Web-Scale Language-Independent Cataloging of Noisy Product Listings for E-Commerce

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

Abstract

The cataloging of product listings through taxonomy categorization is a fundamental problem for any e-commerce marketplace, with applications ranging from personalized search recommendations to query understanding. However, manual and rule based approaches to categorization are not scalable. In this paper, we compare several classifiers for categorizing listings in both English and Japanese product catalogs. We show empirically that a combination of words from product titles, *navigational breadcrumbs*, and *list prices*, when available, improves results significantly. We outline a novel method using correspondence topic models and a lightweight manual process to reduce noise from mislabeled data in the training set. We contrast linear models, gradient boosted trees (GBTs) and convolutional neural networks (CNNs), and show that GBTs and CNNs yield the highest gains in error reduction. Finally, we show GBTs applied in a language-agnostic way on a large-scale Japanese e-commerce dataset have improved taxonomy categorization performance over current state-of-the-art based on deep belief network models.

1 Introduction

Web-scale e-commerce catalogs are typically exposed to potential buyers using a taxonomy categorization approach where each product is categorized by a label from the taxonomy tree. Most e-commerce search engines use taxonomy labels to optimize query results and match relevant listings to users' preferences (Ganti et al., 2010). To illustrate the general concept, consider Fig. 1. A merchant pushes new men's clothing listings to an online catalog infrastructure, which then organizes the listings into a taxonomy tree. When a user searches for a denim brand, *"DSquared2"*, the search engine first has to understand that the user is searching for items in the *"Jeans"* category. Then, if the specific items cannot be found in the inventory, other relevant items in the *"Jeans"* category are returned in the search results to encourage the user to browse further. However, achieving good product categorization for e-commerce market-places is challenging.

Commercial product taxonomies are organized in tree structures three to ten levels deep, with thousands of leaf nodes (Sun et al., 2014; Shen et al., 2012b; Pyo et al., 2016; McAuley et al., 2015). Unavoidable human errors creep in while uploading data using such large taxonomies, contributing to mis-labeled listing noise in the data set. Even EBay, where merchants have a unified taxonomy, reported a 15% error rate in categorization (Shen et al., 2012b). Furthermore, most e-commerce companies receive millions of new listings per month from hundreds of merchants composed of wildly different formats, descriptions, prices and meta-data for the *same products*. For instance, the two listings, *"University of Alabama all-cotton non iron dress shirt"* and *"U of Alabama 100% cotton no-iron regular fit shirt"* by two merchants refer to the same product.

E-commerce systems trade-off between classifying a listing directly into one of thousands of leaf node categories (Sun et al., 2014; **?**) and splitting the taxonomy at predefined depths (Shen et al., 2011; **?**) with smaller subtree models. In the latter case, there is another trade-off between the number of hierarchical subtrees and the propagation of error in the prediction cascade. Similar to (Shen et al., 2012b; Cevahir and Murakami, 2016), we classify product listings in two or three steps, depending on the taxonomy size. First, we predict the top-level category and then clas-

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 969–979,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

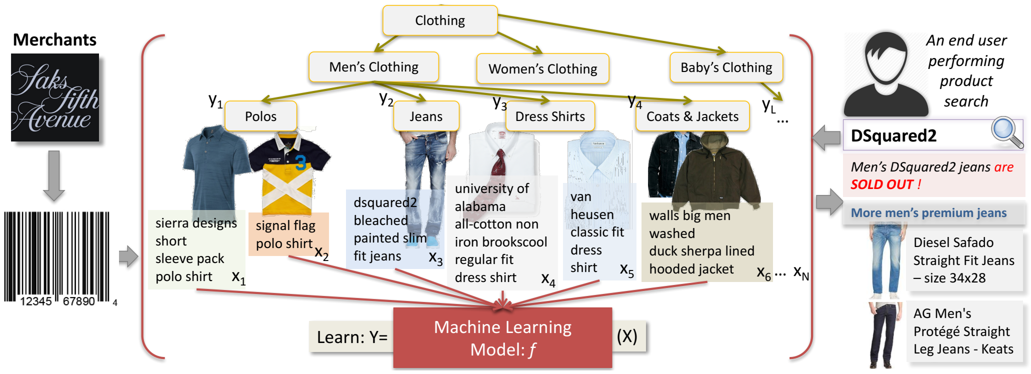

Merchants
Saks Fifth Avenue
12345 67890 4
Clothing
Men's Clothing
Women's Clothing
Baby's Clothing
y_1
Polos
y_2
Jeans
y_3
Dress Shirts
y_4
Coats & Jackets
y_L ...
sierra designs short sleeve pack polo shirt x_1
signal flag polo shirt x_2
dsquared2 bleached painted slim fit jeans x_3
university of alabama all-cotton non iron brookscool regular fit dress shirt x_4
van heusen classic fit dress shirt x_5
walls big men washed duck sherpa lined hooded jacket x_6 ... x_N
Learn: Y=
Machine Learning Model: *f*
(X)
An end user performing product search
DSquared2
Men's DSquared2 jeans are SOLD OUT !
More men's premium jeans
Diesel Safado Straight Fit Jeans – size 34x28
AG Men's Protégé Straight Leg Jeans - Keats

Figure 1: E-commerce platform using taxonomy categorization to understand query intent, match merchant listings to potential buyers as well as to prevent buyers from navigating away on search misses.

sify the listings using another one or two levels of subtree models selected by the previous predictions. For our large-scale taxonomy categorization experiments on product listings, we use two in-house datasets,[1] a publicly available Amazon product dataset (McAuley et al., 2015), and a publicly available Japanese product dataset.[2]

