Neurocomputing 92 (2012) 54–60

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A study of smoothing algorithms for item categorization on e-commerce sites

Dan Shen[a,*], Jean-David Ruvini[b], Rajyashree Mukherjee[b], Neel Sundaresan[b]

[a] eBay Research Labs, No. 88 KeYuan Rd, Shanghai, China
[b] eBay Research Labs, 2005 E Hamilton Ave, San Jose, CA 95125, United States

ARTICLE INFO

Available online 22 February 2012

Keywords:
Item categorization
Catalog
Smoothing
Hierarchy

ABSTRACT

One central issue in a long-tail online marketplace such as eBay is to automatically put user self-input items into a catalog in real time. This task is extremely challenging when the inventory scales up, the items become ephemeral, and the user input remains noisy. Indeed, catalog learning has emerged as a key technical property for other major online e-commerce applications including search and recommendation. We formulate the item cataloging task as a Bayesian classification problem, which shall scale well in very large data set and have good online prediction performance. The inherent data sparseness issue, especially for those tail categories, is key to the overall model performance. We address the data sparseness issue by adapting statistically sound smoothing methods well studied in language modeling tasks. However, there are data characteristics specific to the e-commerce domain, including short yet focused item description, very large and hierarchical catalog taxonomy, and highly skewed distribution over types of items. We investigate these domain-specific regularities empirically, and report practically significant results with real-world true-scale data. Furthermore, we propose a two-stage learning strategy which effectively leverages domain-specific hierarchical catalog taxonomy. The experiment shows that the two-stage learning significantly outperforms the conventional learning by 2.4% precision increase.

© 2012 Published by Elsevier B.V.

1. Introduction

Online commerce has gained popularity since the past decade. eBay, one of the largest online C2C marketplaces, features a very large and long-tail inventory with millions of items entered to the marketplace everyday. To manage items effectively and help buyers find them easily, eBay organizes items into fine-grained categories structured as a topic hierarchy. For an e-commerce site that features a long-tail inventory, it is critical to assign items to proper categories. Item categorization will impact the exposure chances of items through either searching or browsing, and further influence the purchase probabilities as well as users' experiences. The task of item categorization can be formulated as a text classification problem intuitively. However, it is more difficult compared with traditional text classification tasks due to the nature of a long-tail dynamic online marketplace, specifically:

- *Large collection of categories*: There are more than 20,000 leaf categories on the eBay site. Specially, some sibling leaf categories are quite hard to distinguish from each other since they share a large portion of common words in their vocabularies, such as the category "*Sports Mem, Cards & Fan Shop* → *Manufacturer Authenticated* → *Basketball-NBA*", the category "*Sports Mem, Cards & Fan Shop* → *Fan Apparel & Souvenirs* → *Basketball-NBA*" and the category "*Sports Mem, Cards & Fan Shop* → *Autographs-Original* → *Basketball-NBA*".
- *Short and noisy text descriptions of items*: When seller lists an item on eBay, he or she is asked to write a title to briefly describe the item. Once the title is submitted, the site will be capable of suggesting the most proper categories for the item. So the item categorization is only based on the title. As opposed to free text, eBay limits the length of item title in less than 50 characters (about 10 words on average), which might not be adequate to describe an item well. Furthermore, the quality of title descriptions varies a lot on the eBay site. Some inexperienced or vicious sellers give inaccurate or fraud titles, which makes our data set quite noisy.
- *Scalability and efficiency*: As a large-scale online system for e-commerce sites, it first has to train a model offline on millions of items and then has to return a real-time response for an item in a few milliseconds. So good scalability and high efficiency become an imperative for the system.

[*] Corresponding author.
E-mail addresses: dashen@ebay.com (D. Shen), jruvini@ebay.com (J.-D. Ruvini), rmukherjee@ebay.com (R. Mukherjee), nsundaresan@ebay.com (N. Sundaresan).

0925-2312/$ - see front matter © 2012 Published by Elsevier B.V.
doi:10.1016/j.neucom.2011.08.035

Some statistical approaches, such as TFIDF [1] and Naive Bayes [2,3], have demonstrated the effectiveness and efficiency for classifying text documents. They typically represent documents as vectors of words, and learn by gathering statistics from the observed frequencies of the words within the documents from different classes. Because these approaches rely on the learned word statistics, they are data-intensive. That is, they require adequate labeled documents for each class to learn reliable frequency statistics, which guarantees high classification accuracy. However, it is observed in our domain that the distribution of items over categories, as shown in Section 3.2, is extremely unbalanced although eBay has a huge item inventory. Items for some categories are too sparse, which causes severe damage to the performance of such data-intensive approaches.

