where $\boldsymbol{p}_t \in \mathbb{R}^{d_e}$ is the final representation of the sequence of products in the $t-$th branch.

In order to get the branch representation, we propose a combine gate to determine how much information from the query and the product content will be used. Formulas are shown bellow:

$$\begin{aligned} \boldsymbol{c}_t &= \sigma(\boldsymbol{W}_c \cdot (\boldsymbol{q}_t \oplus \boldsymbol{p}_t) + \boldsymbol{b}_c) , \\ \boldsymbol{x}_t &= (1 - \boldsymbol{c}_t) \odot \boldsymbol{p}_t + \boldsymbol{c}_t \odot \boldsymbol{q}_t , \end{aligned} \quad (7)$$

where $\boldsymbol{W}_c \in \mathbb{R}^{d_e \times 2 \cdot d_e}$ is a weight matrix, $\boldsymbol{b}_c \in \mathbb{R}^{d_e}$ is a bias, σ denotes the sigmoid function, $\oplus$ denotes the concatenate operation, and $\odot$ represents the element-wise multiplication.

Finally, all branches in the berrypicking tree are represented as $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{n_s}]$.

2.3 Tree Representation

As the buyer is the root in the berrypicking tree, besides the semantics hidden in the sequence of branches, we also explore the latent purchase intent of buyer (e.g., an normal user or an illegal product seeker) among all the information seeking efforts in the tree. However, existing recurrent models, such as LSTM, GRU, and SRU can only represent one of them. To this end, in this paper, we propose a novel recurrent unit, Berrypicking Tree Recurrent Unit (BPTRU), to capture the hidden semantics and latent buyer intent simultaneously in the berrypicking tree as follows:

$$\begin{aligned}
\boldsymbol{z}_t &= \sigma(\boldsymbol{W}_z \cdot \boldsymbol{x}_t + \boldsymbol{v}_z \odot \boldsymbol{h}^1_{t-1} + \boldsymbol{b}_z) \,, \\
\boldsymbol{r}_t &= \sigma(\boldsymbol{W}_r \cdot \boldsymbol{x}_t + \boldsymbol{v}_r \odot \boldsymbol{h}^2_{t-1} + \boldsymbol{b}_r) \,, \\
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}^1_i \cdot \boldsymbol{h}^1_{t-1} + \boldsymbol{W}^2_i \cdot \boldsymbol{h}^2_{t-1} + \boldsymbol{b}_i) \,, \\
\tilde{\boldsymbol{h}}^1_t &= \boldsymbol{i}_t \odot \boldsymbol{h}^2_{t-1} \,, \\
\tilde{\boldsymbol{h}}^2_t &= \boldsymbol{i}_t \odot \boldsymbol{h}^1_{t-1} \,, \\
\boldsymbol{h}^1_t &= tanh(\boldsymbol{z}_t \odot \boldsymbol{h}^1_{t-1} + (1 - \boldsymbol{z}_t) \odot (\boldsymbol{W}^1_h \cdot \boldsymbol{x}_t) + \tilde{\boldsymbol{h}}^1_t) \,, \\
\boldsymbol{h}^2_t &= tanh(\boldsymbol{r}_t \odot \boldsymbol{h}^2_{t-1} + (1 - \boldsymbol{r}_t) \odot (\boldsymbol{W}^2_h \cdot \boldsymbol{x}_t) + \tilde{\boldsymbol{h}}^2_t) \,,
\end{aligned} \tag{8}$$

where $\boldsymbol{x}_t$ denotes the current representation of the branch, $\boldsymbol{h}^1_{t-1}$ and $\boldsymbol{h}^2_{t-1}$ represent the previous hidden semantics and previous latent intent respectively, $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}^1_h, \boldsymbol{W}^2_h \in \mathbb{R}^{d_e \times d_r}$ and $\boldsymbol{W}^1_i, \boldsymbol{W}^2_i \in \mathbb{R}^{d_r \times d_r}$ are weight matrices, $\boldsymbol{v}_z, \boldsymbol{v}_r \in \mathbb{R}^{d_r}$ are weight vectors, and $\boldsymbol{b}_z, \boldsymbol{b}_r, \boldsymbol{b}_i \in \mathbb{R}^{d_r}$ are biases.

In BPTRU, $\boldsymbol{z}, \boldsymbol{r} \in \mathbb{R}^{d_r}$ are two hidden gates to determine the combination of the previous hidden state (latent intent) and the current branch. More importantly, we employ an interact gate $\boldsymbol{i} \in \mathbb{R}^{d_r}$ to supplement the joint information of previous hidden semantics and previous latent intent in the current branch.

In this work, we take advantage of the bidirectional model for berrypicking tree representation learning as follows:

$$\begin{aligned}
\overrightarrow{\boldsymbol{H}^1}, \overrightarrow{\boldsymbol{H}^2} &= \overrightarrow{BPTRU}(\boldsymbol{X}) \,, \\
\overleftarrow{\boldsymbol{H}^1}, \overleftarrow{\boldsymbol{H}^2} &= \overleftarrow{BPTRU}(\boldsymbol{X}) \,,
\end{aligned} \tag{9}$$

where $\overrightarrow{\boldsymbol{H}^1}, \overleftarrow{\boldsymbol{H}^1} \in \mathbb{R}^{n_s \times d_r}$, and $\overrightarrow{\boldsymbol{H}^2}, \overleftarrow{\boldsymbol{H}^2} \in \mathbb{R}^{n_s \times d_r}$ are the forward and backward hidden semantics and latent intent of all branches, respectively.