Our paper makes several contributions: 1) We perform large-scale comparisons with several robust classification methods and show that Gradient Boosted Trees (GBTs) (Friedman, 2000; **?**) and Convolutional Neural Networks (CNNs) (LeCun and Bengio, 1995; **?**) perform substantially better than state-of-the-art linear models (Section 5). We further provide analysis of their performance with regards to imbalance in our datasets. 2) We demonstrate that using both *listing price* and *navigational breadcrumbs* – the branches that merchants assign to the listings in web pages for navigational purposes – boost categorization performance (Section 5.3). 3) We effectively apply correspondence topic models to detect and remove mis-labeled instances in training data with minimal human intervention (Section 5.4). 4) We empirically demonstrate the effectiveness of GBTs on a large-scale Japanese product dataset over a recently published state-of-the-art method (Cevahir and Murakami, 2016), and in turn the otherwise language-agnostic capabilities of our system given a language-dependent word tokenization method.

2 Related Work

The nature of our problem is similar to those reported in (Bekkerman and Gavish, 2011; Shen et al., 2011; Shen et al., 2012b; Yu et al., 2013b; Sun et al., 2014; Kozareva, 2015; **?**), but with more pronounced data quality issues. 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.

The work in Sun et al. (2014) emphasizes the use of simple classifiers in combination with large-scale manual efforts to reduce noise and imperfections from categorization outputs. While human intervention is important, we show how unsupervised topic models can substantially reduce such expensive efforts for product listings crawled in the wild. Further, unlike Sun et al. (2014), we adopt stronger baseline systems based on regularized linear models (Hastie et al., 2003; Zhang, 2004; Zou and Hastie, 2005).

A recent work from Pyo et al. (2016) emphasizes the use of recurrent neural networks for taxonomy categorization purposes. Although, they mention that RNNs render unlabeled pre-training of word vectors (Mikolov et al., 2013) unnecessary, in contrast, we show that training word embeddings on the whole set of three product title corpora improves performance for CNN models and opens up the possibility of leveraging other product corpora when available.

Shen et al. (2012b) advocate the use of algorithmic splitting of the taxonomy using graph theoretic latent group discovery to mitigate data imbalance problems at the leaf nodes. They use a combination of k-NN classifiers at the coarser level and SVMs (Cortes and Vapnik, 1995) classifiers at the leaf levels. Their SVMs solve much easier k-way multi-class categorization problems where $k \in \{3, 4, 5\}$ with much less data imbalance. We, however, have found that SVMs do not work well in scenarios where k is large and the data is imbalanced. Due to our high-dimensional feature spaces, we avoided k-NN classifiers that can cause

[1] The in-house datasets are from Rakuten USA, managed by Rakuten Ichiba, Japan's largest e-commerce company.

[2] This dataset is from Rakuten Ichiba and is released under Rakuten Data Release program.

prohibitively long prediction times under arbitrary feature transformations (Manning et al., 2008; Cevahir and Murakami, 2016).

The use of a bi-level classification using k-NN and hierarchical clustering is incorporated in Cevahir and Murakami (2016)'s work, where they use nearest neighbor methods in addition to Deep Belief Networks (DBN) and Deep Auto Encoders (DAE) over both titles and descriptions of the Japanese product listing dataset. We show in Section 5.6, that using a tri-level cascade of GBT classifiers over titles, we significantly outperform the k-NN+DBN classifier on average.

3 Dataset Characteristics

We use two in-house datasets, named BU1 and BU2, one publicly available Amazon dataset (AMZ) (McAuley et al., 2015), and a Japanese product listing dataset named RAI (Cevahir and Murakami, 2016) (short for Rakuten Ichiba) for the experiments in this paper.

BU1 is categorized using human annotation efforts and rule-based automated systems. This leads to a high precision training set at the expense of coverage. On the other hand, for BU2, noisy taxonomy labels from external data vendors have been automatically mapped to an in-house taxonomy without any human error correction, resulting in a larger dataset at the cost of precision. BU2 also suffers from inconsistencies in regards to incomplete or malformed product titles and metadata arising out of errors in the web crawlers that vendors use to aggregate new listings. However, for BU2, the noise is distributed identically in the training and test sets, thus evaluation of the classifiers is not impeded by it.

The Japanese RAI dataset consists of $172,480,000$ records split across $26,223$ leaf nodes. The distribution of product listings in the leaf nodes is based on the popularity of certain product categories and is thus highly imbalanced. For instance, the top level "Sports & Outdoor" category has $2,565$ leaf nodes, while the "Travel / Tours / Tickets" category has only 38. The RAI dataset has 35 categories at depth one (level-one categories) and 400 categories at depth two of the full taxonomy tree. The total depth of the tree varies from three to five levels.

The severity of data imbalance for BU2 is shown in Figure 2. The top-level *"Home, Furniture and Patio"* subtree that accounts for almost half of the BU2 dataset. Table 1 shows dataset characteristics for the four different kinds of product datasets we use in our analyses. It lists the number of branches for the top-level taxonomy subtrees, the total number of branches ending at leaf nodes for which there are a non-zero number of listings and two important summary statistics that helps quantify the nature of imbalance. We first calculate the Pearson correlation coefficient (PCC) between the number of listings and branches in each of the top-level subtrees for each of the four datasets.