This paper aims to study how to effectively categorize items by employing the Bayesian algorithm. We specially explore the scaling-up of the learning algorithm to cope with highly skewed item distribution by leveraging state-of-the-art smoothing algorithms established in language modeling, including Laplace Smoothing, Jelinek–Mercer Smoothing, Dirichlet Priors, Absolute Discounting and Shrinkage Smoothing. The research issues that motivate this work are: (1) How the various smoothing methods perform on our task? (2) How sensitive is the performance of prediction to the parameter values of smoothing? (3) How does smoothing algorithm interact with the size of training data set, the size of category and the word specificity of category? Our experimental results show that there are interesting domain-specific statistical regularities that can guide our design decisions in terms of smoothing methods and parameters, e.g., the data-sparseness and vocabulary focusedness of different categories. Moreover, we study how to take advantage of the domain-specific information. Since the categories on eBay site are related with each other and represented as a hierarchical structure, we propose a two-stage learning strategy by using the hierarchical catalog information. The experiment shows that the two-stage learning can significantly outperform the conventional learning by 2.4% precision increase.

The item categorization model will be applied to determine the most proper category or recommend the top N categories for sellers' consideration when they list items on eBay. Moreover, the work can be further adapted to detect outlier items in categories which will be of help for fraud item detection. To our best knowledge, it is among the first to present the large-scale item categorization on e-commerce sites. It might be regarded as a benchmark for future studies in this domain.

The remainder of this paper is structured as follows. We first introduce the probabilistic approach to item classification and present various smoothing algorithms in this context. Next, we analyze category hierarchy and item distribution on the eBay site. Then we show and discuss experimental results. After that, we propose a two-stage learning strategy. Finally we conclude the paper with future work.

2. Methods

2.1. Bayesian learning framework

The task of item categorization is formalized in Bayesian learning framework. We employ the Naive Bayes with Multinomial likelihood function [4] which is to find the most likely class c^* with the maximum posterior probability of generating item t:

$$c^* = \arg\max_{c \in \mathcal{C}} P(c|t) = \arg\max_{c \in \mathcal{C}} P(c) \prod_{f_i \in t} P(f_i|c),$$

where $P(c)$ is the prior probability of the class c. It indicates the relative frequency of c and is estimated as $P(c) = N_c/N$, in which N_c is the number of items belonging to c and N is the total number of items. The conditional probability $P(f_i|c)$ is the weight that indicates how good a feature f_i is for c. In the Multinomial model, the maximum likelihood estimate (MLE) of $P(f_i|c)$ is obtained as follows:

$$p(f_i|c) = \frac{T_{cf_i}}{\sum_{f' \in F} T_{cf'}},$$

where T_{cf_i} is the number of occurrences of f_i in c, including multiple occurrences of the feature in an item and F is the set of all features in the vocabulary.

Previous works [2,3] has found that the Multinomial Naive Bayes model performs quite well empirically. It becomes a good choice especially for large-scale online data classification because of its foundation in statistical theory, simple implementation, great scalability and efficient computation. However, this kind of maximum likelihood-based learning algorithms is data intensive: they require adequate training data for each class to learn reliable frequency statistics. Data sparseness, such as unseen features or outlier features, will result in unreliable estimates which hurt performance badly. Unfortunately, our system also suffers from the data sparseness for many categories due to extremely unbalanced item distribution, as discussed in Section 3.2.

Smoothing has become standard for maximum likelihood models to cope with the problem of sparseness and unbalanced distribution. Indeed, many maximum likelihood models are centered around the issue of smoothing. At the very least purpose, it is to eliminate zero probability to unseen features. More importantly, it is capable of offering more robust estimates of parameters that would otherwise be uncertain due to limited amounts of training data. Zhai and Lafferty [5] state that smoothing accuracy directly impacts the performance of classification. Therefore, this paper will focus on studying smoothing of the Multinomial Naive Bayes model in the context of large-scale item categorization. In the following sections we present a few smoothing methods well studied in language modeling and investigate their behaviors in the context of item recognition for e-commerce.

2.2. Smoothing algorithms

We investigate various smoothing methods, including Laplace Smoothing, Jelinek–Mercer Smoothing, Dirichlet Priors, Absolute Discounting and Shrinkage Smoothing in the context of item categorization. For the requirement of efficient computation, the choice of smoothing algorithms is constrained by its efficiency.

Laplace Smoothing. It is the simplest smoothing method which is only motivated to eliminate zero probability to unseen words. It is also called add-one smoothing which simply adds one to each count when estimating $p(f_i|c)$

$$p(f_i|c) = \frac{T_{cf_i}+1}{\sum_{f' \in F}(T_{cf'}+1)} = \frac{T_{cf_i}+1}{(\sum_{f' \in F} T_{cf'})+B},$$

where $B = |F|$ is the number of features in the feature vocabulary F. Laplace Smoothing can be more rigorously interpreted as imposing a uniform prior, that is, each feature occurs once *a priori* for each class.

