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

Bozhao Zhang
729007584

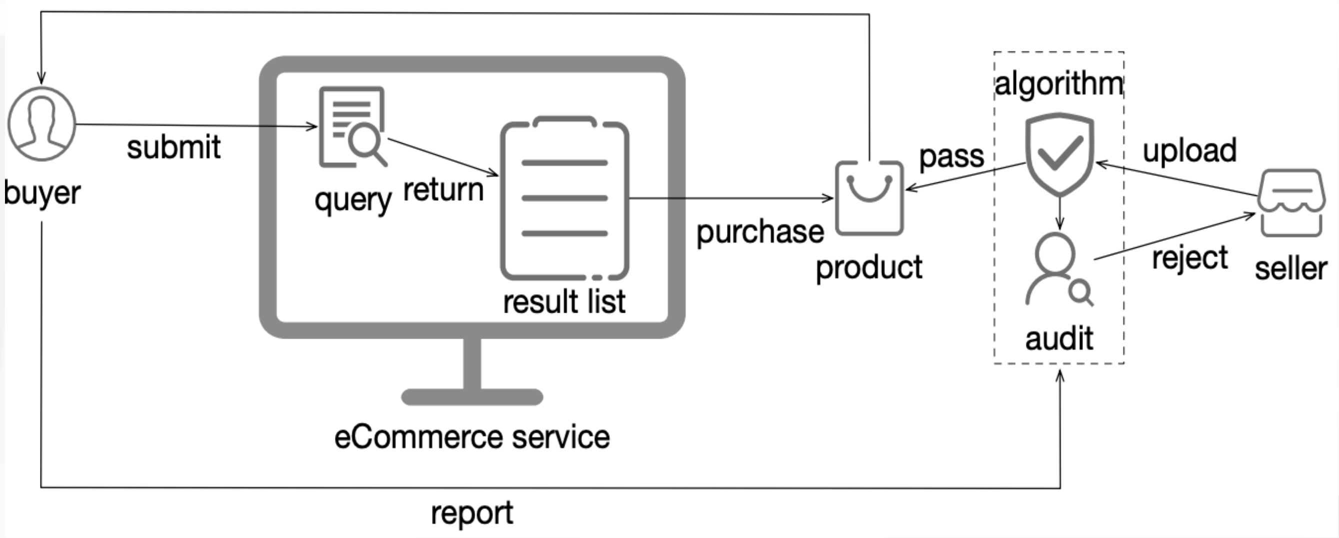
01 Motivation

Background:

- 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.

- While most of decentralized eCommerce platforms don’t have their own inventory, the illegal products, uploaded by some problematic sellers, can spread more easily than ever before. Such risk can be quite harmful to both buyers and cybermarkets.