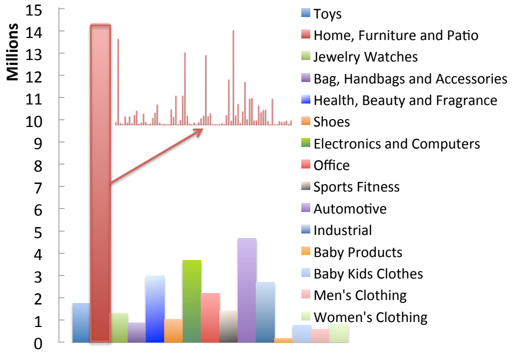


Figure 2: Top-level category distribution of 40 million deduplicated listings from an earlier Dec 2015 snapshot of BU2. Each category subtree is also imbalanced, as seen in exploded view of the *"Home, Furniture, and Patio"* category.

A perfectly balanced tree will have a PCC of 1.0. BU1 shows the most *benign* kind of imbalance with a PCC of 0.643. This confirms that the number of branches in the subtrees correlate well with the volume of listings. Both AMZ and RAI datasets show the highest branching factors in their taxonomies. For the AMZ dataset, it could be due to the fact that the crawled taxonomy is different from Amazon's internal catalog. The Rakuten Ichiba taxonomy has been incrementally adjusted to grow in size over several years by creating new branches to support newer and popular products. We observe that for RAI, AMZ and BU2 in particular, the number of branches in the subtrees do not correlate well with the volume of listings. This indicates a much higher level of imbalance.

Datasets	*Subtrees*	*Branches*	*Listings*	PCC	KL
BU1	16	1,146	12.1M	0.643	0.872
BU2	15	571	60M	0.209	0.715
AMZ	25	18,188	7.46M	0.269	1.654
RAI	35	26,223	172.5M	0.474	7.887

Table 1: Dataset properties on: total number of top-level category subtrees, branches and listings

We also compute the average Kullback-Leibler

(KL) divergence, $KL(p(\mathbf{x})|q(\mathbf{x}))$, (Cover and Thomas, 1991) between the empirical distribution over listings in branches for each subtree rooted in the nodes at depth one, $p(\mathbf{x})$, compared to a uniform distribution, $q(\mathbf{x})$. Here, the KL divergence acts as a measure of imbalance of the listing distribution and is indicative of the categorization performance that one may obtain on a dataset; high KL divergence leads to poorer categorization and vice-versa (see Section 5).

4 Gradient Boosted Trees and Convolutional Neural Networks

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, such as a decision tree $f(\mathbf{x})$ over $\mathbf{X}$. 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. To compute $F^\star(\mathbf{x})$, we need to calculate, for each boosting round m,

$$\{\mathbf{a}_m, \mathbf{w}_m\} = \arg\min_{\mathbf{a},\mathbf{w}} \sum_{i=1}^{N} L(y_i, F_m(\mathbf{x}_i)) \quad (1)$$

with $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + f_m(\mathbf{x}, \mathbf{a}_m, \mathbf{w}_m)$. This expression is indicative of a gradient descent step:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m\left(-g_m(\mathbf{x}_i)\right) \quad (2)$$

where ρ_m is the step length and $\left[\frac{\partial L(y,F(\mathbf{x}))}{\partial F(\mathbf{x})}\right]_{F(\mathbf{x}_i)=F_{m-1}(\mathbf{x}_i)} = g_m(\mathbf{x}_i)$ being the search direction. To solve $\mathbf{a}_m$ and $\mathbf{w}_m$, we make the basis functions $f_m(\mathbf{x}_i; \mathbf{a}, \mathbf{w})$ correlate most to $-g_m(\mathbf{x}_i)$, where the gradients are defined over the *training data* distribution. In particular, using Taylor series expansion, we can get closed form solutions for $\mathbf{a}_m$ and $\mathbf{w}_m$ – see Chen and Guestrin (2016) for details. It can be shown that $\mathbf{a}_m = \arg\min_{\mathbf{a}} \sum_{i=1}^{N} (-g_m(\mathbf{x}_i) - \rho_m f_m(\mathbf{x}_i, \mathbf{a}, \mathbf{w}_m))^2$ and $\rho_m = \arg\min_\rho \sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho f_m(\mathbf{x}_i; \mathbf{a}_m, \mathbf{w}_m))$ which yields,

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m f_m(\mathbf{x}, \mathbf{a}_m, \mathbf{w}_m) \quad (3)$$

Each boosting round m updates the weights $w_{m,j}$ on the leaves and helps create a new tree in the next iteration. The optimal selection of decision tree parameters is based on optimizing the $f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$ using a logistic loss. For GBTs, each decision tree is resistant to imbalance and outliers (Hastie et al., 2003), and $F(\mathbf{x})$ can approximate arbitrarily complex decision boundaries.

The convolutional neural network we use is based on the CNN architecture described in LeCun and Bengio (1995; Kim (2014) using the TensorFlow framework (Abadi and others, 2015). As in Kim (2014), we enhance the performance of "vanilla" CNNs (Fig. 3 **right**) using word embedding vectors (Mikolov et al., 2013) trained on the product titles from all datasets, without taxonomy labels. Context windows of width n, corresponding to n-grams and embedded in a 300 dimensional word embedding space, are convolved with L filters followed by rectified non-linear unit activation and a max-pooling operation over the set of all windows W. This operation results in a $L \times 1$ vector, which is then connected to a softmax output layer of dimension $K \times 1$, where K is the number of classes. Section A lists more details on parameters.