Unlike Laplace Smoothing, which adds an extra count to every word, more sophisticated smoothing algorithms treat features with different counts differently. They use the probability of feature f_i in the whole collection C (the collection model) as a "fallback" to the probability of the feature in given class c (the maximum likelihood model). The following three smoothing algorithms, including Jelinek–Mercer Smoothing, Dirichlet Priors and Absolute Discounting [5] are well-known.

Jelinek–Mercer Smoothing: Jelinek–Mercer Smoothing incorporates the maximum likelihood model $p(f_i|c)$ and the collection model $p(f_i|C)$ using a linear interpolation where a coefficient λ controls the influence of the two models

$$p_\lambda(f_i|c) = (1-\lambda)p(f_i|c)+\lambda p(f_i|C).$$

Dirichlet Priors: As the generalized case of Laplace Smoothing, Dirichlet Priors for Bayesian analysis estimates feature probability as

$$p_\mu(f_i|c) = \frac{T_{cf_i}+\mu p(f_i|C)}{\sum_{f' \in F}T_{cf'}+\mu},$$

where the scalar μ controls how strong we believe in the priors from the collection.

Absolute Discounting: Absolute Discounting is to reduce the probability of seen features by subtracting a constant δ from their counts:

$$p_\delta(f_i|c) = \frac{\max(T_{cf_i}-\delta,0)}{\sum_{f' \in F}T_{cf'}}+\frac{\delta|c|_u}{|c|}p(f_i|C),$$

where $\delta \in [0,1]$ is a discount constant, $|c|_u$ is the number of unique features in the class c and $|c|$ is the total number of features in c, so that $|c| = \sum_{f' \in F}T_{cf'}$.

Shrinkage Smoothing: Even more sophisticated smoothing algorithm, Shrinkage Smoothing [6] is further studied. The category structure in our study of eBay data is hierarchical in nature. Different from the classification on flat categories, eBay categories are arranged in a hierarchy. To obtain more robust parameter estimates, the model leverages the available category hierarchy by shrinking parameter estimates in data-sparse children categories toward the estimates in data-rich ancestor categories. The approach is also employed in *deleted interpolation*, a technique for smoothing n-grams in language modeling for speech recognition [7]. McCallum et al. [6] has stated that the method scales well on large data set with numerous categories in hierarchy and significantly outperforms traditional flat classifiers.

The core idea is to estimate $p(f_i|c_j)$ using the linear interpolation of a set of probabilities along the path from leaf category to root category.

$$p(f_i|c_j) = \lambda_j^0 p^0(f_i|c_j)+\lambda_j^1 p^1(f_i|c_j)+\lambda_j^2 p^2(f_i|c_j) \\ +\cdots+\lambda_j^{K-1}p^{K-1}(f_i|c_j)+\lambda_j^K\theta(f_i),$$

where $p^0(f_i|c_j)$ is the MLE at the leaf category c_j; $p^k(f_i|c_j)$ $(k=1,\ldots,K-1)$ is the MLE at the kth ancestor of c_j; and $\theta(f_i) = 1/|F|$ ($|F|$ is the size of feature vocabulary) is the uniform estimate of the feature f_i. The interpolation weights among the leaf category c_j and its ancestors are written as $\{\lambda_j^0,\lambda_j^1,\ldots,\lambda_j^K\}$, where $\sum_{k=0}^K \lambda_j^k = 1$.

We empirically derive the optimal weights, λ_j^k, between the ancestors of c_j by finding the weights that maximize the likelihood of training data. A form of Expectation Maximization (EM) [8] is applied to search the optimal set $\{\lambda_j^0,\lambda_j^1,\ldots,\lambda_j^K\}$ for the leaf category c_j which maximizes the total likelihood using an iterative procedure

$$\text{E-step}: \beta_j^k = \sum_{f_i \in H_j}\frac{\lambda_j^k p^k(f_i|c_j)}{\sum_k \lambda_j^k p^k(f_i|c_j)},$$

$$\text{M-step}: \lambda_j^k = \frac{\beta_j^k}{\sum_k \beta_j^k}. \quad (1)$$

Shrinkage Smoothing reduces the estimation risk of the probability $p(f_i|c_j)$. Furthermore, to ensure that the MLE along a given path are independent, we subtract child's data from its parent's before calculating the parent's ML estimate. That is, the parent estimate is based on the data that belongs to all siblings of the child, but not the child itself.

