Unconstrained Product Categorization with Sequence-to-Sequence Models

Maggie Yundi Li∗
National University of Singapore
Singapore
a0131278@comp.nus.edu.sg

Liling Tan, Stanley Kok, Ewa Szymanska
Rakuten Institute of Technology
Singapore
{first.lastname}@rakuten.com

ABSTRACT
Product categorization is a critical component of e-commerce platforms that enables organization and retrieval of the relevant products. Instead of following the conventional classification approaches, we consider category prediction as a sequence generation task where we allow product categorization beyond the hierarchical definition of the full taxonomy.

This paper presents our submissions for the Rakuten Data Challenge at SIGIR eCom’18. The goal of the challenge is to predict the multi-level hierarchical product categories given the e-commerce product titles. We ensembled several attentional sequence-to-sequence models to generate product category labels without supervised constraints. Such unconstrained product categorization suggests possible addition to the existing category hierarchy and reveals ambiguous and repetitive category leaves.

Our system achieved a balanced F-score of 0.8256, while the organizers’ baseline system scored 0.8142, and the best performing system scored 0.8513.

CCS CONCEPTS
• Computing methodologies → Natural language processing;
• Applied computing → Electronic commerce;

KEYWORDS
Text Classification, Sequence-to-Sequence

1 INTRODUCTION
Product categorization is necessary to ensure that e-commerce platforms accurately and efficiently retrieve the relevant items [9]. E-commerce sites use hierarchical taxonomies to organize products from generic to specific classes. For instance, the product ‘Dr. Martens Air Wair 1460 Mens Leather Ankle Boots’ falls under the ‘Clothing, Shoes, Accessories -> Shoes -> Men -> Boots’ category on Rakuten.com.

Product taxonomies allow easy detection of similar products and are used for product recommendation and duplicate removal on e-commerce sites [16, 18]. Although merchants are encouraged to manually input categories for their products when they post them on e-commerce platforms, the process is labor-intensive and leads to inconsistent categories for similar items [3, 10].

Automatic product categorization based on available product information, such as product titles, would thus significantly smooth this process.

Previous approaches to e-commerce product categorization focused on mapping product information (titles, descriptions, images, etc.) to the specific categories based on the existing labels from the training data. Despite the effectiveness of such approaches, products can only be classified into the categories given by the platform. In contrast, the static product category hierarchies would not be able to adapt to the ever-growing number of products on the e-commerce platform. We want to automatically learn the cross-pollination of sub-categories beyond the predefined hierarchy, instead of imposing the hard boundaries inherited from higher level categories.

By redefining the classic product category classification task as a sequence generation task, we were able to generate categories that were not predefined in training data. For example, our model assigned ‘Canon 9167b001 12.8 Megapixel Powershot(R) G1 X Mark II Digital Camera’ to the 3292>1041>4380>4953 category which does not exist in the product taxonomy in the train set. Table 1 shows a sample of related product titles and their respective categories from...
Table 2: Distribution of First Level Categories and the Most Common Label in Each First Level Categories

<table>
<thead>
<tr>
<th>Top-level Categories</th>
<th>Count</th>
<th>(%)</th>
<th>Largest Sub-category</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4015</td>
<td>268,295</td>
<td>0.3353</td>
<td>4015&gt;2337&gt;1458&gt;40</td>
<td>0.031851</td>
</tr>
<tr>
<td>3292</td>
<td>200,945</td>
<td>0.2511</td>
<td>3292&gt;3581&gt;3145&gt;2201</td>
<td>0.037682</td>
</tr>
<tr>
<td>2199</td>
<td>96,714</td>
<td>0.1208</td>
<td>2199&gt;4592&gt;12</td>
<td>0.087393</td>
</tr>
<tr>
<td>1608</td>
<td>85,554</td>
<td>0.1069</td>
<td>1608&gt;4269&gt;1667&gt;4910</td>
<td>0.013727</td>
</tr>
<tr>
<td>3625</td>
<td>29,557</td>
<td>0.0369</td>
<td>3625&gt;4399&gt;1598&gt;3903</td>
<td>0.021400</td>
</tr>
<tr>
<td>2296</td>
<td>28,412</td>
<td>0.0355</td>
<td>2296&gt;3597&gt;689</td>
<td>0.004927</td>
</tr>
<tr>
<td>4238</td>
<td>23,529</td>
<td>0.0294</td>
<td>4238&gt;2240&gt;4187</td>
<td>0.001985</td>
</tr>
<tr>
<td>2075</td>
<td>20,086</td>
<td>0.0251</td>
<td>2075&gt;4764&gt;272</td>
<td>0.004962</td>
</tr>
<tr>
<td>1395</td>
<td>18,847</td>
<td>0.0235</td>
<td>1395&gt;2736&gt;4447&gt;1477</td>
<td>0.004720</td>
</tr>
<tr>
<td>92</td>
<td>8172</td>
<td>0.0102</td>
<td>92</td>
<td>0.010215</td>
</tr>
<tr>
<td>3730</td>
<td>8113</td>
<td>0.0101</td>
<td>3730&gt;1887&gt;3044&gt;4882</td>
<td>0.003978</td>
</tr>
<tr>
<td>4564</td>
<td>5648</td>
<td>0.0070</td>
<td>4564&gt;1265&gt;1706&gt;1158&gt;2064</td>
<td>0.001281</td>
</tr>
<tr>
<td>3093</td>
<td>5098</td>
<td>0.0063</td>
<td>3093&gt;4104&gt;2151</td>
<td>0.001907</td>
</tr>
<tr>
<td>1208</td>
<td>1030</td>
<td>0.0012</td>
<td>1208&gt;546&gt;4262&gt;572</td>
<td>0.000195</td>
</tr>
</tbody>
</table>