The CNN model tries to allocate as few filters to the context windows while balancing the constraints on the back-propagation of error residuals with regards to cross-entropy loss $\mathcal{L} = -\sum_{k=1}^{K} q_k \log p_k$, where p_k is the probability of a product title $\mathbf{x}$ belonging to class k predicted by our model, and $q \in \{0,1\}^K$ is a one-hot vector that represents the true label of title $\mathbf{x}$. This results in a higher predictive power for the CNNs, while still matching complex decision boundaries in a smoother fashion than GBTs. We note here that for all models, the predicted probabilities are *not* calibrated (Zadrozny and Elkan, 2002).

5 Experimental Setup and Results

We use Naïve Bayes (NB) (Ng and Jordan, 2001) similar to the approach described in Shen et al. (2012a) and Sun et al. (2014), and Logistic Regression (LogReg) classifiers with L_1 (Fan et al., 2008) and Elastic Net regularization, as robust baselines. Parameter setups for the various models and algorithms are mentioned in Section A.

5.1 Data Preprocessing

Product listing datasets in English – BU1 is exclusively comprised of product titles, hence, our features are primarily extracted from these titles. For AMZ and BU2, we additionally extract the list

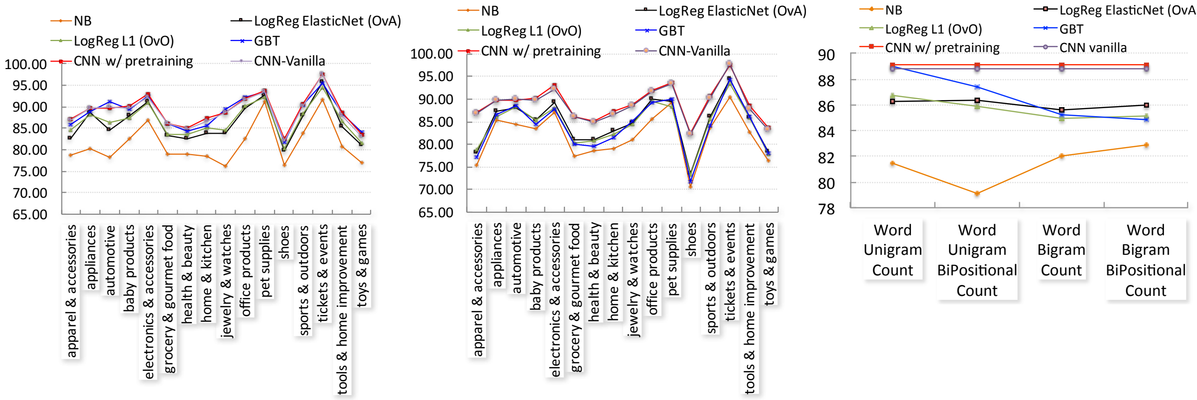


Figure 3: Classifier performance on BU1 test set. The CNN classifier has only one configuration and thus shows constant curves in all plots. **Left** figure shows prediction on 10% test set using word unigram count features; **middle** figure shows prediction on 10% test set using word bigram bi-positional count features; and the **right** figure shows mean micro-precision over different feature setups except CNNs. In all figures, "OvO" means "One vs. One" and "OvA" means "One vs All".

price whenever available. For BU2, we also use the leaf node of any available *navigational breadcrumbs*. In order to decrease training and categorization run times, we employ a number of vocabulary filtering methods. Further, English stopwords and rare tokens that appear in 10 listings or less are then filtered out. This reduces vocabulary sizes by up to 50%, without a significant reduction in categorization performance. For CNNs, we replace numbers by the nominal form `[NUM]` and remove rare tokens. We also remove punctuations and then lowercase the resulting text. Parts of speech (POS) tagging using a generic tagger from Manning et al. (2014) trained on English text produced very noisy features, as is expected for out-of-domain tagging. Consequently, we do not use POS features due to the absence of a suitable training set for listings unlike that in Putthividhya and Hu (2011). For GBTs, we also experiment with title word expansion using nearest neighbors from Word2Vec model (Mikolov et al., 2013), for instance, to group words like *"t-shirts"*, *"tshirt"*, *"t-shirt"* in their respective equivalence classes, however, the overall results have not been better.

Product listing datasets in Japanese – CJK languages like Japanese lack white space between words. Hence, the first pre-processing step requires a specific Japanese tokenization tool to properly segment the words in the product titles.

For our experiments, we used the MeCab[3] tokenizer 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 also use canonical normalization to change the code points of the resulting Japanese text into an NFKC normalized[4] form, then remove anything outside of standard Japanese UTF-8 character ranges. Finally, the resulting text is lowercased.

Due to the size of the RAI dataset taxonomy tree, three groups of models are trained to classify new listings into one of 35 level-one categories, then one of 400 level-two categories, and, finally, the leaf node of the taxonomy tree. We have found this scheme to be working better for the RAI dataset than a bi-level scheme that we adopted for the other English datasets.

Applying GBTs on the Japanese dataset involved a bit more feature engineering. 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. Consequently, character 2, 3, and 4-grams are extracted with their counts, where extractions include single spaces appearing at the end of word boundaries. Identification of the best set of feature combinations in this case has been performed during cross-validation.

5.2 Initial Experiments on BU1 dataset

Our initial experiments use unigram counts and three other features: word bigram counts, bi-positional unigram counts, and bi-positional bigram counts. Consider a title text *"120 gb hdd 5400rpm sata fdb 2 5 mobile"* from the *"Data*

[3] `https://sourceforge.net/projects/mecab/`

[4] `http://unicode.org/reports/tr15/`

storage” leaf node of the *Electronics* taxonomy subtree and another title text *“acer aspire v7 582pg 6421 touchscreen ultrabook 15 6 full hd intel i5 4200u 8gb ram 120 gb hdd ssd nvidia geforce gt 720m”* from the *“Laptops and notbooks”* leaf node. In such cases, we observe that merchants tend to place terms pertaining to storage device specifics in the front of product titles for *“Data storage”* and similar terms towards the end of the titles for *“Laptops”*. As such, we split the title length in half and augment word uni/bigrams with a left/right-half position.