3. Experiment and results

3.1. Experiment setting

The basic Naive Bayes model is trained on a sample of sold items on eBay site for one month (15 April 2010–11 May 2010). We assume that items which were successfully sold have been placed in right categories by sellers. So we directly use the sold items with their existing tags of categories as ground truth. Although the assumption is not true for all the cases, it greatly releases the cost of human labeling and serves as a good approximation. Furthermore, the model is evaluated on the sold items in the day following the training period, i.e., 12 May 2010.

In data preprocessing step, titles are first tokenized, and then numbers, punctuations and stop words are removed. Our training set had 1.2 million different valid words. We regard the unigram of words within titles as features and conduct feature selection based on document frequency measure (*DF*) as described in [9]. The *DF* threshold is set to 100 since it is the optimal setting in our preliminary experiments. Finally, there were about 24,000 features in the model.

3.2. Data specification

The eBay category structure we studied is a 7-level-deep topic hierarchy, where there are 34 top level nodes, called meta categories, and about 19,000 bottom level nodes, called leaf categories. Our task is to categorize items into leaf nodes. The training set consists of 18 million items in 18,755 categories while the testing set consists of 278,000 items. The distribution of items on leaf categories is shown in Fig. 1. It is evident from the figure to see the high skewedness and the heavy-tail nature of the item distributions over categories. There are 86.9% categories (the head

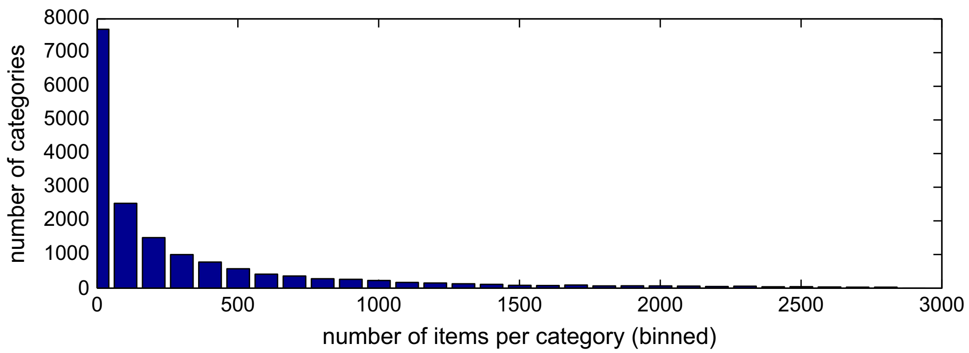


Fig. 1. Distribution of items on categories.

portion) only containing less than 1k items per category, while the tail-portion 1% categories with more than 10k items per category account for 51.7% items. As a consequence, a large portion of categories still suffers from data sparseness in spite of a huge training set on the whole.

3.3. *Effectiveness of smoothing methods*

We evaluate and compare the behavior of individual smoothing methods on the test item collection. For each smoothing method, we investigate the sensitivity of its performance on its smoothing parameter and optimize the performance by experimenting with a wide range of parameter values. Then we compare various smoothing methods on their best run. The precision is reported in terms of 1 (P@1), 5 (P@5), 10 (P@10) and 15 (P@15) categories returned respectively. For each seller's listing, eBay currently recommends 15 categories for his/her consideration.

Table 1 shows the overall precisions of the model without smoothing (*No Smoothing*) and with different smoothing methods respectively. It appears that all smoothing methods perform better than *No Smoothing* on the whole data collection. Even the simplest smoothing method, *Laplace Smoothing* is better than *No Smoothing* with all four precision measures, specially significantly better with P@5, P@10 and P@15 measures. It indicates that smoothing is definitely helpful for our task.

Furthermore, there seems to be a clear order among the five smoothing methods in terms of all four precision measures: *Jelinek–Mercer*, *Dirichlet Prior* and *Absolute Discounting* are significantly better than *Shrinkage Smoothing* by about 3% precision increase, which is better than *Laplace Smoothing* by 2% precision increase.

The three methods, *Jelinek–Mercer*, *Dirichlet Prior* and *Absolute Discounting* achieve close performance on their best runs (the difference of precision is less than 0.3%). Their performance is relatively insensitive to the choice of the parameter λ, μ and δ respectively. The range of the parameter values make the model deviate from the optimal precision by no more than 3%. Among them, the performance of *Dirichlet Prior* is more sensitive to the choice of smoothing parameter than that of *Jelinek–Mercer* and *Absolute Discounting*.

Table 1
Overall precision of No Smoothing and different smoothing algorithms regarding 1, 5, 10 and 15 categories returned.

