Large-Scale Categorization of Japanese Product Titles Using Neural Attention Models

Yandi Xia, Aaron Levine, Pradipto Das
Giuseppe Di Fabbrizio, **Keiji Shinzato** and **Ankur Datta**
Rakuten Institute of Technology, Boston, MA, 02110 - USA
{ts-yandi.xia, aaron.levine, pradipto.das,
giuseppe.difabbrizio, keiji.shinzato, ankur.datta}@rakuten.com

Abstract

We propose a variant of Convolutional Neural Network (CNN) models, the Attention CNN (ACNN); for large-scale categorization of millions of Japanese items into thirty-five product categories. Compared to a state-of-the-art Gradient Boosted Tree (GBT) classifier, the proposed model reduces training time from three weeks to three days while maintaining more than 96% accuracy. Additionally, our proposed model *characterizes* products by imputing attentive focus on word tokens in a language agnostic way. The attention words have been observed to be semantically highly correlated with the predicted categories and give us a choice of automatic feature extraction for downstream processing.

1 Introduction

E-commerce sites provide product catalogs with millions of items that are continuously updated by thousands of merchants. To list new products in an e-commerce marketplace and expose them to online users, merchants must supply several pieces of meta-data. Rakuten Ichiba[1] is an example of such a large-scale e-commerce platform in Japan, hosting more than 239 million products from over $44,000$ merchants. To improve search relevance and catalog navigation, products must be categorized into a taxonomy tree with thousands of nodes several levels deep (e.g., 6 levels with more than $43,000$ nodes for Rakuten Ichiba).

For such a large taxonomy, manual item categorization is often inaccurate and inconsistent across merchants. Automatic categorization into a full taxonomy tree is feasible, although a layered approach is more practical for scalability and accuracy reasons. For instance, Shen et al. (2012b) uses a two level strategy to combat imbalance. Das et al. (2017) also exploits a similar 2-step cascade categorization.

This work focuses on large-scale categorization of Japanese products for the top-level categories of the Rakuten Ichiba catalog taxonomy. Examples of top-level product categories include *Clothing*, *Electronics*, *Shoes*, and *Books & Media*, as well as less represented categories such as *Travel*, *Communication*, and *Cars & Motorbikes*. We compare Convolutional Neural Network (CNN), Attention CNN (ACNN), and state-of-the-art Gradient Boosted Tree (GBT) classification models trained on more than 18 million catalog items. ACNN model performance is comparable to that of the GBT model with a 7-fold reduction in training time without the need for feature engineering. Additionally, ACNN's attention mechanism selects salient words that are semantically relevant to identifying categories and potentially useful for automatic language-agnostic feature extraction.

2 Related Work

Research on large-scale product categorization has recently come into focus (Shen et al., 2011; Shen et al., 2012b; Shen et al., 2012a; Yu et al., 2013; Chen and Warren, 2013; Sun et al., 2014; Kozareva, 2015). Most contemporary work in this area points out the noise issues that arise in large product datasets and address the problem with a combination of a wide variety of features and standard classifiers. However, the existing methods for noisy product classification have only been applied to English. Their efficacy for *moraic* and *agglutinative* languages such as Japanese remains unknown.

Application of deep learning techniques is gaining grounds for text categorization applications (Kim, 2014; Ma et al., 2015; Yang et al., 2016), however, their application to product data has only been recently reported. Pyo et al. (2016) uses Re-

[1] Ichiba `http://www.rakuten.co.jp`

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 663–668,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

current Neural Networks (RNNs) without word embeddings. Furthermore, unlike our proposed model, RNNs cannot impute tokens in title text with attention weights that can be helpful in downstream applications.

Dependency-based deep learning (Ma et al., 2015) has proven useful for sentence classification, but product titles, whether in English or Japanese, are not beholden to the same grammatical rigor. We do not use deeper linguistic techniques such as parsing or Part-of-Speech tagging due to the language-agnostic nature of our categorization techniques. Attention-based deep learning models have been used in the image domain (Xu et al., 2015) and in the generic text classification domain (Yang et al., 2016). However, to the best of our knowledge, this is the first work on simultaneous categorization and attention based salient token selection on Japanese product data.

