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. 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

Category: 3625>2644>921>1615
Clean Matte Pressed Powder - # 545 Warm Beige by CoverGirl for Women - 0.35 oz Powder
Laura Mercier Silk Creme Oil Free Photo Edition Foundation - Sand Beige 1oz
Aqva Pour Homme After Shave Balm (Tube) - 100ml/3.4oz
IMAN Second to None Luminous Foundation, Clay 5 .35 oz (10 g)
Category: 3625>2644>2805>3870>1102
Makeup Blender, Assorted Colors, 10 Count
Category: 3625>2644>2805>2522
SHANY Detox Professional Brush Cleanser - Instant dry - Refill - 16oz
Category: 3625>2644>2805>3870>2697
8Pcs Make Up Cosmetic Brushes Set Powder Foundation Eyeshadow Lip Brush Tool Kit
Category: 3625>2644>2805>3870>4627
Studded Couture - 12 Piece Brush Set

Table 1: Product Titles and Categories in the Training Data

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].

*This is the corresponding author

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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. However, 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 pre-defined 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

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

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

that were not pre-defined in training data. For example, our model assigned *"Cover Girl - Proctor Outlast Stay Luminous Foundation Cls Tan"* the `3625>2644>2805>1615` label which does not exist in the product taxonomy in the training set. Table 1 shows a sample of related product titles and their respective categories from the training data that overlapped with the `3625>2644>2805>1615` label.

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 training 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 training 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].

```
1       ['\x9d', '\x9f', ', ', '\xad', '¨', '⅜', '⅝',
         'i', '¦', 'm²', 'µ', 'œ', '!', 'ђ', 'њ', '´',
         'ù', '\x96', '', ' (', ') ', 'ō', '×', '£',
         '↑', '\x81', '| ']

2-10    ['ú', '\x94', '\x7f', 'в', 'ē', 'ç', 'ū', '-',
         '—', '⅛', '•', 'ï', '¿', '·', 'ê', 'º', '¦',
         'ä', 'ñ', '²', '├', '⌐']

10-50   ['ñ', '²', '├', '⌐', 'ö', '–', '¼', '†', 'ž',
         '±', '¡', '…', 'í', 'μ', '„', 'ó', 'å', 'á',
         'æ', '¾', '\x99', 'š'

>50     ['¬', '€', ',', 'ƒ', 'ë', '¢', 'ã', 'â', '\xa0', '�']
```

Figure 1: Lists of Characters not in Printable ASCII Range

We checked for common noise signatures in the RDC product titles by searching for characters beyond the printable ASCII range (0x20 to 0x7E). Figure 1 shows the list of characters outside the range, the left side shows the number of product titles that contain one or more of the characters on the right, e.g., the `\x99` appears in 2 to 10 product titles.[1]

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>3031>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[2,3]. 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[7] (commit `f429d4a`) with the following hyperparameters.

- **RNN Cell**: GRU
- **Source/Target Vocab size**: 120,000
- **Embedding dim.:** 512
- **En/Decoder dim.:** 1024
- **Embedding dropout:** 0.1
- **RNN 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.

Model	Random Seed	Epoch	Cross-entropy	Perplexity
M1	0	77	0.8446	1.1835
M2	1	189	0.0191	1.0038
M3	1	470	0.0723	1.0145
M4	2	54	0.0542	1.0108

Table 3: Cross-entropy and Perplexity during Model Training

Phase	Model(s)	P	R	F
1	M1 (Baseline)	0.82	0.81	0.81
	M1-3	0.83	**0.83**	0.82
	M1-4	**0.8311**	0.8296	**0.8245**
2	M1-4	0.8267	0.8305	0.8256
	Best system (`mcskinner`)	0.8697	0.8418	0.8513

Table 4: Precision, Recall, F1 Scores on Held-out Test Set

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 ensembled systems improved the performance scored 0.82 in phase 1 and 2 of the challenge.[4,5]

Similarly, the best system (`mcskinner`) in the competition is an ensembled neural network system[11]. 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 CONCLUSION

By framing the product categorization task as a sequence generation task, we trained an attentional sequence-to-sequence model to generate non-constrained product categories that are not limited to the supervised labels from the training dataset. Our system achieved an F1-score of 0.82 in the Rakuten Data Challenge at SIGIR eComâĂŹ18.

[1]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]

[2]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[3]Python port: https://github.com/alvations/sacremoses

[4]Initially, the data challenge reported scores to 2 decimal places, and the change to report 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

[5]The full ranking of the data challenge is available on https://sigir-ecom.github.io/data-task.html