In general, we can utilize the average pooling to obtain the final semantic representation for hidden semantics $\boldsymbol{h}^1 \in \mathbb{R}^{2 \cdot d_r}$:

$$\begin{aligned}
\boldsymbol{H}^1 &= [\overrightarrow{\boldsymbol{H}^1}, \overleftarrow{\boldsymbol{H}^1}] \,, \\
\boldsymbol{h}^1 &= AvgPooling(\boldsymbol{H}^1) \,.
\end{aligned} \tag{10}$$

Differ from the hidden semantics, latent buyer intent is updated by every proactive seeking effort dynamically. Hence we pick up the last step as the final latent intent representation $\boldsymbol{h}^2 \in \mathbb{R}^{2 \cdot d_r}$:

$$\boldsymbol{h}^2 = last(\overrightarrow{\boldsymbol{H}^2}) \oplus last(\overleftarrow{\boldsymbol{H}^2}) \,. \tag{11}$$

In the end, the tree is represented as $\boldsymbol{h} \in \mathbb{R}^{4 \cdot d_r}$:

$$\boldsymbol{h} = \boldsymbol{h}^1 \oplus \boldsymbol{h}^2 \,. \tag{12}$$

2.4 Pruning Mechanism

User's behavior, in the eCommerce environment, can be somehow noisy. For instance, in a 2-hour window, buyer's search and browsing behavior may focus on multiple information needs, e.g., looking for normal products and also a pornographic product, which might pollute the target berrypicking tree for illegal product detection. In this study, we propose a pruning mechanism to filter the noisy branches from the berrypicking tree automatically. There is no doubt that the target (purchased) product (e.g., the illegal one) exists in the last branch, so we project and weight the hidden semantics from all the prior branches with respect to the last branch on the tree:

$$\begin{aligned}
\boldsymbol{H}^{1l} &= copy(last(\boldsymbol{H}^1)) \,, \\
\boldsymbol{pm} &= softmax(\sigma(similar(\boldsymbol{H}^{1l}, \boldsymbol{H}^1))) \,, \\
\boldsymbol{h}^{1*} &= \boldsymbol{pm}^T \cdot \boldsymbol{H}^1 \,,
\end{aligned} \tag{13}$$

where the cosine similarity is used as *similar*, and $\boldsymbol{pm}$ denotes the weight distribution. The irrelevant branches will be ignored after sigmoid followed by softmax, and $\boldsymbol{h}^{1*} \in \mathbb{R}^{2 \cdot d_r}$ represents the hidden semantics which can replace the general one mentioned above.

With the pruning mechanism, the tree is represented as $\boldsymbol{h'} \in \mathbb{R}^{4 \cdot d_r}$:

$$\boldsymbol{h}^* = \boldsymbol{h}^{1*} \oplus \boldsymbol{h}^2 \,. \tag{14}$$

2.5 Training

By using the berrypicking tree encoding $\boldsymbol{h}$ or $\boldsymbol{h}^*$ described above, two Multilayer Perceptrons (MLPs) are applied to generate the probability y' of whether the target product is illegal:

$$y' = \sigma(\boldsymbol{v}_o^T \cdot relu(\boldsymbol{W}_m \cdot \boldsymbol{h} + \boldsymbol{b}_m) + b_o) \,, \tag{15}$$

where $\boldsymbol{W}_m \in \mathbb{R}^{d_m \times 4 \cdot d_r}$ denotes the weight matrix, $\boldsymbol{v}_o \in \mathbb{R}^{d_m}$ denotes the weight vector, and $\boldsymbol{b}_m \in \mathbb{R}^{4 \cdot d_r}$ and b_o represent the biases.

The BIRD, overall, is an end-to-end deep neural network, which can be trained by using stochastic gradient descent (SGD) algorithms, such as Adam [25]. More implementation details will be given in Section 3.

For each product, there can be multiple buyer seeking behavior sessions. In the online environment, we calculate how many sessions are detected as illegal via BIRD for each product and rank all products as the final result for the eCommerce illegal product detection.

3 EXPERIMENTS

In this section, we collect a large buyers' behavior dataset and conduct extensive experiments to evaluate the proposed BIRD against a number of alternative solutions, including state-of-the-art text classification models and base models built on berrypicking tree.

3.1 Data Collection

The Pornographic Products Detection Dataset (PPDD) is collected and constructed from Taobao[3], one of the world largest decentralized eCommerce platforms, containing product text content and related buyers' seeking behavior session logs for each pornographic product. PPDD will enable us to provide insights on this problem, and train and evaluate the proposed model.

We first located 3,002 pornographic products, which were accumulated from Aug. 1st, 2016 to Sep. 1st, 2018. Most of these products were either reported by buyers or detected by eCommerce illegal product detection professionals. For each product, 2-hour buyers'

[3] https://www.taobao.com/

Table 1: Details of the PPDD including number of products, related sessions, and total log records.