Smoothing methods	P@1	P@5	P@10	P@15
No Smoothing	52.2	69.6	71.6	72.2
Laplace Smoothing	52.3	76.1	82.1	84.8
Jelinek–Mercer				
$\lambda = 0.1$	55.5	80.1	85.6	88.0
$\lambda = 0.2$	**57.0**	**82.1**	**87.3**	**89.3**
$\lambda = 0.4$	56.8	81.8	87.1	89.2
$\lambda = 0.6$	56.4	81.4	86.8	88.9
$\lambda = 0.8$	55.6	80.4	86.0	88.3
Dirichlet Priors				
$\mu = 100$	54.1	77.9	84.0	86.9
$\mu = 500$	57.1	82.0	87.3	89.4
$\mu = 1000$	**57.3**	**82.2**	**87.5**	**89.7**
$\mu = 2000$	57.3	82.1	87.5	89.7
$\mu = 5000$	56.6	81.4	87.0	89.4
Absolute Discounting				
$\delta = 0.1$	55.0	79.4	85.2	87.7
$\delta = 0.2$	56.8	81.9	87.2	89.2
$\delta = 0.4$	57.0	82.2	87.4	89.5
$\delta = 0.6$	**57.1**	**82.3**	**87.6**	**89.6**
$\delta = 0.8$	57.2	82.4	87.6	89.7
Shrinkage Smoothing	54.2	76.9	83.4	86.5

Since *Dirichlet Prior* achieves the best performance comparing with all the others, we will focus on this smoothing method alone and investigate its detailed behavior on size of training set, size of categories and word specificity.

3.4. *Influence of size of training set on smoothing*

This section investigates how sensitive the performance of system is on training data size. We separated the original one-month training set into four weeks in chronological order. Then we train four models using one week (1-*week*), the first two weeks (2-*weeks*), the first three weeks (3-*weeks*) and all of the data (4-*weeks*) respectively. Finally, the four models are evaluated on the same test set.

As shown in Table 2, with No Smoothing we need more data (1–4 weeks takes us from 41% to 52.2%) but with smoothing we can achieve the same numbers with just 2 weeks of data. Additional weeks marginally improve precision. Furthermore, we observe that the *Dirichlet Prior* smoothing performs better than *No Smoothing* by 11.5% precision on 1-*week* data while it is better by 5.1% on 4-*weeks* data. It is clear that the contribution of smoothing to the system turns slighter with the increase of training size, but at least smoothing never hurts the system in our task. In addition, the optimal values of the prior sample μ ($\mu = 1000$) are consistent with the increase of training data.

3.5. *Influence of category size on smoothing*

There are two purposes of smoothing in maximum likelihood models: one is to address insufficient sample problem and eliminate the zero probability of unobserved words; And the second is to better explain common or non-informative words and decrease the discrimination power of such words when estimating parameters. Therefore, we feel intuitively that the performance of smoothing will be sensitive to the characteristics of category, including whether the category has sufficient training data and whether it has distinguishable features. This section is to understand how the smoothing is correlated with the size of category.

In addition to evaluating smoothing on the whole test set, we further evaluate its performance on two subsets (*LargeCat* and *SmallCat*). The *LargeCat* is the set in which each category contains

Table 2
Precision of Dirichlet Prior Smoothing on different size of training set.

Smoothing methods	1-Week	2-Weeks	3-Weeks	4-Weeks
No Smoothing	41.0	45.6	48.1	52.2
Dirichlet prior				
$\mu = 100$	51.4	53.0	53.9	54.1
$\mu = 500$	52.4	53.9	54.7	57.1
$\mu = 1000$	52.5	54.1	54.9	57.3
$\mu = 2000$	52.2	53.9	54.9	57.3
$\mu = 5000$	50.8	53.0	54.2	56.6

Table 3
Precision of Dirichlet Prior Smoothing on categories with different size.

Smoothing methods	LargeCat	SmallCat
No Smoothing	69.4	35.9
Dirichlet Priors		
$\mu = 100$	69.8	39.1
$\mu = 500$	69.4	45.1
$\mu = 1000$	70.1	43.7
$\mu = 2000$	71.0	41.0
$\mu = 5000$	72.8	35.1

more than 10k training instances while the *SmallCat* is the set in which each category has less than 1k training instances.

The results are shown in Table 3. It is clear that the system on *LargeCat* significantly outperforms that on *SmallCat* by 27.7%, which is consistent with the statement that data sparseness may harm a system seriously. As an effective way to cope with the problem, smoothing saves the system on both *SmallCat* and *LargeCat*, which significantly increases precision by 9.2% and 3.4% respectively on *No Smoothing*. It is also evident that smoothing contributes more to *SmallCat* than to *LargeCat*.