Table 2: Distribution of First Level Categories and the Most Common Label in Each First Level Categories

2 SEQUENCE-TO-SEQUENCE LEARNING

The most common Sequence-to-Sequence (Seq2Seq) models belong to the encoder-decoder family. The source sequence, i.e. product title string in our case, is first encoded as a fixed-length vector. This vector is then fed to a decoder, which steps through to generate the predicted output sequence one symbol at a time until an end-of-sequence (EOS) symbol is generated. In the context of product categorization, every sub-category is a symbol in our experiments, and a sequence of the sub-categories forms a full hierarchical category label. The encoder and decoder are jointly trained to maximize the probability of generating the correct output sequence given its input[4, 5, 8, 13].

Simple encoder-decoder performance deteriorates when translating long input sequences; the single fixed-size encoded vector is not expressive enough to encapsulate that much information. To address this problem, the attention mechanism was proposed to learn an implicit alignment between the input and output sequences. Before the decoder generates an item, it first aligns for a set of positions in the source sequence with the most relevant information [1]. The model then predicts the target item based on the context vectors of these relevant positions and the history of generated items. In other words, attention extracts contextual information for every symbol processed.

3 DATASET CHARACTERISTICS

The Rakuten Data Challenge (RDC) dataset consists of 1 million product titles and the anonymized hierarchical category labels. The data was split 80-20 into training and testing set. The test labels were kept unknown until the end of the competition.

3.1 Class Imbalance

Unbalanced class distribution presents a significant challenge to general classification systems, such as nearest neighbors and multi-layered perceptron, despite remedies, like up-/downsampling and cost-sensitive learning, with limited effectiveness [12].

Like most e-commerce product categorization data [2, 6, 17], the distribution of the 14 top-level categories is highly skewed, as shown in Table 2. A similar imbalance is found in the distribution of the sub-category labels. From the train set, there are over 3000 unique sub-categories. The largest category (2199>4592>12) contains ~69,000 product titles that made up 8.7% of the 800,000 product titles from the train set.

3.2 Noisy Product Titles

Noise is inherent to product categories datasets; the RDC dataset is no different. Related works on product categorization had dedicated approach to address the noise through a combination of feature engineering and classifier ensembles [3, 10].

The training data that overlapped with the 3292>1041>4380>4558 label.
one or more of the characters on the right, e.g., the \x99 appears in 2 to 10 product titles.\footnote{The penultimate character in the >50 list is the non-breaking space \xa0 and the last character is a replacement character. They appear in 643 and 766 product titles respectively. Usually, these are breadcrumbs of the HTML to Unicode conversion [14, 15].}

Upon inspection, we find that the noise can be helpful to the learning systems due to their systematic nature. For example, the same strings of non-ASCII-printable characters appear consistently in clothing category (1608>4269), such as "I (Heart) My *string of non-ASCII-printable characters* - INFANT One Piece - 18M" in category 1608>4269>4411>4306 and "Frankie Says Relax Statement Women's T-Shirt by American Apparel by Spreadshirt *string of non-ASCII-printable characters*" in category 1608>4269>3831>62. Hence, we decided not to remove the noise detected in the product titles.