	Local			Online Test 1			Online Test 2			
	normal(-)	porn.(+)	sum	normal(-)	porn.(+)	sum	normal(-)	porn.(+)	sum	sum
products	398,644	3,002	401,646	2,262	685	2,947	795	411	1,206	405,799
sessions	2,068,683	114,250	2,182,933	4,489	1,649	6,138	1,587	1,236	2,823	2,191,894
records	15,226,790	1,068,527	16,295,317	22,272	7,686	29,958	6,971	5,441	12,412	16,337,687

seeking behavior logs before purchasing were collected for berrypicking tree construction. The seeking logs include search queries and clicked products, and Figure 2 depicts the structure of each search session.

Meanwhile, to train the machine learning models, we random sampled the normal products from the nearly 1 billion products, limited in the categories that the pornographic products occurred. Buyers' seeking behavior logs were extracted same as what the pornographic products did. Finally, there are 401,646 products and 2,182,933 sessions in the local set.

The challenge of this work is that sellers can consistently update the product content to hack the classical detection algorithms. Therefore, we will need to evaluate the proposed algorithm compared with baselines in the real and most recent online eCommerce platform. From the top 3 suspected product categories, we collected nearly 7 million popular products and also extracted the associated text content and seeking data. For fair comparison, we repeated the experiment twice, denoted as **Online Test 1** (from Nov. 3rd, 2018 to Nov. 16th, 2018) and **Online Test 2** (from Dec. 3rd, 2018 to Dec. 16th, 2018). For each online test, the proposed model and baseline models generated a candidate pornographic products pool. All the candidate products were examined by experts in Taobao. For some highly suspected but camouflaged products, experts needed to communicate with the sellers to decide the real categories. Finally, there are 2,947 products and 6,138 sessions in the set of online test 1, and there are 1,206 products and 2,823 sessions in the set of online test 2. More details of PPDD are shown in Table 1.

3.2 Classification Baselines

We employ the following baselines (see Table 3) for product classification based on product content, including traditional approaches, word embedding [34] based shallow neural networks, deep learning based models and recent state-of-the-art (SOTA) models:

SVM [15]: Support Vector Machine (SVM) is a strong and robust baseline model, when training data is somehow sparse, based on bag-of-word features.

SWEM [38]: Simple Word Embedding Model (SWEM) is a simple but efficient model based on word embeddings with pooling mechanisms such as max pooling, average pooling, concatenate pooling, and hierarchical pooling.

SimpleCNN [24]: This is a simple CNN model with average pooling using different kernels. There are 7 kinds of kernels whose widths are from 1 to 7 and each has 100 different ones.

RNN: We both try Bidirectional RNN (BiRNN) and BiRNN with attention mechanism using LSTM [16], GRU [7], or SRU [30] as the RNN cell in this experiment respectively. We set word embedding dimension as 200, RNN dimension as 50, and dense dimension as 100.

CNNLSTM [49]: This baseline utilizes the CNN to encode local information and then uses a LSTM model to capture the dependency information.

BiGRUCNN [43]: The motivation of this more recent baseline is similar to CNNLSTM. The difference is that the BiRNN is used to figure out dependency information in front of the CNN.

DPCNN [21]: Deep Pyramid CNN (DPCNN) is a low-complexity word-level deep CNN architecture for text categorization that can efficiently represent long-range dependency in text.

Transformer [41]: This is the state-of-the-art model to encode the deep semantic information via self-attention mechanism[4].

Unless otherwise specified, for stable performance and fair comparison, we use the recommended or default hyper-parameter settings by the authors for all the baseline models.

3.3 Baselines with Different Seeking Features

In order to verify the usefulness of the information seeking features and the effects of the proposed model, in this part, we explore various kinds of features along with the straightforward classification model, SVM. The comparison results are available in Table 4.

For each product in the collection, as aforementioned, we extracted all the related products and queries in the search session logs. Then, there are 5 kinds of feature combinations for SVM training as follows: target product content only (content), target product content with last query (content+query), queries sequence (queries), all products sequence (products), target product content with queries sequence (content+queries), all product content sequences with all queries sequence (products+queries). Finally, we use these features to train SVM models and evaluate them on the two online test sets.

3.4 Base Models via Berrypicking Tree

To the best of our knowledge, this work is the first effort to investigate pornographic product detection based on the information seeking behavior logs. For validating the effects of the proposed BIRD, we compare it with a number of base models built on berrypicking tree (also see Table 5).

Average Pooling (AvgPool): Word embedding average pooling is a strong baseline and should be paid more attention demonstrated by SWEM [38]. For this approach, we apply average pooling to every related product and query first. Then, for each branch, we use average pooling to characterize the clicked products sequence, followed by concatenating query representation and products sequence representation for the branch representation. Finally, average pooling is applied again to the sequence of branches.

Pooling with Attention (AttenPool): As a more advanced model compared with AvgPool, we utilize the attention mechanism

[4] https://github.com/tensorflow/models/tree/master/official/transformer

to estimate the semantic representation of the product sequence as described in Section 2.2.

Pooling with Attention plus Combine Gate (AttenPoolGate): Instead of concatenating for two parts in each branch, we utilize the gate to determine the combination of them as described in Section 2.2.

Standard Recurrent Models (LSTM, GRU, SRU): Instead of average pooling in AttenPoolGate, we apply LSTM, GRU, and SRU to encode the sequence of the branches.

The Proposed Recurrent Model (BPTRU): We employ the proposed BPTRU described in Section 2.3 to replace the standard recurrent model for an enhanced berrypicking tree encoding. For this method, we validate two more variants by characterizing the hidden semantic and latent seeking intent. BPTRU.sub1 denotes that we apply average pooling to all steps of BPTRU output, and BPTRU.sub2 denotes that we just select the last representation from BPTRU.