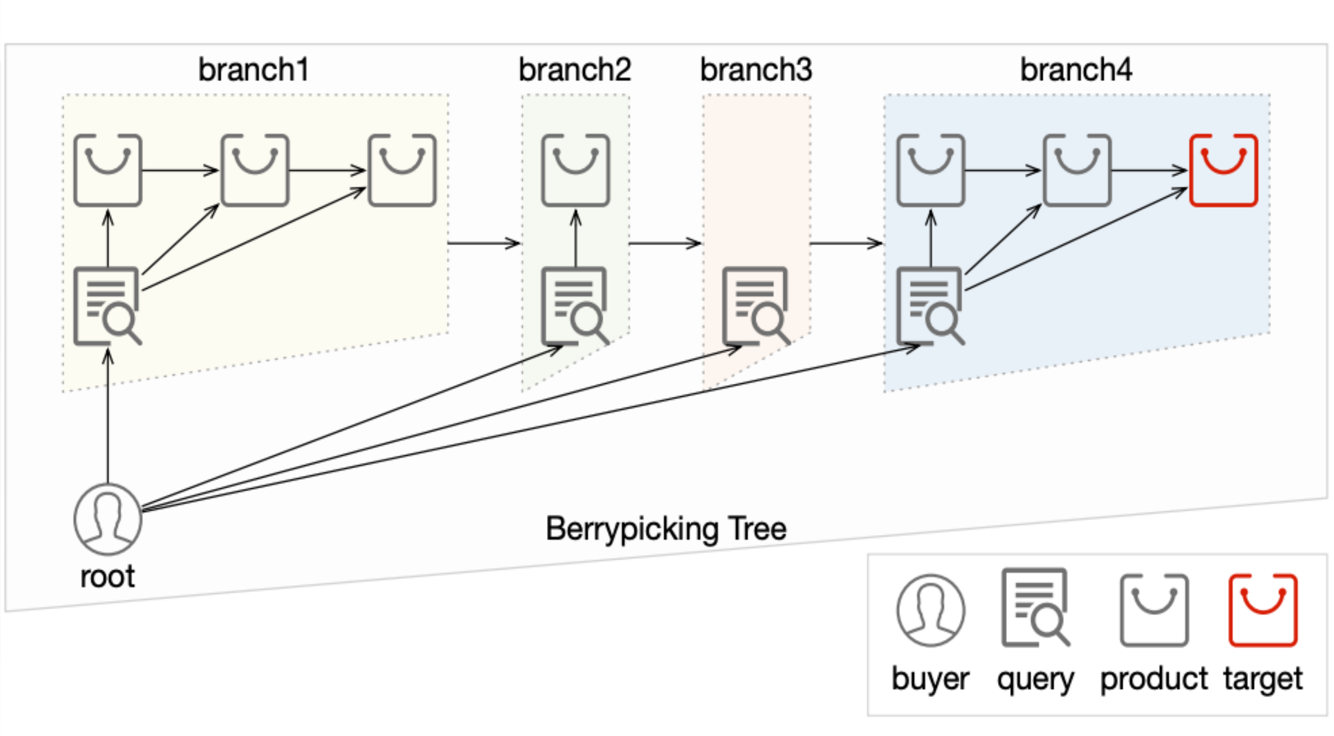
01 Motivation



Traditional Detection System in an eCommerce Service

Brutal Reality: 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 chameleons in the eCommerce ecosystem while traditional learning algorithms can hardly detect them effectively.