This makes sense from a Naïve Bayes point of view, since terms like *“120_gb”*[Left_Half], *“gb_hdd”*[Left_Half], *“120_gb”*[Right_Half] and *“gb_hdd”*[Right_Half] de-correlates the feature space better, which is suitable for the naïve assumption in NB classification. This also helps in sightly better explanation of the class posteriors. These assumptions for NB are validated in the three figures: Fig. 3 **left**, Fig. 3 **middle** and Fig. 3 **right**. Word unigram count features perform strongly for all classifiers except NB, whereas bi-positional word bigram features helped only NB significantly.

Additionally, the micro-precision and F1 scores for CNNs and GBTs are significantly higher compared to other algorithms on word unigrams using paired t-test with a p-value < 0.0001. The performances of GBTs and LogReg L_1 classifiers deteriorate over the other feature sets as well. The bi-positional and bigram feature sets also do not produce any improvements for the AMZ dataset. Based on these initial results, we focus on word unigrams in all of our subsequent experiments.

5.3 Categorization Improvements with Navigational Breadcrumbs and List Prices on BU2 Dataset

BU2 is a challenging dataset in terms of class imbalance and noise and we sought to improve categorization performance using available meta-data. To start, we experiment with a smaller dataset consisting of $\approx 500,000$ deduplicated listings under the *“Women’s Clothing”* taxonomy subtree, extracted from our Dec 2015 snapshot of 40 million records. Then we train and test against $\approx$ 2.85 million deduplicated *“Women’s Clothing”* listings from the Feb 2016 snapshot of BU2. In all experiments, 10% of the data is used as test set. The womens clothing category had been chosen due to the importance of the category from a business standpoint, which provided early access to listings in this category. Further, data distributions remain the same in the two snapshots and the Feb 2016 snapshot consists of listings in addition to those for the Dec 2015 snapshot.

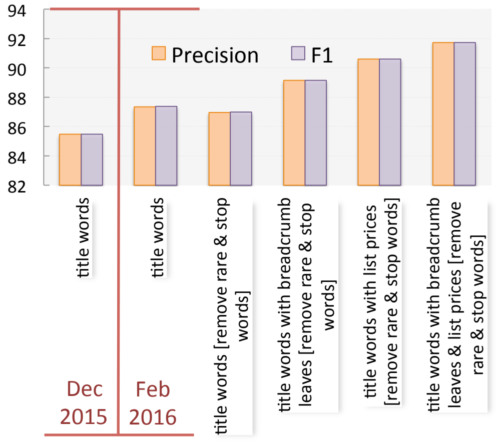


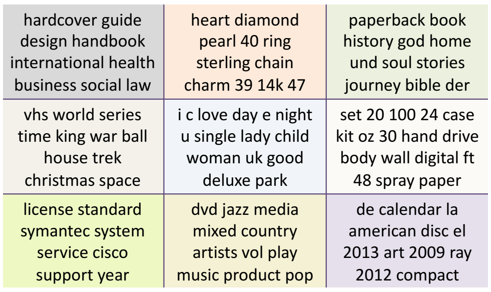
Figure 4: Improvements in micro-precision and F1 for GBTs on BU2 dataset for *“Women’s Clothing”* subtree

The first noteworthy fact in Fig. 4 is that the micro-precision and F1 of the GBTs substantially improve after increasing the size of the dataset. Further, stop words and rare words filtering decrease precision and F1 by less than 1%, despite halving the feature space. The addition of navigational leaf nodes and list prices prove advantageous, with both features independently boosting performance and raising micro-precision and F1 to over 90%. Despite finding similar gains in categorization performance for other top-level subtrees by using these meta features, we needed a system to filter mis-categorized listings from our training data as well.

5.4 Noise Analysis of BU2 Dataset using Correspondence LDA for Text

The BU2 dataset has the noisiest ground-truth labels, as incorrect labels have been assigned to product listings. However, since the manual verification of millions of listings is infeasible, using some proxy for ground truth is a viable alternative that has previously produced encouraging results (Shen et al., 2012b). We next describe how resorting to unsupervised topic models helped to detect and remove incorrect listings.

As shown in Fig. 8, categorization performance for the *“Shoes”* taxonomy subtree is over 25 points below the *“Women’s Clothing”* category. Such a large difference could be caused by incorrect assignments of listings to the correct cat-



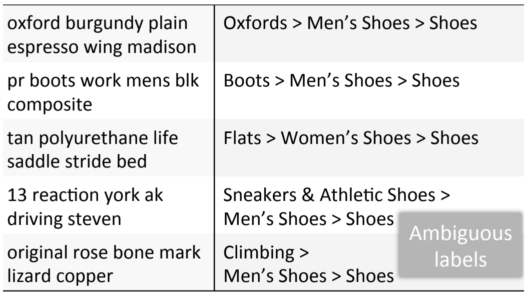
hardcover guide design handbook international health business social law	heart diamond pearl 40 ring sterling chain charm 39 14k 47	paperback book history god home und soul stories journey bible der
vhs world series time king war ball house trek christmas space	i c love day e night u single lady child woman uk good deluxe park	set 20 100 24 case kit oz 30 hand drive body wall digital ft 48 spray paper
license standard symantec system service cisco support year	dvd jazz media mixed country artists vol play music product pop	de calendar la american disc el 2013 art 2009 ray 2012 compact

Figure 5: Selection of most probable words under the latent "noise" topics over listings in *"Shoes"* subtree. Human annotators inspect whether such sets of words belong to a Shoes topic or not.

egories. However, unlike Sun et al. (2014), as there are over 3.4 million *"Shoes"* listings in the BU2 dataset, a manual analysis to detect noisy labels is infeasible. To address this problem, we compute $p(\mathbf{x})$ over latent topics z_k, and automatically annotate the most probable words over each topic.