Query Only Based Models: In order to verify the usefulness of the explicit efforts of the buyer, we simplify the berrypicking tree by removing all leaves (clicked products in the seeking sessions). The average pooling, LSTM, GRU, SRU, and BPTRU are employed to investigate the performance.

BIRD: As described in Section 2, by using the proposed BIRD, queries and related products information are encoded along with the BPTRU and pruning mechanism, which has the potential to penalize the noisy branches to optimize the final information seeking representation for illegal product detection.

3.5 Evaluation Metrics

For training, F1 score is used to evaluate the performance in local evaluating and testing set according to pornographic **session**. For online testing, all evaluation and empirical analyses are reported by precision (P), recall (R), F1 score (F1), F2 score (F2), Average Precision (AP), and Normalized Discounted Cumulative Gain (NDCG) [18] in the light of pornographic **product**. More importantly, since the pornographic product is a smaller set than normal set, number of recall and order of the rank list is more important than precision in practical using. Therefore we use R, F1, F2, AP, and NDCG as the major indicators to evaluate the models performance and robustness. Furthermore, the statistical significance is conducted via the student t-test with *p-value*$< 10^{-4}$.

3.6 Experiment Settings

We divide PPDD into training, validation, and testing sets at a ratio of 14: 3: 3. Word vocabulary is chosen to cover 98% words in the local dataset. Since the pornographic set is a very small set in PPDD, we oversample them in local training set to a more balance size. Furthermore, the length of product content, the length of products sequence, and the length of query are padded to the max length in each batch respectively. Note that Chinese tokenization has been done in the released PPDD, so there is no need for more efforts. For recurrent models built on berrypicking tree, the dimension of embedding and MLP is set to 32, the dimension of recurrent is set to 16, the batch size is set to 32, and the keep dropout rate is set to 0.75. For other models built on berrypicking tree, the dimension of embedding and MLP is set to 64, batch size is set to 16, and the keep dropout rate is set to 0.75 too. The learning rate and training epochs are set to 0.002 and 5 respectively for all models.

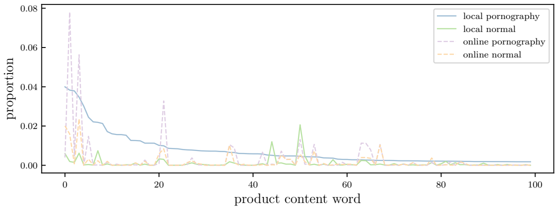


Figure 4: Product Content Statistics in PPDD: word distributions among two categories in local and online sets for product contents. Note that the online test set is consist of the two online test sets. X-axis is the top 100 words in the local pornographic set, and Y-axis is the proportion of each word.

Table 2: KL Divergence about product contents among two categories in local and online sets.

	LN-LP	ON-OP	LN-ON	LP-OP
product	2.4515	0.3317	0.0802	2.6292

4 RESULTS AND ANALYSIS

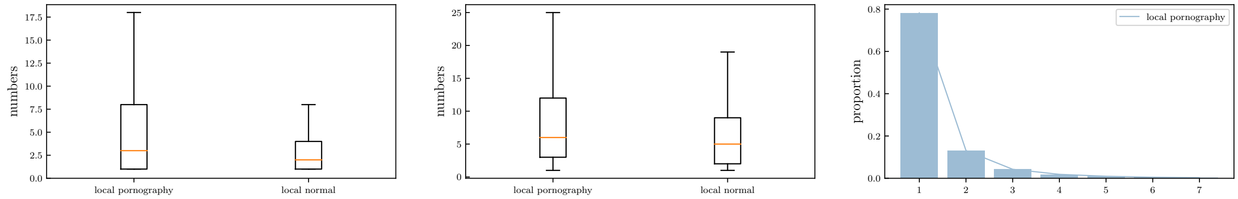
In this section, we provide a detailed analysis on data statistics and experimental results to show more insights of this problem and the proposed model.

4.1 Data Statistics

4.1.1 Product Content Analysis. From the content viewpoint, as Figure 4 depicts, pornographic product word distribution presents quite differently in local and online sets. As Table 2 shows, product content in the local normal set (LN) and local pornographic set (LP) are quite different, which leads to a high KL-Divergence, while the online pornographic (OP) product content is somehow similar with the online normal (ON) ones which results in a low KL-Divergence.

It is clear that, compared with the normal products, sellers are trying to provide the camouflaged content to protect their pornographic products, and they also change the content from time to time. This problem can challenge the traditional classification models built mainly based on product content.

4.1.2 Interaction Analysis of Local Set. In this paper, the BIRD is proposed based on the buyers' seeking behavior information, and we hypothesize that buyers may need to spend more efforts to find the pornographic products in the large product collection. Figure 5 (a) and (b) prove that buyers need to send more queries, and browse and click more products, to locate the target pornographic product compared with the normal product hunting behavior. It is obvious that compared with normal eCommerce product retrieval, pornographic product search is more like an exploratory seeking task. Meanwhile, as depicted in Figure 5 (c), almost 80% of buyers only purchase the pornographic product once. That is to say, pornographic product search can be an ad-hoc information need, and most buyers may not purchase such products very often. As a result,



(a) numbers of queries in each session (b) numbers of records in each session (c) number of times buyers purchase products

Figure 5: Interaction Statistics between Buyers and Products in Local Set of PPDD, in which (a) and (b) are the comparison for number of queries and records distribution in each session between the local normal and pornography set, and (c) is the distribution of number of times that buyer purchase products.