01 Motivation

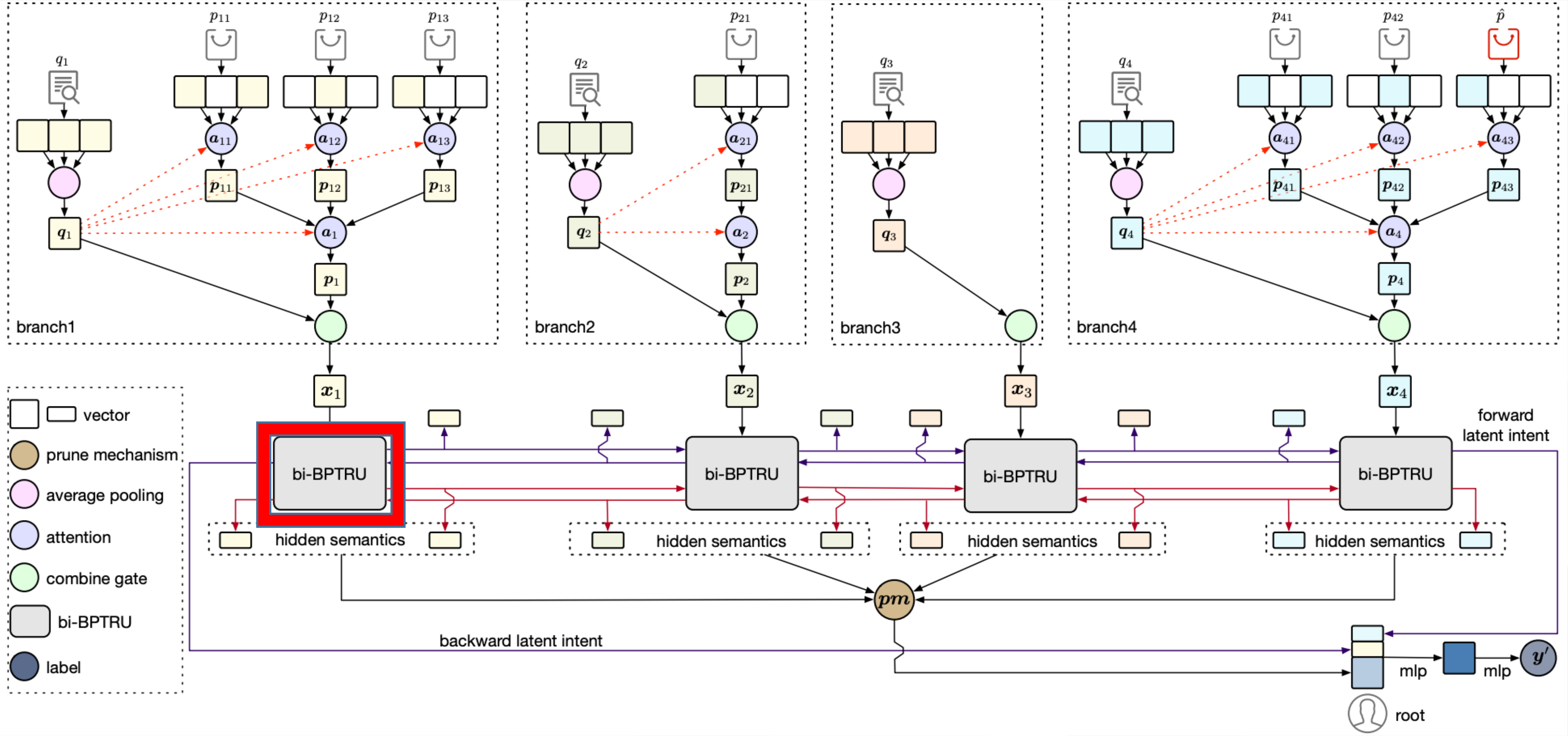


Traditional Berrypicking Tree

Berrypicking (Marcia Bates, 1989):

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.

02 Main Technical Contribution



BerryPIcking TRee MoDel (BIRD): Representation

02 Main Technical Contribution

Berrypicking Tree Recurrent Unit (BPTRU)

$$z_t = \sigma(W_z \cdot x_t + v_z \odot h_{t-1}^1 + b_z)\,,$$

$$r_t = \sigma(W_r \cdot x_t + v_r \odot h_{t-1}^2 + b_r)\,,$$

two hidden gates to determine the combination of the previous hidden state (latent intent) and the current branch.

$$i_t = \sigma(W_i^1 \cdot h_{t-1}^1 + W_i^2 \cdot h_{t-1}^2 + b_i)\,,$$

an interact gate to supplement the joint information.

$$\tilde{h}_t^1 = \boxed{i_t} \odot h_{t-1}^2\,,$$

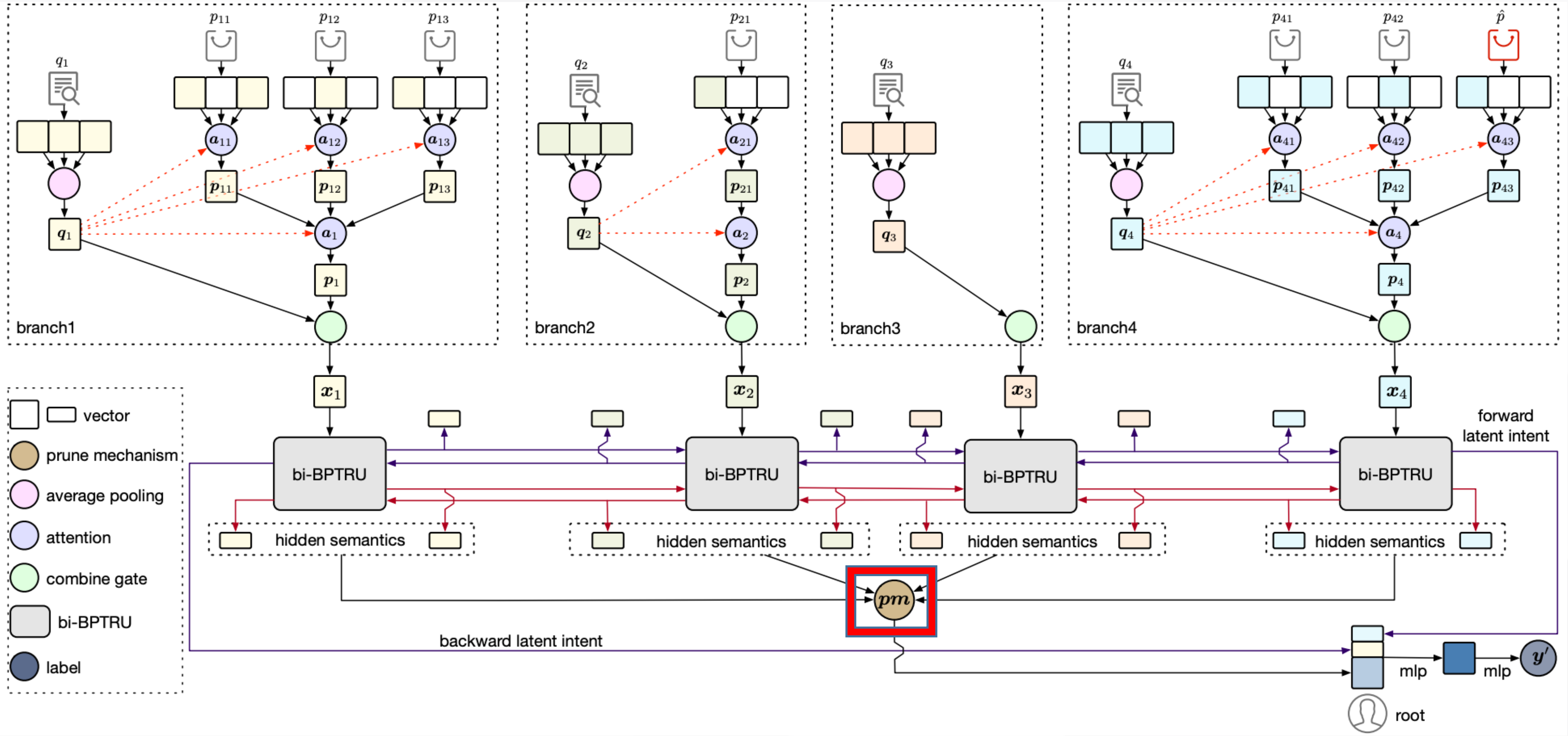
$$\tilde{h}_t^2 = \boxed{i_t} \odot h_{t-1}^1\,,$$