We choose our CorrMMLDA model (Das et al., 2011) to discover the latent topical structure of the listings in the *"Shoes"* category because of two reasons. Firstly, the model is a natural choice for our scenario since it is intuitive to assume that store and brand names are distributions over words in titles. This is illustrated in the graphical model in Fig. 7, where the store *"Saks Fifth Avenue"* and the brand *"Joie"* are represented as words $w_{d,m}$ in the M plate of listing d and are distributions over the words in the product title *"Joie kidmore Embossed slipon sneakers"* represented as words $w_{d,n}$ in the N plate of the same listing d. The title words are in turn distributions over the latent topics $\mathbf{z}_d$ for listing $d \in \{1..D\}$.

Secondly, the CorrMMLDA model has been shown to exhibit lower held-out perplexities indicative of improved topic quality. The reason behind the lower perplexity stems from the following observations: Using the notation in Das et al. (2011), we denote the free parameters of the variational distributions over words in brand and store names, say $\lambda_{d,m,n}$, as multinomials over words in the titles and those over words in the title, say $\phi_{d,n,k}$, as multinomials over latent topics $\mathbf{z}_d$. It is easy to see that the posterior over the topic $z_{d,k}$ for each $w_{d,m}$ of brand and store names, is dependent on λ and ϕ through $\sum_{n=1}^{N_d} \lambda_{d,m,n} \times \phi_{d,n,k}$. This means that if a certain topic $\mathbf{z}_d = j$ generates all words in the title, i.e., $\phi_{d,n,j} > 0$, then



oxford burgundy plain espresso wing madison	Oxfords > Men's Shoes > Shoes
pr boots work mens blk composite	Boots > Men's Shoes > Shoes
tan polyurethane life saddle stride bed	Flats > Women's Shoes > Shoes
13 reaction york ak driving steven	Sneakers & Athletic Shoes > Men's Shoes > Shoes
original rose bone mark lizard copper	Climbing > Men's Shoes > Shoes

Ambiguous labels

Figure 6: Interpretation of latent topics using predictions from a GBT classifier. The topics here do not include those in Fig. 5, but are all from Feb 2016 snapshot of the BU2 dataset.

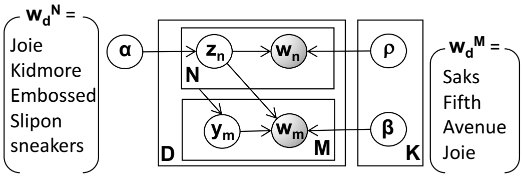


Figure 7: Correspondence MMLDA model.

only that topic also generates the brand and store names thereby increasing likelihood of fit and reducing perplexity. The other topics $\mathbf{z}_d \neq j$ do not contribute towards explaining the topical structure of the listing d.

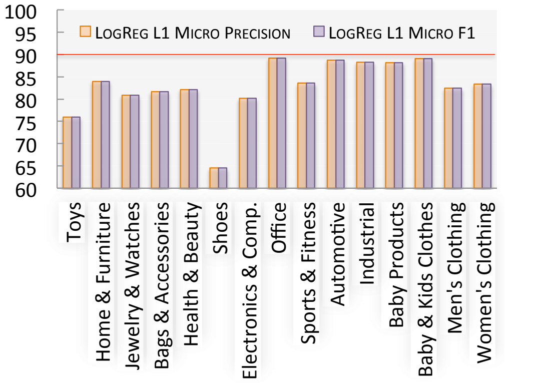


Figure 8: Micro-precision and F1 across fifteen top-level categories on 10% (4 million listings) of Dec 2015 BU2 snapshot.

We train the CorrMMLDA model with K=100 latent topics. A sample of nine latent topics and their most probable words shown in Fig. 5 demonstrates that topics outside of the *"Shoes"* domain can be manually identified, while reducing human annotation efforts from 3.4 million records to one hundred. We choose $K = 100$ since it is roughly twice the number of branches for the Shoes subtree. This choice provides modeling flexibility while respecting the number of ground

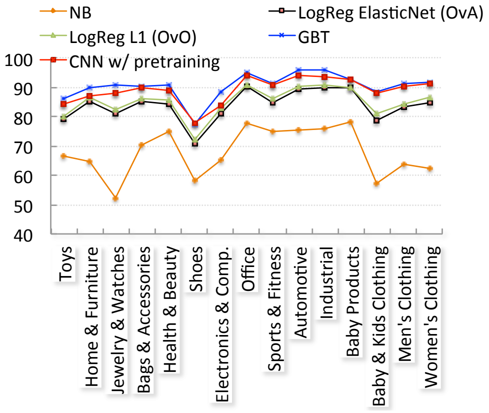


Figure 9: Micro-precision on 10% of BU2 across categories (see Sect. 5.4)