Furthermore, precision is more sensitive to the prior sample size μ for *SmallCat* than for *LargeCat*. The system achieves the optimal performance (45.1%) given a relatively small value of $\mu(\mu=500)$ on *SmallCat* while the optimal value of $\mu(\mu=5000)$ on *LargeCat* tends to be larger. The optimal parameter μ seems to vary from different category sets even in one task. Intuitively, we felt that small categories would require more aggressive smoothing to achieve optimal performance. However, the difference in the optimal μ values suggests that *LargeCat* needs more sufficient smoothing than *SmallCat* which is somehow unexpected. It might be because that there are other factors strongly influencing smoothing. In next section, we will further study the influence of word specificity on smoothing.

3.6. *Influence of word specificity on smoothing*

This section studies how performance is affected by the word specificity of categories. If the words in category are not distinguishable enough, the category should be hard to predict by nature. Since smoothing has been stated as one of the most effective way to explain common/non-informative words [5], this section designs an experiment to test whether the Dirichlet Prior smoothing can effectively cope with the common word problem in our task.

The specificity of a word (w) in given data set is measured with the Inversed Document Frequency (IDF), as $\mathrm{IDF}_w = \log N/n_w$, where, N is the total number of categories and n_w is the number of categories containing the word w. Furthermore, the specificity of category is measured by the average of IDFs of all words in the category. A larger average IDF value of category indicates the words in the category are more specific and distinguishable since they seldom occur in other categories. Therefore, the category with larger IDF value should be easier to classify in principle. We sort categories on their average IDF values in descending order. Table 4 shows some examples of the most and least specific categories on eBay site. Furthermore, two subsets are built based on the ordered list: a set containing categories with high word specificity (*SpecCat*) and a set containing categories with low word specificity (*NotSpecCat*). The *SpecCat* set takes the top 10% categories in the ordered list while the *NotSpecCat* takes the bottom 10% categories. Dirichlet Prior smoothing will be evaluated on *SpecCat* and *NotSpecCat* respectively.

Table 4
Example of categories with the most words specificity and the least words specificity.

Categories with the most word specificity	
6.02	Crafts → Home Arts & Crafts → Floral Crafts → Dried Flowers & Plants → Poppy Pods
5.86	Sports Mem, Cards & Fan Shop → Cards → Etopps → Etopps Hockey
5.38	Toys & Hobbies → Trading Card Games → Vampire, The Eternal Struggle
Categories with the least word specificity	
1.35	Collectibles → Holiday & Seasonal → Christmas → Current (1991–now) → Ornaments → Sleighs
1.42	Entertainment Memorabilia → Music Memorabilia → Rock & Pop → Artists D → Decemberists
1.45	Everything Else → Adult Only → Clothing, Shoes & Accessories → Costumes and Fantasy Wear → Unisex

As shown in Table 5, the system on *SpecCat* performs much better than that on *NotSpecCat* by 31.4% with the best smoothing configuration ($\mu=5000$), which indicates that common word is the key problem to damage a system severely. Although smoothing contributes to the system on both *SpecCat* and *NotSpecCat*, it appears that the system on *SpecCat* (+5.0% on *No Smoothing*) benefits more from smoothing than that on *NotSpecCat* (+2.2% on *No Smoothing*). Furthermore, precision is more sensitive to the prior sample size μ for *SpecCat* than for *NotSpecCat*. The optimal performance is achieved at the same value of μ ($\mu=5000$) on both sets. The large optimal value of μ suggests that the system on both sets needs sufficient smoothing.

Table 5
Precision of Dirichlet Prior Smoothing on categories with different word specificity.

Smoothing methods	SpecCat	NotSpecCat
No Smoothing	71.4	42.8
Dirichlet Priors		
$\mu=100$	73.0	43.7
$\mu=500$	74.7	44.3
$\mu=1000$	75.1	44.7
$\mu=2000$	75.6	44.5
$\mu=5000$	76.4	45.0

4. Two stage classification

In previous section, we considered the item categorization task as a flat classification problem. However, eBay actually has a hierarchical catalog structure. As described in Section 3.2, there are 34 nodes, called meta categories, on the top level and 19,000 nodes, called leaf categories, on the bottom level. Table 6 shows the distribution of leaf categories and items on meta categories.