3 Dataset Characteristics

The data we use is a selection of product listings from Rakuten Ichiba, a large Japanese E-commerce service for thousands of merchants. Each merchant submits their own product data, leading to item names with inconsistent formats and disagreements on genres for the same sets of items. Our training set consists of $18,199,420$ listings and the test set of $2,274,928$ listings, for a 90/10$\%$ split. The training data is uniformly sampled before the split. Due to the popularity of certain product types, the balance is unevenly distributed between 35 top-level categories: There are $1,869,471$ in the largest category, but only 925 in the smallest.

Statistics	Training set	Test set
Mean character count/title	62.510	62.506
Standard deviation	31.496	31.492
Mean word count/title	23.187	23.188
Standard deviation	11.945	11.942
Mean character count/word	05.800	05.799

Table 1: Word and character level statistics for our Rakuten Ichiba dataset.

Table 1 shows character and word level statistics per product title in the training and test set. It is evident from the mean word and character counts that, on average, Japanese product titles in our dataset are quite verbose. We thus expect that convolutional neural network based models that rely on full context of the input text, to work better for the categorization task.

4 Modeling Approaches

4.1 Attention Neural Networks

Our ACNN model is related to the work described in Yang et al. (2016), which has been more suitable for well-formed document classification tasks with a limited number of categories.

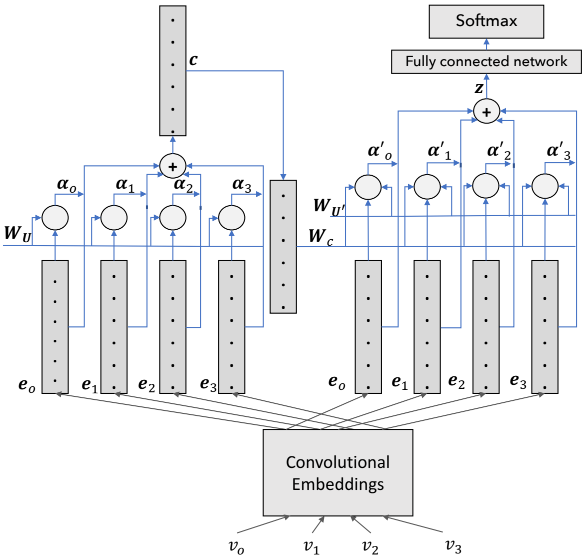


Figure 1: Attentional CNN model architecture

The gating mechanisms shown in the left module of Fig. 1 are akin to the hierarchical model in (Yang et al., 2016). However, their model ends at that module, at which point it is connected to the softmax output layer. In our model, the left module acts as a *context* encoder and the right module acts as an *attention* mechanism that is dependent on the encoded context and input.

Generally speaking, the local convolutional operations are unaware of the existence of preceding or succeeding convolutions over the text sequence. The context module enables propagation of stronger shared parameters through a *context embedding*, leading to the higher weighting of attention over specific parts of the inputs. The propagation strength in the network builds up based on the pattern of context present in the input sequences across several training examples and the training loss incurred for the encoding.

The filters of the convolutional layer (LeCun and Bengio, 1995; Kim, 2014) are convolved with with a window of consecutive observations (characters or words), and produce an encoding of the window. In Fig. 1, $\mathbf{e}_i$ is the encoding of i^{th} window, where, each window is defined over a word or character in the input sequence. For our ACNN model, the input sequence is treated as a sequence of words and then as a separate sequence of characters with the two distinct sequences being concatenated as a single input sequence, $\{v_0, v_1, ...v_L\}$. The value of L is three in Fig. 1. Each variable, $\mathbf{u}_i$, encodes the non-linearity of the linear manifold on $\mathbf{e}_i$ over a set of shared parameters, $\mathbf{W_U}$, and biases, $\mathbf{b_U}$, as $\mathbf{u}_i = \tanh\left(\mathbf{W}_\mathbf{U}^T\mathbf{e}_i + \mathbf{b_U}\right)$. The variables $\mathbf{e}_i$ actually correspond to *parts of the data* and $\mathbf{u}_i$ help aggregate the values of $\mathbf{e}_i$ projected along the learned directions in the parameter space for $(\mathbf{W_U}, \mathbf{b_U})$. Each value of $\mathbf{u}_i$ is computed independent of $\mathbf{u}_{j \neq i}$. The shared parameter $\mathbf{W_U}$ is a $D \times F$ matrix, where D is a hyper parameter chosen for the attention mechanism and F is the number of convolution filters, both chosen during cross-validation. The variables $\mathbf{e}_i$ and $\mathbf{b_U}$ are F dimensional vectors.