Table 3: Experimental Results of Performance Comparison With Text Classification Baselines.

	Val(%)	Test(%)	Online Test 1(%)						Online Test 2(%)					
Model	F1 Score		P	R	F1	F2	AP	NDCG	P	R	F1	F2	AP	NDCG
SVM	92.62	92.68	53.39	9.20	15.69	11.02	5.84	15.00	61.43	10.46	17.88	12.54	7.50	17.71
SWEMAvg	91.12	91.59	48.46	9.20	15.46	10.98	5.00	14.30	64.71	13.38	22.18	15.91	8.99	20.56
SWEMMax	90.19	90.85	46.24	12.55	19.75	14.70	6.82	17.39	53.62	18.00	26.96	20.76	12.22	25.40
SWEMHiera	88.71	89.59	43.43	**17.37**	**24.82**	**19.74**	8.68	**22.48**	48.82	**20.19**	**28.57**	**22.88**	11.67	**26.85**
SWEMConcat	90.31	90.94	47.51	12.55	19.86	14.72	6.96	17.51	54.84	16.55	25.42	19.23	11.29	23.90
SimpleCNN	92.68	92.94	48.53	9.64	16.08	11.47	5.53	14.59	62.79	13.14	21.73	15.61	10.40	21.35
BiLSTM	92.78	93.15	**64.77**	8.32	14.75	10.08	5.95	14.17	67.27	9.00	15.88	10.89	7.30	16.46
BiGRU	91.76	92.94	59.14	8.03	14.14	9.71	5.46	13.84	68.33	9.98	17.41	12.03	8.19	17.79
BiSRU	91.51	91.86	46.81	9.64	15.98	11.45	6.19	15.65	65.69	16.3	26.12	19.19	**12.40**	24.59
BiLSTMAtten	92.65	92.85	55.07	5.55	10.08	6.76	3.57	10.39	58.54	5.84	10.62	7.12	4.39	12.18
BiGRUAtten	92.47	92.85	60.67	7.88	13.95	9.54	5.42	13.64	**73.68**	10.22	17.95	12.35	8.47	18.06
BiSRUAtten	92.52	92.65	50.43	8.47	14.50	10.16	5.01	13.32	60.92	12.9	21.29	15.31	9.21	20.59
CNNLSTM	92.79	**93.35**	49.12	4.09	7.55	5.01	2.41	8.37	54.55	5.84	10.55	7.11	4.34	12.15
BiGRUCNN	**92.80**	93.15	61.25	7.15	12.81	8.69	4.80	12.71	70.91	9.49	16.74	11.48	8.24	17.42
DPCNN	91.31	91.33	5.86	13.43	21.85	15.88	**8.93**	19.85	61.86	14.60	23.62	17.23	10.68	22.54
Transformer	90.69	90.88	44.14	9.34	15.42	11.09	4.96	14.32	59.79	14.11	22.83	16.66	10.26	22.00

we can hardly use the buyer ID to directly locate the pornographic products. Furthermore, due to the sparsity of the interaction matrix between pornographic products and buyers, we need to employ more sophisticated features extracted from seeking behavior logs.

Based on these evidences, compared with the content information, buyers' search behavior can provide more important tips for pornographic product detection.

4.2 Experimental Results and Analysis of Text Classification Baselines

Table 3 presents the experimental results of all text classification baseline models mentioned in Section 3.2. All the SWEMs and CNN models implement decent results, while SWEMHiera is superior than other baselines for most of the metrics. Unfortunately, as more sophisticated baselines, transformer and RNN variants don't perform well in the online testing. This result shows that intuitive features, such as bag-of-words, can be more useful than deep semantic mining for product content encoding (may be due to the training data sparseness). In particular, RNN variants can be limited by getting a high P by sacrificing R.

More importantly, all the baselines reach good performance to distinguish pornographic products from normal ones in the historical local dataset, but drop significantly in the current online testing environments. That is because the product content is camouflaging and changing as aforementioned in Figure 4 and Table 2.

We also find that SVM is a competitive model in the baseline table compared with deep learning methods. Training data sparseness could be the reason. In the real eCommerce environment, it can be hard to collect massive pornographic products for training.

4.3 Behavior Features Exploration

In this part, we explore the performance of different seeking behavior features, mentioned in Section 3.3. SVM is employed as the testing method, and the results are presented in Table 4.

Experimental results tell that the features extracted from the seeking sessions, e.g, queries and clicked products, can be very helpful for pornographic product detection. In contrast, content only and content plus the last query achieve the best results in local set, but they are not successful in the online testing. In other words, sellers are able to escape from those content-dependent algorithms.