$$h_t^1 = tanh(z_t \odot h_{t-1}^1 + (1 - z_t) \odot (W_h^1 \cdot x_t) + \tilde{h}_t^1)\,,$$

$$h_t^2 = tanh(r_t \odot h_{t-1}^2 + (1 - r_t) \odot (W_h^2 \cdot x_t) + \tilde{h}_t^2)\,,$$

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 among all the information seeking efforts in the tree.

02 Main Technical Contribution



BerryPIcking TRee MoDel (BIRD): Pruning Mechanism

02 Main Technical Contribution

Pruning Mechanism

the last branch contains the target (purchased) product

$$H^{1l} = copy(last(H^1)),$$

consine similarity

$$pm = softmax(\sigma(similar(H^{1l}, H^1))),$$

$$h^{1*} = pm^T \cdot H^1,$$

sigmoid

- 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.

03 Experimental Result

	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	<u>56.12</u>	13.38	21.61	15.79	9.52	21.02
AttenPool	<u>81.81</u>	<u>83.03</u>	<u>58.22</u>	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	<u>58.88</u>	50.10	55.03	34.37	<u>60.09</u>
BPTRU	74.37	74.78	57.63	<u>44.09</u>	<u>49.96</u>	<u>46.26</u>	<u>27.17</u>	<u>47.73</u>	49.26	56.93	<u>52.82</u>	<u>55.21</u>	<u>36.09</u>	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**

Experimental Results of Performance Comparison with Base Models Built on Berrypicking Tree

03 Experimental Result

- The proposed BIRD significantly outperforms all of the baseline models, for nearly all of the metrics except for Precision.

- Except for the simplest models, all models built on berrypicking tree are superior than text classification baselines via product content, which validates that buyers' information seeking behavior can be useful for pornographic products detection.

- Overall, recurrent models perform better, which indicates the deep information hidden in the branches sequence can be useful for this task.

04 My Thoughts

- This task, which dynamically locate the pornographic products, is very novel and interesting.
- The proposed BerryPIcking TRee MoDel (BIRD) not only using text information but also via buyers' seeking behavior logs is very interesting.
- Usually we often use user's behavior on recommendation task, but this method apply it on classification task.

04 My Thoughts (cont'd)

Future Work:

- Enhance the model by using other types of buyer information, such as products dwell time and query similarities across different sessions, which can potentially improve the performance.

- Also apply user information into other classification tasks, such as spam email. We can not only use text information in email but also users how to deal with emails. For example, how long users delete after receiving them and how long users delete them after reading the content.