The inputs to the context encoding vector, represented by the variable $\mathbf{c}$, are local softmax functions of the form:

$$\alpha_i = \frac{\exp(\mathbf{u}_i^T\mathbf{w}_u)}{\sum_j \exp(\mathbf{u}_j^T\mathbf{w}_u)} \qquad (1)$$

The encoded *context vector* $\mathbf{c}$ is then simply $\mathbf{c} = \sum_i \alpha_i\mathbf{e}_i$. Obtaining the *input encoding* for the attention module is similar to context encoding except that $\mathbf{u}'_i$ depends on a separate set of shared parameters, $\mathbf{W}_{\mathbf{U}'}$ as well as $\mathbf{W}_c$, for the context and corresponding bias term $\mathbf{b}$. In this case, we have:

$$\mathbf{u}'_i = \tanh\left(\mathbf{W}_{\mathbf{U}'}^T\mathbf{e}_i + \mathbf{W}_c^T\mathbf{c} + \mathbf{b}\right) \qquad (2)$$

The softmax functions α'_i are similarly defined as in Equ. 1, but w.r.t. $\mathbf{u}'_i$ and $\mathbf{w}_{u'}$. The α'_is can be thought as the maximum of the *relevance* of the variables $\mathbf{u}'_i$, according to the context $\mathbf{c}$. The output, $\mathbf{z}$, from the attention module is the weighted arithmetic mean, $\sum_i \alpha'_i\mathbf{e}_i$, where the weight represent the relevance for each input variable v_i, through $\mathbf{e}_i$, according to the context $\mathbf{c}$.

We use windows over both word and character observation embeddings of the input text (either tokens or single Japanese characters). We concatenate the word and character encoding vectors and input it to a fully connected layer. A cross-entropy loss is imposed at the output layer.

4.2 Gradient Boosted Trees

GBTs (Friedman, 2000) optimize a loss functional: $\mathcal{L} = E_y[L(y, F(\mathbf{x})|\mathbf{X})]$ where $F(\mathbf{x})$ can be a mathematically difficult to characterize function, e.g., a decision tree $f(\mathbf{x})$. The optimal value of the function is expressed as $F^\star(\mathbf{x}) = \sum_{m=0}^{M} f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$, where $f_0(\mathbf{x}, \mathbf{a}, \mathbf{w})$ is the initial guess and $\{f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})\}_{m=1}^{M}$ are *additive boosts* on $\mathbf{x}$ defined by the optimization method. The parameter $\mathbf{a}_m$ of $f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$ denotes split points of predictor variables and $\mathbf{w}_m$ denotes the boosting weights on the leaf nodes of the decision trees corresponding to the partitioned training set $\mathbf{X}_j$ for region j.

Each boosting round m updates the weights $\mathbf{w}_{m,j}$ on the leaves and creates a new tree. The optimal selection of decision tree parameters is based on optimizing the $f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$ using a logistic loss.

5 Experimental Setup and Results

5.1 Data Preprocessing

Tokenization of Japanese product titles is done using MeCab[2]. The tokenizer is trained using features that are augmented with in-house product keyword dictionaries. Romaji words written using Latin characters are separated from Kanji and Kana words. All brackets are normalized to square brackets and punctuations from non-numeric tokens are removed. We remove anything outside of standard Japanese UTF-8 character ranges. Finally, canonical normalization changes the code points of the resulting Japanese text into an NFKC normalized[3] form.