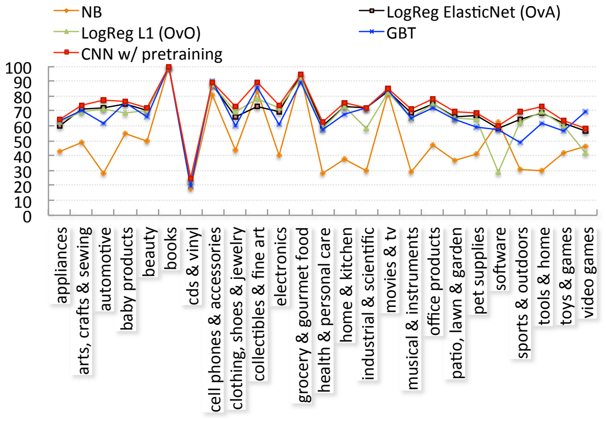


Figure 10: Micro-precision on 10% of AMZ across categories

Dataset	NB	LogReg ElasticNet	LogReg L1	GBT	CNN w pretraining	Mean KL	$\log(N/B)$
BU1	81.45	86.30	86.75	89.03*	89.12*	0.872	9.27
BU2	68.21	84.29	85.01	90.63*	88.67	0.715	11.54
AMZ	49.01	69.39	66.65	67.17	72.66*	1.654	6.02

Table 2: Mean micro-precision on 10% test set from BU1, BU2 and AMZ English datasets

truth classes.

We next run a list of the most probable six words, the average length of a *"Shoes"* listing's title, from each latent topic through our GBT classifier trained on the full, noisy data, *but without considering any metadata*, due to bag-of-words nature of the topic descriptions. As shown in the bottom two rows in Fig. 6, categories mismatching their topics are manually labeled as ambiguous. As a final validation, we uniformly sampled a hundred listings from each ambiguous topic detected by the model. Manual inspections revealed numerous listings from merchants not selling shoes are wrongly cataloged in the *"Shoes"* subtree due to vendor's error. To this end, we remove listings corresponding to such *"out-of-category"* merchants from all top-level categories.

Thus, by manually inspecting $K \times 6$ most probable words from the K=100 topics and $J \times 100$ listings, where $J << K$, instead of 3.4 million, a few annotators accomplished in hours what would have taken hundreds of annotators several months according to the estimates in Sun et al. (2014).

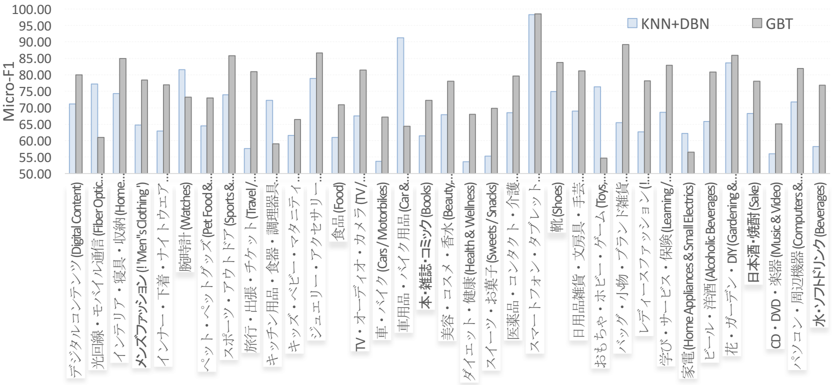
5.5 Results on BU2 and AMZ Datasets

In section 5.2, we have shown the efficacy of word unigram features on the BU1 dataset. Figure 8 shows that LogReg with L_1 regularization (Yu et al., 2013b; Yu et al., 2013a) initially achieves 83% mean micro-precision and F1 on the initial BU2 dataset. This falls short of our expectation of achieving an overall 90% precision (red line in Fig. 8), but forms a robust baseline for our subsequent experiments with the AMZ and the cleaned BU2 datasets. We additionally use the list price and the navigational breadcrumb leaf nodes for the BU2 dataset and, when available, the list price for the AMZ dataset.

Overall, Naïve Bayes, being an overly simplified generative model, generalizes very poorly on all datasets (see Figs. 3, 9 and 10). A possible option to improve NB's performance is to use sub-sampling techniques as described in Chawla et al. (2002); however, sub-sampling can have its own problems for when dealing with product datasets (Sun et al., 2014).

From Table 2, we observe that most classifiers tend to perform well when $\log(N/B)$ is relatively high. The N in the previous ratio is the total number of listings and B is the total number of categories. Figures with a $*$ are statistically better than other non-starred ones in the same row except the last two columns. From Fig. 9 and Table 2, it is clear that GBTs are better on BU2.

We also experiment with CNNs augmented to use meta-data while respecting the convolutional constraints on title text, however, the performance improved only marginally. It is not immediately clear why all the classifiers suffer on the *"CDs and Vinyl"* category, which has more than 500 branches – see Fig. 10. The AMZ dataset also suffers from novel cases of data imbalance. For instance, most of the listings in "*Books*" and "*Gro-*



Important statistics for the Rakuten Ichiba dataset and classifier performance:

- Average PCC at level one – 0.47
- Average KL at level one – 7.88
- Average KL at level two – 3.79
- Average of the micro-F1 scores across level one categories shown:
  - KNN+DBN – 73.85
  - GBT – **76.89***
- GBT performs significantly better than KNN+DBN for 28 out of 35 level one categories. There is one other category, *Smartphones and Tablets*, where GBT performs only slightly better than KNN+DBN.

Figure 11: Comparison of GBTs versus the method from Cevahir and Murakami (2016) on a 10% test set from the Rakuten Ichiba Japanese product listing dataset.

cery" are in one branch, with most other branches containing less than 10 listings. In summary, from both Figs. 9 and 10, we observe that GBTs and CNNs with pre-training perform best even in extreme data imbalance. It is possible that GBTs need finer parameter tuning per top-level subtree for datasets resembling AMZ.