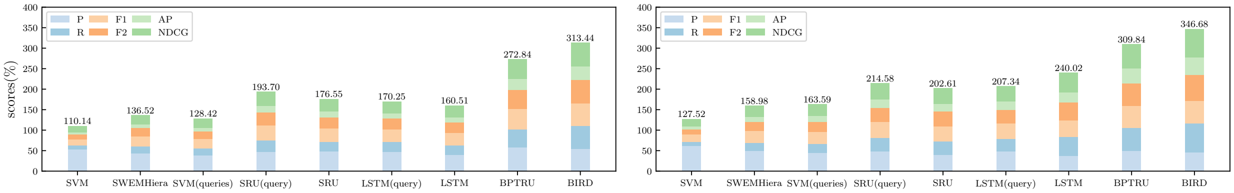
However, when applying the seeking behavior features into the SVM model, the performance is not as good as expected due to the

Table 4: Experimental Results of Performance Comparison with Different Features Combinations.

	Val(%)	Test(%)	Online Test 1(%)						Online Test 2(%)					
Feature	F1 Score		P	R	F1	F2	AP	NDCG	P	R	F1	F2	AP	NDCG
content	**92.62**	92.68	53.39	9.20	15.69	11.02	5.84	15.00	61.43	10.46	17.88	12.54	7.50	17.71
content+query	92.58	**92.71**	**73.44**	6.86	12.55	8.38	5.83	13.15	**80.00**	9.73	17.35	11.81	8.72	17.89
queries	75.20	75.26	38.70	**16.50**	**23.13**	**18.63**	**9.07**	**22.39**	44.61	**22.14**	**29.59**	**24.62**	**13.61**	**29.02**
products	86.88	86.60	58.82	11.68	19.49	13.91	9.02	18.71	68.89	15.09	24.75	17.88	12.04	23.48
content+queries	89.68	89.57	65.22	8.76	15.44	10.59	6.81	15.06	65.85	13.14	21.91	15.64	10.58	21.47
products+queries	86.88	86.67	56.78	9.78	16.69	11.72	7.36	16.41	61.70	14.11	22.97	16.69	10.47	22.14

Table 5: Experimental Results of Performance Comparison with Base Models Built on Berrypicking Tree.

	Val(%)	Test(%)	Online Test 1(%)						Online Test 2(%)					
Model	F1 Score		P	R	F1	F2	AP	NDCG	P	R	F1	F2	AP	NDCG
Avg(query)	70.55	72.33	53.64	8.61	14.84	10.35	5.86	14.65	51.58	11.92	19.37	14.09	7.74	18.16
AvgPool	69.30	71.83	51.89	8.03	13.91	9.66	5.47	13.96	56.12	13.38	21.61	15.79	9.52	21.02
AttenPool	81.81	83.03	58.22	12.41	20.46	14.73	8.57	18.10	46.05	17.03	24.87	19.49	10.90	24.25
AttenPoolGate	**83.56**	**86.92**	**66.49**	18.83	29.35	21.98	13.98	26.17	**67.11**	24.82	36.23	28.40	19.54	33.45
GRU(query)	72.69	74.10	51.35	24.96	33.60	27.82	14.91	30.54	49.81	32.36	39.23	34.80	22.12	39.49
GRU	69.53	75.62	38.10	21.02	27.09	23.09	10.78	25.56	35.03	36.74	35.87	36.39	17.98	40.21
LSTM(query)	73.12	74.29	47.48	23.36	31.31	26.00	13.32	28.78	48.28	30.66	37.50	33.07	20.22	37.61
LSTM	73.99	75.41	39.33	23.94	29.76	25.97	12.64	28.87	37.23	45.74	41.05	43.74	23.92	48.34
SRU(query)	70.83	71.15	46.71	29.05	35.82	31.43	16.48	34.21	48.54	32.36	38.83	34.67	21.10	39.08
SRU	74.32	74.65	47.58	24.38	32.24	27.01	14.88	30.46	39.71	33.33	36.24	36.24	18.40	38.69
BPTRU(query)	70.77	72.48	43.82	30.51	35.97	32.48	16.73	35.08	50.16	38.69	43.68	40.54	25.64	44.87
BPTRU.sub1	74.06	74.36	44.36	25.26	32.19	27.64	13.95	30.37	45.45	49.88	47.56	48.93	27.13	52.08
BPTRU.sub2	72.99	73.41	52.37	43.50	47.53	45.03	25.31	46.62	43.60	58.88	50.10	55.03	34.37	60.09
BPTRU	74.37	74.78	57.63	44.09	49.96	46.26	27.17	47.73	49.26	56.93	52.82	55.21	36.09	59.53
BIRD	70.56	71.04	53.48	**57.23**	**55.29**	**56.44**	**33.45**	**57.55**	45.74	**70.56**	**55.50**	**63.65**	**41.81**	**69.42**



(a) Online Test 1

(b) Online Test 2

Figure 6: Performance Comparison among the Proposed BIRD and Several Representative Models in terms of All Metrics.

mission of the behavior structure information and the confusion of all types of features. We can conclude that, when the buyer seeking behavior data is complex and different types of data may have strong dependence, we need more sophisticated model to characterize the berrypicking process.

4.4 Detailed Analysis of Models on Berrypicking Tree

To the best of our knowledge, this work is the first effort to investigate the illegal products detection via the buyers' seeking behavior session logs from eCommerce services. Therefore, we compare the proposed BIRD with a number of base models, described in Section 3.4, along with berrypicking tree. The comparison results are shown in Table 5. More details about comparison among the proposed BIRD and several representative models are visualized in Figure 6.