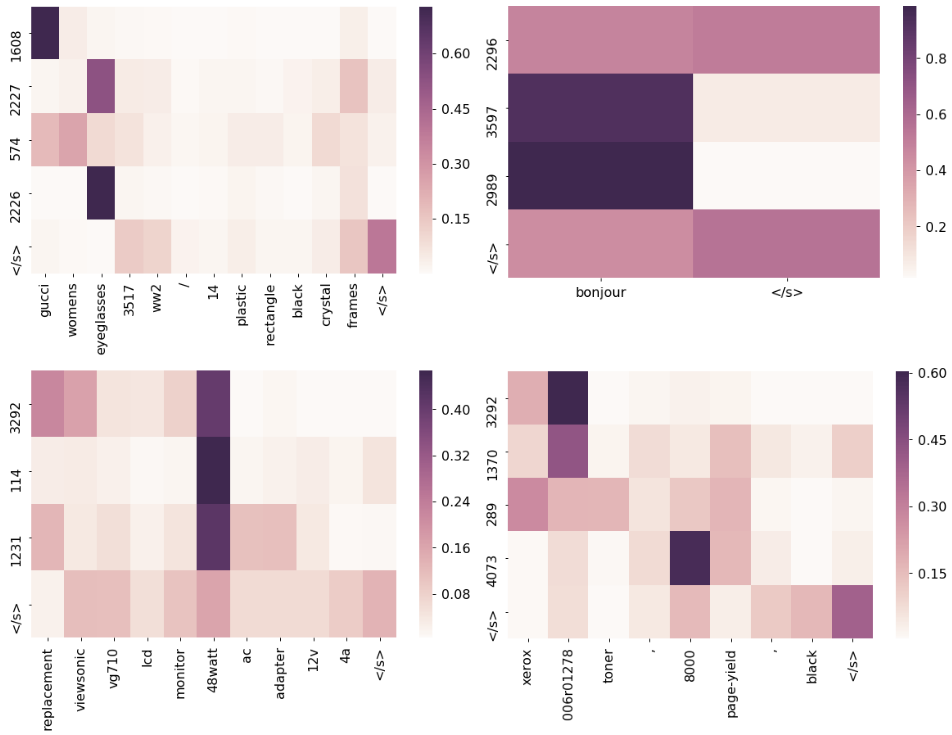

Figure 2: Examples of Attention Alignment on Correctly Labeled Products from the Training Set

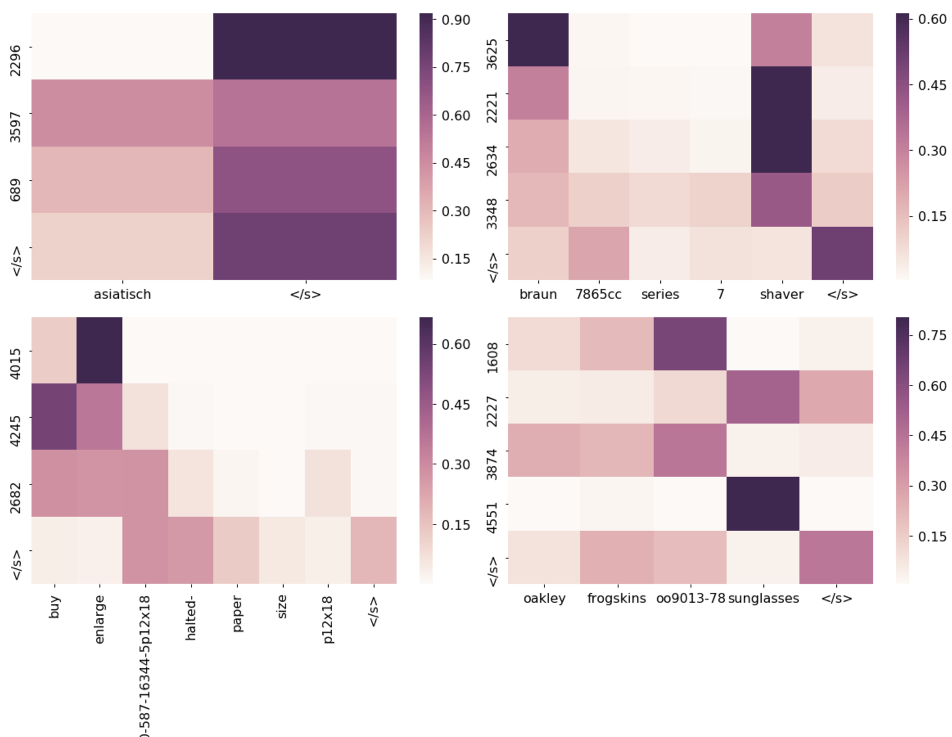

Figure 3: Examples of Attention Alignment on Mislabeled Products from the Training Set

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 and Kaidi Yue, for sharing their knowledge and insights in related research subjects.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Ali Cevahir and Koji Murakami. 2016. Large-scale Multi-class and Hierarchical Product Categorization for an E-commerce Giant. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 525–535.

[3] Jianfu Chen and David Warren. 2013. Cost-sensitive Learning for Large-scale Hierarchical Classification. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management (CIKM '13)*.

[4] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics.

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

[6] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee.

[7] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, AndrÃľ F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. *arXiv preprint arXiv:1804.00344* (2018). https://arxiv.org/abs/1804.00344

[8] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

[9] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. 2012. Supercharging Recommender Systems Using Taxonomies for Learning User Purchase Behavior. In *Proceedings of VLDB Endowment*.

[10] Zornitsa Kozareva. [n. d.]. Everyone Likes Shopping! Multi-class Product Categorization for e-Commerce. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, year = 2015*.

[11] Michael Skinner. 2018. Product Categorization with LSTMs and Balanced Pooling Views. In *SIGIR 2018 Workshop on eCommerce (ECOM 18)*.

[12] Yanmin Sun, Andrew K. C. Wong, and Mohamed S. Kamel. 2009. Classification of Imbalanced Data: a Review. *International Journal of Pattern Recognition and Artificial Intelligence* 23, 4 (2009), 687–719.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*.

[14] Liling Tan and Francis Bond. 2011. Building and Annotating the Linguistically Diverse NTU-MC (NTU-Multilingual Corpus). In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*.

[15] Liling Tan, Marcos Zampieri, Nikola Ljubesic, and Jorg Tiedemann. 2014. Merging comparable data sources for the discrimination of similar languages: The dsl corpus collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC)*.

[16] Li-Tung Weng, Yue Xu, Yuefen Li, and Richi Nayak. 2008. Exploiting Item Taxonomy for Solving Cold-Start Problem in Recommendation Making. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*.

[17] 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 *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics.

[18] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. 2004. Taxonomy-driven computation of product recommendations. In *CIKM*.