In this section, we study whether we could benefit from such domain knowledge. We present a two-stage classification approach by leveraging the category hierarchical structure [10–12]. In the first stage, we pre-classify an item t into one of the meta categories c_{meta} and further narrow down to one of the leaf categories c_{leaf} which belongs to c_{meta} in the second stage. Based on Bayesian learning framework, the task of items categorization is now formalized as a joint conditional probability of meta category classification model $P(c_{meta}|t)$ and leaf category classification model $P(c_{leaf}|t,c_{meta})$

$$c^*_{leaf} = \arg \max_{c_{leaf} \in c_{meta}} P(c_{meta}|t)P(c_{leaf}|t,c_{meta}).$$

Previous experiments showed that the performance is sensitive to the choice of smoothing parameter in *Dirichlet Prior* specially on the small category *SmallCat* and the word specific category *SpecCat*. With the two-stage learning, we can do more fine-grained feature selection and optimalize the parameters of classification models on each stage respectively. Table 7 shows the performance of the two-stage learning as compared to the previous best run of the conventional classifications. It appears

Table 6
Distribution of leaf categories, items on meta categories.

Meta category	Leaf category		Item	
Collectibles	4526	24.13%	936,968	5.14%
Business & Industrial	1418	7.56%	345,701	1.90%
Jewelry & Watches	1271	6.78%	1,647,556	9.04%
Toys & Hobbies	1117	5.96%	1,142,041	6.26%
Sporting Goods	1056	5.63%	847,471	4.65%
Home & Garden	1006	5.36%	814,743	4.47%
Everything Else	907	4.84%	136,312	0.75%
Clothing, Shoes & Accessories	844	4.50%	3,410,061	18.71%
Entertainment Memorabilia	828	4.41%	63,563	0.35%
Computers & Networking	717	3.82%	751,387	4.12%
Antiques	621	3.31%	130,640	0.72%
Electronics	541	2.88%	627,507	3.44%
Crafts	525	2.80%	580,886	3.19%
Coins & Paper Money	510	2.72%	552,275	3.03%
Pottery & Glass	472	2.52%	127,784	0.70%
Dolls & Bears	407	2.17%	145,077	0.80%
Health & Beauty	341	1.82%	564,879	3.10%
Stamps	332	1.77%	146,303	0.80%
Musical Instruments	308	1.64%	220,410	1.21%
Cameras & Photo	275	1.47%	308,796	1.69%
Sports Mem, Cards & Fan Shop	206	1.10%	1,082,971	5.94%
Pet Supplies	152	0.81%	92,969	0.51%
Baby	139	0.74%	80,402	0.44%
Specialty Services	64	0.34%	6324	0.03%
Books	36	0.19%	706,476	3.88%
Art	26	0.14%	95,706	0.52%
Travel	25	0.13%	12,818	0.07%
Cell Phones & PDAs	24	0.13%	802,424	4.40%
DVDs & Movies	22	0.12%	551,968	3.03%
Music	14	0.07%	614,139	3.37%
Video Games	12	0.06%	552,639	3.03%
Real Estate	6	0.03%	2673	0.01%
Gift Cards & Coupons	4	0.02%	64,366	0.35%
Tickets	3	0.02%	64,324	0.35%

Table 7
Overall precision of two-stage learning regarding 1, 5, 10 and 15 categories returned.

Methods	P@1	P@5	P@10	P@15
Best run of conventional learning	57.3	82.2	87.5	89.7
Two-stage learning	59.7	83.3	88.2	90.2

Table 8
Precision@Top1 on different meta categories.

Meta category	One-stage	Two-stage
Tickets	88.9	97.0
Cell Phones & PDAs	80.7	80.8
Music	77.9	79.4
Gift Cards & Coupons	75.9	66.2
Sports Mem, Cards & Fan Shop	75.7	77.0
DVDs & Movies	74.7	81.4
Computers & Networking	67.2	68.9
Pottery & Glass	66.2	67.4
Video Games	62.8	63.5
Toys & Hobbies	61.7	62.7
Electronics	60.4	63.7
Sporting Goods	60.1	62.3
Baby	59.8	64.8
Musical Instruments	59.7	60.6
Cameras & Photo	59.6	60.9
Pet Supplies	59.3	60.4
Clothing, Shoes & Accessories	57.7	58.2
Crafts	57.1	59.0
Home & Garden	56.3	58.8
Health & Beauty	56	59.1
Business & Industrial	53.9	57.1
Coins & Paper Money	53.5	53.3
Stamps	52.6	52.4
Travel	50.4	56.1
Everything Else	50	54.0
Jewelry & Watches	50	50.4
Dolls & Bears	47.1	48.2
Books	46.5	50.4
Antiques	44.1	51.1
Art	44	48.5
Collectibles	40.7	42.6
Specialty Services	31.9	37.8
Entertainment Memorabilia	25.7	31.0

that the two-stage learning boosts the performance by 2.4% precision increase on Top1, 1.1% on Top5, 0.7% on Top10 and 0.5% on Top15. Table 8 further lists the detailed performance comparison between the conventional learning and the two-stage learning on individual meta categories. It is evident to see that the performance boosting is more obvious on the meta categories which has less leaf categories, such as *Tickets*, *DVDs & Movies*, *Specialty Services*, *Travel*, *Baby*, *Art* and *Books*. On the contrary, the two stage learning are not so helpful on the categories, such as *Clothing, Shoes & Accessory*, *Jewelry & Watches*, *Stamp* and *Coins & Paper Money*. It might be explained that most of the items in such categories are confused between the sibling of leaf nodes, therefore, they are still hard to classify even if they have been put into the right meta categories successfully.