In the online testing, we find the proposed BIRD significantly ($p < 0.0001$) outperforms all of the baseline models, with or without buyer behavior data, for nearly all of the metrics except for Precision, which proves the effectiveness of the proposed berrypicking tree representation, semantics plus buyer-intent encoding, and the pruning mechanism. Meanwhile, except for the simplest models, such as AvgPool and AttenPool, all models built on berrypicking tree are superior than text classification baselines via product content, which validates our initial hypothesis that buyers' information seeking behavior can be very useful for pornographic products detection.

From model perspective, AttenPool achieves a better performance than AvgPool, and AttenPoolGate is superior than AttenPool, which demonstrates the usefulness of the query attentive mechanism on product content and the combine gate proposed for query and clicked products sequences. Overall, recurrent models perform better, which indicates the deep information hidden in the branches sequence can be useful for this task. In particular, BPTRU, encoding the hidden content semantics and latent buyer intent together, significantly ($p < 0.0001$) outperforms the standard recurrent models. Performance comparison among BPTRU, BPTRU.sub1, and BPTRU.sub2 demonstrates that using the last state to maintain the long range buyer intent plus semantics from all the tree branches can be an effective strategy for berrypicking tree representation. In addition, the result of simplified berrypicking tree, mentioned in Section 3.4, is not good especially for BPTRU, even though these features can enhance the SVM outcomes. This evidence tells that the clicked products sequence can enhance the seeking behavior encoding. Furthermore, there is a significant ($p < 0.0001$) improvement after applying pruning mechanism on the BPTRU, proving the usefulness of cutting the noisy branches for berrypicking tree quality enhancement.

5 RELATED WORKS

This work is related to the research on spam and intrusion detection, information seeking, log analysis, and neural text representation.

5.1 Spam and Intrusion Detection

Recently, spam detection, in eCommerce service [36, 45, 47], community question answering (CQA) [1, 12], and online social networks [28, 29], has attracted many researchers' attention. It might be difficult to distinguish the spam from the normal content based on the syntactic features, whose distribution is abnormal [11]. With manually labeled training examples, they further train a supervised learning model to detect spam content. It is found that the spam content is highly related to several features to spammers' behavior. [31] and [39] use a graph model to detect spam content or activity on a large labeled dataset. Social networks have attracted much attention for improving spam content and spammer detection by investigating individual-based and group-based user behavior [5, 46]. There are also many works on intrusion detection by using behavior analysis [37]. For illegal products detection in an eCommerce environment, where social network and traditional spammer information are not available, it is important to propose novel features and models to address this new problem.

5.2 Information Seeking and Log Analysis

Information seeking is the process or activity of attempting to obtain information [26]. For exploratory information seeking, users may suffer from search uncertainty, e.g., they need more knowledge about the search keywords in the relevant document [44]. A number of information behavior theories, which seek to understand the process that surround information seeking, such as Zipf's Principle of Least Effort [17], Brenda Dervin's Sense Making [9], and Elfreda Chatman's Life in the Round [6], from other disciplines have been applied in investigating an aspect or whole process of information seeking [23]. Query and session logs are often employed to investigate information seeking questions [10]. In this work, we explore the hidden semantics and latent buyer intent by mining the buyer seeking behavior logs organized via berrypicking model initially introduced by Marcia Bates [3].

5.3 Neural Text Representation

As method section shows, in this work, we utilize deep neural networks and semantic/tree representation learning for product classification. Existing efforts mainly focus on the application of LSTM [16, 40], GRU [7, 8], SRU [30], and CNNs [13, 21, 22, 24, 48] based on word embeddings [32, 34] drawing on the main idea of either language model [4, 33, 35] or joint representation learning [19, 20]. All these models have demonstrated impressive results in NLP applications. The attention mechanism proposed by Bahdanau et al. [2] is used to select the reference words in original language in encoder for words in foreign language in decoder before translation. Many previous works have shown that the performance of deep neural networks can be improved by attention mechanism. For example, in attention based RNN models [40, 42], the final semantic representation of the target sentence is aggregated from the weighted hidden state to enhance the long dependency information. In addition, self-attention mechanism with position embedding characterizes [41] the mutual relationship between one and others as dependency to capture the semantic encoding information. There are some other works that combine RNN and CNN for text representation to classification [14, 43, 49]. Inspired by existing studies, we propose a new recurrent model and a special attention mechanism to investigate the buyer seeking behavior sequence.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we introduce a novel task to dynamically locate the pornographic products from very large product collections in the decentralized eCommerce ecosystem. Unlike prior product classification efforts, the proposed **B**erryP**I**cking T**R**ee Mo**D**el (**BIRD**) employs complex buyers' seeking behavior logs along with berrypicking tree representation learning. Three lines of experimental results indicate that the proposed BIRD significantly outperforms other strong baselines, which proves the importance of the buyer seeking behavior data, the efficiency of the berrypicking tree, the usefulness of the proposed BPTRU, and the effects of the pruning mechanism to encode the hidden branches' semantics and latent buyer intent for pornographic product classification. More importantly, sellers can hardly hack the proposed BIRD, because they cannot directly change buyers' behavior and the associated berrypicking trees. We also make our codes and buyers' seeking behavior data publicly available to motivate other scholars to future investigate this important but underestimated problem. In the future, we will enhance the model by using other types of buyer information, e.g., products dwell time and query similarities across different sessions, which can potentially improve the performance.

7 ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (71473183, 61876003) and Fundamental Research Funds for the Central Universities (18lgpy62).