5.6 Results on Rakuten Ichiba Dataset

In this section, we report our findings on the efficacy of GBTs vis-a-vis another hybrid nearest neighbor and deep learning based method from Cevahir and Murakami (2016). Our decision to employ a tri-level classifier cascade, instead of the bi-level one used for the other datasets, stems from our observations of the KL divergence values (see Section 3 and Table 1) at the first and second level depths of the RAI taxonomy tree. Moving from the first level down to the second decreases the KL divergence by more than 50%. We thus expect GBTs to perform better due to this reduced imbalance. We also cross-validated this assumption on some popular categories, such as *"Clothing"*.

From Fig. 11 and the statistics noted therein, we observe that, on average, GBTs outperform the KNN+DBN model from Cevahir and Murakami (2016) by 3 percentage points across all top level categories, which is statistically significant under a paired t-test with $p < 0.0001$. As with previous experiments, only a common best parameter configuration has been set for GBTs, without resorting to time consuming cross-validation across all categories. For the 29 categories on which GBTs do better, the mean of the absolute percentage improvement is 11.78, with a standard deviation of 5.07. Also, it has been observed that GBTs **significantly** outperform KNN+DBN in 28 of those categories.

The comparison in Fig. 11 is more holistic. Unlike the top level categorization scores obtained in Figs. 3, 9 and 10, the scores in Fig. 11 have been obtained by categorizing each test example through the entire cascade of hierarchical models for two classifiers. Even with this setting, the performance of GBTs is significantly better.

6 Conclusion

Large-scale taxonomy categorization with noisy and imbalanced data is a challenging task. We demonstrate deep learning and gradient tree boosting models with operational robustness in real industrial settings for e-commence catalogs with several millions of items. We summarize **our contributions** as follows: 1) We conclude that GBTs and CNNs can be used as new state-of-the-art baselines for product taxonomy categorization problems, **regardless of the language used**; 2) We quantify the nature of imbalance for different product datasets in terms of distributional divergence and correlate that to prediction performance; 3) We also show evidence to suggest that words from product titles, together with leaf nodes from navigational breadcrumbs and list prices, when available, can boost categorization performance significantly on all the product datasets. Finally, 4) we showcase a novel use of topic models with minimal human intervention to clean large amounts of noise particularly when the source of noise cannot be controlled. This is unlike any experiment reported in previous publications on product categorization. Automatic topic labeling for a given category with a pre-trained classifier from another dataset can help create an initial taxonomy over listings for which none exist. A major benefit of this approach is that it reduces manual efforts on initial taxonomy creation.

References

Martn Abadi et al. 2015. Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Ron Bekkerman and Matan Gavish. 2011. High-precision phrase-based document classification on a modern scale. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 231–239, New York, NY, USA. ACM.

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*, pages 525–535, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June.

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.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September.

Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA.

Pradipto Das, Rohini Srihari, and Yun Fu. 2011. Simultaneous joint and conditional modeling of documents tagged from two perspectives. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1353–1362, New York, NY, USA. ACM.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June.

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

Venkatesh Ganti, Arnd Christian König, and Xiao Li. 2010. Precomputing search features for fast and accurate query classification. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2003. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, August.

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.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 785–794, New York, NY, USA. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Andrew Ng and Michael Jordan. 2001. On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14.

Duangmanee (Pew) Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1557–1567, Stroudsburg, PA, USA. 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.

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

Hsiang-Fu Yu, Chia hua Ho, Prakash Arunachalam, Manas Somaiya, and Chih jen Lin. 2013b. Product title classification versus text classification. Technical report, UTexas, Austin; NTU; EBay.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 694–699, New York, NY, USA. ACM.

Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML 2004: Proceedings of the 21st International Conference on Machine Learning*, pages 919–926.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.

A Supplemental Materials: Model Parameters

In this paper, the baseline classifiers comprise of Naïve Bayes (NB) (Ng and Jordan, 2001) similar to the approach described in Shen et al. (2012a) and Sun et al. (2014), and Logistic Regression (LogReg) classifiers with L_1 (Fan et al., 2008) and Elastic Net regularization. The objective functions of both GBTs and CNNs involve L_2 regularizers over the set of parameters. Our development set for parameter tuning is generated by randomly selecting 10% of the listings under the *"apparel / clothing"* categories. The optimized parameters obtained from this scaled-down configuration is then extended to all other classifiers to reduce experimentation time.

For parameter tuning, we set a linear combination of 15% L_1 regularization and 85% L_2 regularization for Elastic Net. For GBTs (Chen and Guestrin, 2016) on both English and Japanese data, we limit each decision tree growth to a maximum depth of 500 and the number of boosting rounds is set to 50. Additionally, for leaf node weights, we use L_2 regularization with a regularization constant of 0.5. For GBTs on English data, the initial learning rate is 0.2. For GBTs on Japanese data, the initial learning rate is assigned a value of 0.05 .

For CNNs, we use context window widths of sizes $1, 3, 4, 5$ for four convolution filters, a batch size of 1024 and an embedding dimension of 300. The parameters for the embeddings are non-static. The convolutional filters are initialized with Xavier initialization (Glorot and Bengio, 2010). We use mini-batch stochastic gradient descent with Adam optimizer (Kingma and Ba, 2014) to perform parameter optimization.

LogReg classifiers and CNN need data to be normalized along each dimension, which is not needed for NB and GBT.