For GBT, we use several features – at the tokenized word level, we use counts of word unigrams and word bi-grams. For character features, the product title is first normalized as discussed above. Character 2, 3, and 4-grams are then extracted with their counts, where extractions include single spaces appearing at the end of word boundaries. Feature engineering for GBT uses cross-validation to identify the best set of feature combinations and is thus *time consuming*.

The embedding representation of words and characters for the CNN-based classifiers is performed over the normalized input on which feature extraction for GBT is done. To reduce GPU memory consumption, the CNN-based models are trained on titles from which words and characters that appear in less than 20 titles in the training set are removed. Such rare token removal is not performed on the training data for the GBT models since they are trained on CPU servers.

[2] https://sourceforge.net/projects/mecab/

[3] http://unicode.org/reports/tr15/

5.2 Classifier Comparison

In this section, we compare categorization performance of a baseline CNN model w.r.t. our proposed model and a state-of-the-art GBT classifier. We use 10-fold cross-validation over 90% of the training data to perform parameter tuning.

ACNN model parameter setup - The words and characters are in an embedding vector space of dimension 300. These embeddings are trained on the product title training corpus. We use four different window sizes $1, 3, 4, 5$ for words and another of size 4 for characters. The dimension of the filter encoders $\mathbf{e}_i$ and $\mathbf{e}'_i$ is 250, which is the same as the number of filters. The hidden layer size is the number of window sizes times the number of filters i.e., $1, 250$ and we also use a dropout with 0.5 probability on the hidden layer. The CNN models are run for a *maximum of three days* on a server with 8 Nvidia TitanX GPUs and the best model corresponding to the iteration for the lowest validation error is used for test set evaluation.

GBT model parameter setup - For each category, the boosted stumps for the GBT (Chen and Guestrin, 2016) models are allowed to grow up to a maximum depth of 500. The initial learning rate is assigned a value of 0.05 and the number of boosting rounds is set to 50. For leaf node weights, we use L_2 regularization with a regularization constant of 0.5. The GBT models are trained on a 64-core CPU server.

Models	Micro-F1	Training Time
GBT	96.23	3 weeks
CNN	95.90	3 **days**
ACNN-word	96.00	3 **days**
ACNN-word-character	**96.27**	3 **days**

Table 2: Micro-F1 measures for evaluated models. The Micro-precision scores (not shown here) are very similar to the micro-F1 scores with occasional differences in the third and fourth decimal places.

Table 2 shows that our proposed ACNN model – the CNN model augmented with word and character based attention mechanisms, improves over the baseline CNN model by an absolute 0.37%, which translates to more than $8,000$ test titles being correctly classified additionally. Although, the improvement of the proposed model is not significant when using a stringent p-value of 0.0001 (i.e., a typical value used in industrial setting), we emphasize that in practice any increase in accuracy helps (e.g., an additional million items when considering the whole Ichiba catalog).

Both GBT and our proposed ACNN model perform well for top level categorization of Japanese product titles. However, do the models make similar mistakes on the test set?

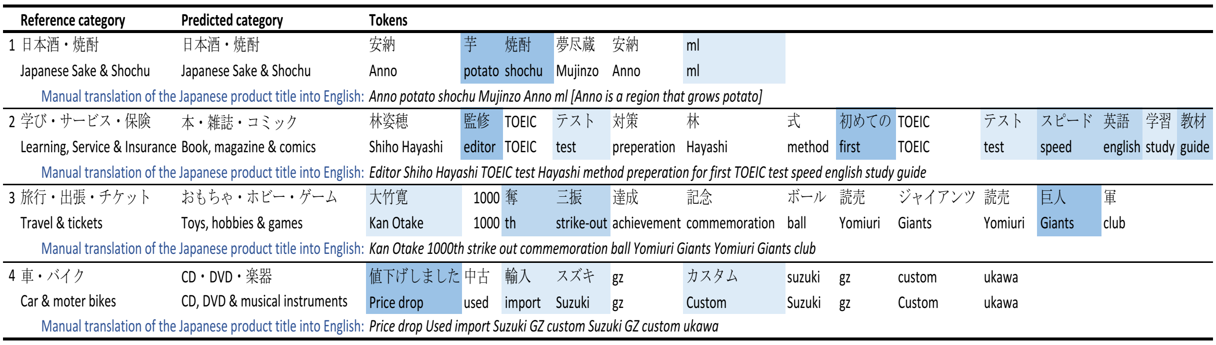
To this end, we computed the ratio of the sum of the number of listings in the test set per category for which both GBT and ACNN mis-classify but agree on the wrong predicted category, to the total number of mis-classifications from ACNN. The upper bound of this ratio is 1.0, which means that ACNN would make the same mistakes as GBT would. However, from our experiments, the ratio turned out to be 0.37, which means that GBT and ACNN make different mistakes more than 60% of the time. The relatively low value of the ratio indicates that we can gain major benefits for the final top level categorization by using an ensemble of GBT and ACNN models. The ACNN model does worse than GBT on 17 categories with a mean error difference, μ, of 0.78 and standard deviation, σ, of 1.15 and it does better than GBT on the rest of the 18 categories with $\mu = 0.39$ and $\sigma = 0.41$.