REFERENCES

[1] Prudhvi Ratna Badri Satya, Kyumin Lee, Dongwon Lee, Thanh Tran, and Jason Jiasheng Zhang. 2016. Uncovering fake likers in online social networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2365–2370.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations* (2015), 1–15.

[3] Marcia J Bates. 1989. The design of browsing and berrypicking techniques for the online search interface. *Online review* 13, 5 (1989), 407–424.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[5] Cheng Cao, James Caverlee, Kyumin Lee, Hancheng Ge, and Jinwook Chung. 2015. Organic or organized?: Exploring url sharing behavior. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 513–522.

[6] Elfreda A Chatman. 1999. A theory of life in the round. *Journal of the American Society for information Science* 50, 3 (1999), 207–217.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1724–1734.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[9] Brenda Dervin. 1998. Sense-making theory and practice: an overview of user interests in knowledge seeking and use. *Journal of knowledge management* 2, 2 (1998), 36–46.

[10] Carsten Eickhoff, Jaime Teevan, Ryen White, and Susan Dumais. 2014. Lessons from the journey: a query log analysis of within-session learning. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 223–232.

[11] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. 2012. Distributional Footprints of Deceptive Product Reviews. *ICWSM* 12 (2012), 98–105.

[12] David Mandell Freeman. 2017. Can you spot the fakes?: On the limitations of user feedback in online social networks. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1093–1102.

[13] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1243–1252.

[14] Guoxiu He and Wei Lu. 2018. Entire Information Attentive GRU for Text Representation. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '18)*. ACM, 163–166.

[15] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications* 13, 4 (1998), 18–28.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Ramon Ferrer i Cancho and Ricard V Solé. 2003. Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences* 100, 3 (2003), 788–791.

[18] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[19] Zhuoren Jiang, Liangcai Gao, Ke Yuan, Zheng Gao, Zhi Tang, and Xiaozhong Liu. 2018. Mathematics Content Understanding for Cyberlearning via Formula Evolution Map. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 37–46.

[20] Zhuoren Jiang, Yue Yin, Liangcai Gao, Yao Lu, and Xiaozhong Liu. 2018. Cross-language Citation Recommendation via Hierarchical Representation Learning on Heterogeneous Graph. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 635–644.

[21] Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 562–570.

[22] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Volume 1: Long Papers* (2014), 655–665.

[23] Mahmood Khosrowjerdi. 2016. A review of theory-driven models of trust in the online health context. *IFLA journal* 42, 3 (2016), 189–206.

[24] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. 1746–1751.

[25] Diederik P Kingma and Jimmy Ba. [n. d.]. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*. 1–15.

[26] James Krikelas. 1983. Information-seeking behavior: Patterns and concepts. *Drexel library quarterly* 19, 2 (1983), 5–20.

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[28] Kyumin Lee, James Caverlee, Zhiyuan Cheng, and Daniel Z Sui. 2013. Campaign extraction from social media. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 1 (2013), 9.

[29] Kyumin Lee, Brian David Eoff, and James Caverlee. 2011. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter.. In *Fifth International AAAI Conference on Weblogs and Social Media*. 185–192.

[30] Tao Lei, Yu Zhang, Sida I Wang, Hui Dai, and Yoav Artzi. 2018. Simple Recurrent Units for Highly Parallelizable Recurrence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4470–4481.

[31] Yuqing Lu, Lei Zhang, Yudong Xiao, and Yangguang Li. 2013. Simultaneously detecting fake reviews and review spammers using factor graph model. In *Proceedings of the 5th annual ACM web science conference*. ACM, 225–233.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Computer Science* (2013).

[33] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model.. In *Interspeech*, Vol. 2. 3.

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[35] Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model.. In *Aistats*, Vol. 5. Citeseer, 246–252.

[36] Myle Ott, Claire Cardie, and Jeff Hancock. 2012. Estimating the prevalence of deception in online review communities. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 201–210.

[37] Sherif Saad, Issa Traore, Ali Ghorbani, Bassam Sayed, David Zhao, Wei Lu, John Felix, and Payman Hakimian. 2011. Detecting P2P botnets through network behavior analysis and machine learning. In *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*. IEEE, 174–180.

[38] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers*. 440–450.

[39] Ning Su, Yiqun Liu, Zhao Li, Yuli Liu, Min Zhang, and Shaoping Ma. 2018. Detecting Crowdturfing "Add to Favorites" Activities in Online Shopping. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1673–1682.

[40] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108* (2015).

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[42] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1288–1297.

[43] Chenglong Wang, Feijun Jiang, and Hongxia Yang. 2017. A hybrid framework for text modeling with convolutional RNN. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2061–2069.

[44] Ryen W White, Gary Marchionini, and Gheorghe Muresan. 2008. Evaluating exploratory search systems. *Information Processing and Management* 44, 2 (2008), 433.

[45] Chang Xu and Jie Zhang. 2015. Towards collusive fraud detection in online reviews. In *2015 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1051–1056.

[46] Chang Xu, Jie Zhang, Kuiyu Chang, and Chong Long. 2013. Uncovering collusive spammers in Chinese review websites. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 979–988.

[47] Junting Ye and Leman Akoglu. 2015. Discovering opinion spammer groups by network footprints. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 267–282.

[48] Wenpeng Yin and Hinrich Schütze. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 901–911.

[49] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630* (2015).