### 4 EXPERIMENTS

We lowercased the product titles from the RDC dataset and tokenized the data with the Moses tokenizer\footnote{https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl} to frame the product categorization task into Seq2Seq generation, we split the categories up into its sub-categories and treat the category as a sentence. For example, "4015>3636>1319>1409>3606" is changed to "4015 3636 1319 1409 3606".

#### 4.1 Models

Without explicit tuning, we trained a single-layer attentional encoder-decoder using the Marian toolkit\footnote{https://github.com/alvations/sacremoses} (commit 429d4a) with the following hyperparameters.

- **RNN Cell**: GRU
- **Source/Target Vocab size**: 120,000
- **Embedding dim.**: 512
- **En/Decoder dim.**: 1024
- **Embedding dropout**: 0.1
- **Dropout**: 0.2
- **Optimizer**: Adam
- **Batch size**: 5000
- **Learning Rate**: 0.0001
- **Beam Size**: 6

We allowed the model to over-fit the training data by using the full training set as our validation set. We trained the baseline model for 2 hours and stopped arbitrarily at the 7th epoch when the perplexity reaches 1.18. Our baseline model achieved 0.81 weighted F-score in the phase 1 result.

For the rest of the submissions, we ensembled the baseline model with the models trained on different random seeds, and we stopped the training when we observed that the perplexity on the validation set falls below 1.0. It is unclear what is the benefit of over-fitting the model to the training set and expecting a 1.0* perplexity, but the assumption is that at inference, given a product title that was seen in training, the model should output the same label.

Table 3 presents the validation metrics (cross-entropy and perplexity) for the different models. In retrospect, we could have been more disciplined in the stopping criteria and monitor the model validation more closely to stop with a consistent criterion, e.g., limiting the no. of epochs/steps or a particular threshold for the validation metric.

### 5 RESULTS

Table 4 presents the precision, recall, and F-score of the baseline and ensemble systems. The phase 1 results are based on a subset of the full test data, and the phase 2 results are based on the entire test dataset. Our baseline system achieved competitive results with 0.81 weighted F-score in phase 1 of the data challenge and the ensemble systems improved the performance scored 0.82 in phase 1 and 2 of the challenge.\footnote{Initially, the data challenge reported scores to 2 decimal places, and the change to 4 decimal places happened in the last couple of days of the challenge. Since the labels for the test set were not available at the time of publication, we could not perform postmortem evaluation to find out the scores for the M1 baseline and M1-3 ensemble models.\footnote{The full ranking of the data challenge is available on https://sigir-eecom.github.io/data-task.html\footnote{We only analyzed the attention weight alignment on the test set minimally because the gold labels on the test set were not made accessible.}}

Similarly, the best system (mcskinner) in the competition is an ensembled neural network system\footnote{https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl}. It used an ensembled of multiple bi-directional Long Short Term Memory (LSTM) with a novel pooling method that balances max- and min-pooling across the recurrent states. The best system scored 0.85 in phase 2. However, the best system follows the traditional classification paradigm where supervised inference produces a fixed set of labels learned from the training data.

### 6 ANALYSIS

#### 6.1 Attention Alignment

The ability to generate alignments between the source and target sequences allows us to easily interpret the category predictions with respect to their product titles. We generated the attention weight alignment between source and target sequences for the training set using the baseline model, $M_1$.\footnote{Python port: https://github.com/alvations/sacremoses}
In this section, we analyze the behaviors of the model predictions in relation to their attention alignment based on cherry-picked examples (Figure 2-6). We also discuss the implications of such behaviors on the existing product category hierarchy.

**6.2 Music Category**

The first row of Example 1 in Figure 3 shows an interesting phenomenon that the ‘</s>’ (end of sentence) token is highly associated with the 2296 first level category. The attention model might have learned to correlate short sequence length with 2296 category. The 2296 category seems to be related to media content whose titles are often succinct; in the train set, there are 2085 single token product titles out of which 1720 titles has 2296 as their first level category.

When the product titles are terse, the model is unable to distinguish between the fine-grained subcategories. In Example 2, the true label in the 2296>3597>1997 refers to the ‘Media>Music>Electronica’ category, but the model predicts it to be 2296>3597>689 i.e. the ‘Media>Music>Pop’ category. Although the model is smart enough to discover the correct top-level(s) categories by learning to associate short sequence with 2296 label, it fails to correctly identify the lowest level category. There are 25 subcategories under the 2296>359, without additional information, it would be hard even for a human to categorize the music genre based on short and sometimes single-word product title.