Statistics from test set	8000 **titles**	18 **categories**
Mean word count/title	20.930	20.120
Mean character count/word	09.245	05.815
Mean rare word count/title	00.158	00.354

Table 3: Word and character level statistics for: 1) The 8000 titles in the test set, for which ACNN predicts correctly over CNN (**Middle** column); and 2) The 18 categories in the test set for which ACNN performs better than GBT (**Rightmost** column).

Table 3 sheds some insights on why the ACNN model may be doing better over the CNN model, for the 8000 titles in the test set. We compare the average number of characters in the words of the 8000 titles in the test set for which our ACNN model provides correct predictions over the CNN model, to that for the overall test set from Table 1. The count for the former case turns out to be 9.245 that is substantially higher than that for the latter case, which is 5.799. It is thus highly likely that the ACNN model is performing better than CNN by leveraging the longer word and character contexts for these 8000 titles.

On the other hand, removal of the rare tokens (words appearing in less than 20 titles) seem to



	Reference category	Predicted category	Tokens													
1	日本酒・焼酎	日本酒・焼酎	安納	芋	焼酎	夢尽蔵	安納	ml								
	Japanese Sake & Shochu	Japanese Sake & Shochu	Anno	potato	shochu	Mujinzo	Anno	ml								
	Manual translation of the Japanese product title into English:		*Anno potato shochu Mujinzo Anno ml [Anno is a region that grows potato]*													
2	学び・サービス・保険	本・雑誌・コミック	林姿穂	監修	TOEIC	テスト	対策	林	式	初めての	TOEIC	テスト	スピード	英語	学習	教材
	Learning, Service & Insurance	Book, magazine & comics	Shiho Hayashi	editor	TOEIC	test	preperation	Hayashi	method	first	TOEIC	test	speed	english	study	guide
	Manual translation of the Japanese product title into English:		*Editor Shiho Hayashi TOEIC test Hayashi method preperation for first TOEIC test speed english study guide*													
3	旅行・出張・チケット	おもちゃ・ホビー・ゲーム	大竹寛	1000	奪	三振	達成	記念	ボール	読売	ジャイアンツ	読売	巨人	軍		
	Travel & tickets	Toys, hobbies & games	Kan Otake	1000	th	strike-out	achievement	commemoration	ball	Yomiuri	Giants	Yomiuri	Giants	club		
	Manual translation of the Japanese product title into English:		*Kan Otake 1000th strike out commemoration ball Yomiuri Giants Yomiuri Giants club*													
4	車・バイク	CD・DVD・楽器	値下げしました	中古	輸入	スズキ	gz	カスタム	suzuki	gz	custom	ukawa				
	Car & moter bikes	CD, DVD & musical instruments	Price drop	used	import	Suzuki	gz	Custom	Suzuki	gz	Custom	ukawa				
	Manual translation of the Japanese product title into English:		*Price drop Used import Suzuki GZ custom Suzuki GZ custom ukawa*													

Figure 2: Examples of attention tokens for correct and incorrect classifications with English translations for tokens, product titles, and categories. Gradient colors are coded by attention model weights. Darker shades of blue have higher attention.

have negligible effect on the context of the titles from the subset of 8000 titles. However, the effect is a little more pronounced for the context of the titles from the subset of the 18 categories for which ACNN does better than GBT, but, with a mean error difference of only *half* of that for the other 17 categories on which it does worse.

5.3 Paying Attention Pays Off!

One of the most important aspects of the ACNN model is the ability to highlight words and characters in sequential text tokens automatically through the attention mechanism. Examples of such selected word tokens from test titles can be observed in Fig. 2.

In order to visualize the importance of the words related to the categorization label contribution, we use the attention vectors (e.g., α' scores) generated by the model. The word attention scores accurately localize words that are closely related to the classification labels. For instance, in Figure 2, line 1, the first word highlighted in the product description (higher score) is *potato*, which is one of the main ingredients in the Japanese alcoholic beverage, Shōchū (焼酎), that is referred to in the product title.

For the second example, there is ambiguity between the reference and the predicted category since the product title can be applied to both. In this case, the attention model is highlighting words like *editor*, *English*, and *guide* that may apply to both *Learning services* and *Books*.

The third example in Fig. 2 is an annotation mistake that was correctly captured by the model. Here the attention model is extracting the salient words *Giants*, *strike-out*, and *Kan Otake*, which are related to the predicted category.

Finally, in the fourth example, the attention mechanism assigns high scores to the words *price drop*, *import*, and *Suzuki* where *Suzuki* is a popular car manufacturer and music curriculum in Japan. “Suzuki” is thus inherently ambiguous and our model fails to put attention on context clues like the token “gz”, which is a motorbike model.

6 Concluding Remarks

We propose a variant of the popular CNN model, the Attention CNN (ACNN) model, for the task of large-scale categorization of millions of Japanese product titles into thirty-five top level categories. The proposed model can leverage GPUs to **reduce training time** from three weeks for a state-of-the-art GBT classifier to three days while maintaining more than 96% accuracy.

Our language agnostic attention model can **highlight salient tokens**, which are semantically highly correlated to predicted categories. This helps in dimensionality reduction **without the need for feature engineering**.

As future work, we will experiment with ensemble methods to exploit differences in prediction errors from the different models, thereby improving overall classification performance.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794.

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, pages 1351–1360.

Pradipto Das, Yandi Xia, Aaron Levine, Giuseppe Di Fabbrizio, and Ankur Datta. 2017. Web-scale language-independent cataloging of noisy product listings for e-commerce. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, April 3-7, 2017, Valencia, Spain.*

Jerome H. Friedman. 2000. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Zornitsa Kozareva. 2015. 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, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1329–1333.

Y. LeCun and Y. Bengio. 1995. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press.

Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2015. Dependency-based convolutional neural networks for sentence embedding. In *Proceedings of the 53st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 174–179, Beijing, China, July. Association for Computational Linguistics.

Hyuna Pyo, Jung-Woo Ha, and Jeonghee Kim. 2016. Large-scale item categorization in e-commerce using multiple recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, New York, NY, USA. ACM.

Dan Shen, Jean David Ruvini, Manas Somaiya, and Neel Sundaresan. 2011. Item categorization in the e-commerce domain. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1921–1924, New York, NY, USA. ACM.

Dan Shen, Jean-David Ruvini, Rajyashree Mukherjee, and Neel Sundaresan. 2012a. A study of smoothing algorithms for item categorization on e-commerce sites. *Neurocomput.*, 92:54–60, September.

Dan Shen, Jean-David Ruvini, and Badrul Sarwar. 2012b. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 595–604, New York, NY, USA. ACM.

Chong Sun, Narasimhan Rampalli, Frank Yang, and AnHai Doan. 2014. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proc. VLDB Endow.*, 7(13), August.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057. JMLR Workshop and Conference Proceedings.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.

Hsiang-Fu Yu, Chia-Hua Ho, Yu-Chin Juan, and Chih-Jen Lin. 2013. LibShortText: A Library for Short-text Classication and Analysis. Technical report, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.