**6.3 Machine Created Categories**

Unlike traditional classification, the Seq2Seq approach has the ability to generate new categories.

<table>
<thead>
<tr>
<th>Model</th>
<th>Data Split</th>
<th>Creation Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Train</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>46</td>
</tr>
<tr>
<td>M1-4</td>
<td>Train</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5: Count of Created Categories

Table 5 shows the breakdown of the created categories when we applied the models to the train and test set. While the baseline

1https://www.rakuten.com/search/asiatisch/4464/
2We found this out by searching the product titles from the train set that are labeled with 2296>3597>689 on Rakuten.com, e.g. https://www.rakuten.com/search/Grey%20Sky%20Over%20Black%20Town/4455/
model created 2 new categories, it created 46 categories on the test set. During model training, the optimizer makes updates that discourage the creation of new categories to minimize cross-entropy loss and perplexity. The M1 baseline model created 46 new categories on the test set, while the M1-4 ensemble model produced only 1 new category.

Example 3 and 4 from Figure 4 demonstrates how Seq2Seq model creates cross-pollinated categories. In this example, the baseline Seq2Seq model M1 assigned the product, "PM Company 07622 One-Ply Adding Machine/Calculator Rolls- 2-1/4" x 17 ft- White- 5/Pack", with a new category, 4238>2149>1286.

To breakdown this created category, we find in the train set that the overarching category 4235>2149 is for paper-related stationary products. The last sub-category 1286 consistently appears in 4238>4960>1286 which includes calculator-like machines and their accessories, like calculator cases.

In 4238>4960>1286, we also spotted a product analogous to Example 6, "Universal Adding Machine/Calculator Roll, 16 lb, 1/2” Core, 2-1/4” x 150 ft,White, 100/CT - UNV35710". The presence of this calculator printing roll from a different brand may suggest that Example 6 should fall under the same category. However, calculator-like machines dominate the category 4238>4960>1286 by constituting 95 out the 105 products in the train set. Therefore, 4238>2149>1286, created by our Seq2Seq model, is an adequate suggestion for a new category of calculator printing rolls.

The ensemble model (M1-4) created one novel category by labelling the product "Natural Tech Well-Being Conditioner - 1000ml/33.8oz" as 3625>594>1928. However, it is unclear whether the created category is a valid one without the true labels of the test set which is not released prior to the paper publication.

There is a variety of creations across almost all categories in the existing category hierarchy. Although some are mislabeling, many of these created categories are worth considering for adaptations and additions to the existing ones.

7 CONCLUSION

By framing the product categorization task as a sequence generation task, we trained attentional sequence-to-sequence models to generate unconstrained product categories that are not limited to the supervised labels from the training dataset. These models created new categories based on the existing sub-categories, suggesting improvement to existing product taxonomy. Categorization outcomes by these models can also highlight repetitive and ambiguous categories. In contrast to the traditional classification paradigm, the attention weight alignment generated for each product title makes the model easily interpretable. With an F1-score of 0.82 in the Rakuten Data Challenge at SIGIR eCom'18, attentional sequence-to-sequence models are shown to be adequate for product categorization.

ACKNOWLEDGEMENTS

We thank the organizers for organizing the Rakuten Data Challenge. Our gratitude goes to Rakuten Institute of Technology (Singapore), for their support and the computation resources for our experiments. Additionally, we thank our dear colleagues, Ali Cevahir

---

9Examples: Paper | FE4280-22-250 in 4238>2149>1644 and Lissom Design 24021 Paper Block Set -WB in 4238>2149>488
10Examples: Hewlett Packard HP 10s Scientific Calculator, Casio DR-210TM Two-Color Desktop Printing Calculator and Ti Nspire Cx Graphing Calc
11Guerrilla Accessories T88BLKSC T83 Plus Silicone Case Black
12By inspecting the training data, most of the hair conditioner in the train set fall under the category 3625>3641>1928. M1-4 combined that category with 3625>594>..., which seems to be the skincare sub-category. This category creation, though sensible, might be a mislabel because 3625>3641>1928 is a well-defined hair product category.
13The full list of created categories and dataset exploratory code described in Section 3 is available on https://github.com/MaggieMeow/neko
and Kaidi Yue, for sharing their knowledge and insights in related research subjects.

REFERENCES