5. Conclusion

We approached the automatic item categorization problem using a Naive Bayes classification method. The statistical foundation and computational efficiency make this approach particularly suitable for a large-scale item inventory and real-time online response requirement. Specifically, we addressed the data sparseness issue by adapting well-established statistical smoothing methods from language modeling tasks. To investigate important domain-specific characteristics such as short and noisy item title descriptions, large and skewed catalog taxonomy, and vocabulary specialty across different categories, we conducted systematic experiments with large-scale real-world e-commerce item data sets. In general, our experiments showed promising results with smoothing methods. In particular, we provided empirical insights into various important hypotheses to guide a real-world cataloging system design, including: (1) the relationship between the size of the training data set and the impact of smoothing; (2) the need and use of a smoothing method across different categories; and (3) the influence of the specificity of the vocabulary to a category and its impact on the effectiveness of smoothing method. We further study how to effectively take advantage of domain knowledge. We propose a two-stage learning strategy by leveraging domain-specific hierarchical catalog information and optimalizing model setting for each individual stage. We believe our findings with real-world large-scale data will contribute to the practices and applications in similar domains such as web data clustering and classification, online data organization, finding, and recommendation (with social networking systems). For future work, we plan to investigate the adaptation of smoothing methods into other important cataloging tasks such as named-entity extraction for learning new catalog, catalog disambiguation and evolution.

References

[1] G. Salton, Developments in automatic text retrieval, Science 253 (1991) 974–979.

[2] D. Lewis, M. Ringuette, A comparison of two learning algorithms for text categorization, in: Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval, 1994, pp. 81–93.

[3] A. Macallum, K. Nigam, A comparison of event models for naive bayes text classification, in: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, 1998.

[4] C.D. Manning, P. Raghavan, H. Schuetze, Introduction to Information Retrieval, Cambridge University Press, 2008 ISBN: 0521865719.

[5] C.X. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2001, pp. 334–342.

[6] A. McCallum, R. Rosenfeld, T. Mitchell, A.Y. Ng, Improving text classification by shrinkage in a hierarchy of classes, in: Proceedings of ICML-98, 15th International Conference on Machine Learning, 1998.

[7] F. Jelinek, R. Mercer, Interpolated estimation of Markov source parameters from sparse data, Pattern Recognition Pract. (1980) 381–402.

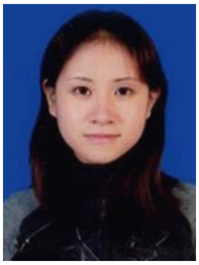
[8] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, R. Stat. Soc. Series B (39) (1977) 1–38.

[9] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: D.H. Fisher (Ed.), Proceedings of ICML-97, 14th International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, USA/Nashville, USA, 1997, pp. 412–420.

[10] S.T. Dumais, H. Chen, Hierarchical classification of web content, in: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, pp. 256–263.

[11] D. Koller, M. Sahami, Hierarchically classifying documents using very few words, in: Proceedings of the 14th International Conference on Machine Learning (ICML), 1997, pp. 171–178.

[12] A.S. Weigend, E.D. Wiener, J.O. Pedersen, Exploiting hierarchy in text categorization, Inf. Retr. (1999) 193–216.



Dan Shen joined eBay in November 2007. She is currently involved in Origami project and focuses on category auto classification. Prior to joining the catalog project, she worked on product review analysis, opinion mining, related item recommendation, collaborative filtering, rare item finding and email classification. Before joining eBay, she studied at Department of Informatics, Edinburgh University for research visiting. She got her PhD degree from Computational Linguistic Department, Saarland University and master degree from School of Computing, National University of Singapore.

Jean-David Ruvini joined eBay Research Lab in 2007. Prior to that he obtained his PhD in computer science (Intelligent User Interfaces) from University of Montpellier, France in 2000, worked on machine learning related projects for 5 years at Bouygues Research Lab (a French conglomerate with Telco, Television, Construction and Water Supply subsidiaries) and joined Shopping.com Research Lab in 2005 where he contributed to design and improve Shopping.com classification and attribute extraction technologies. Since he joined eBay, he worked on machine learning for fraud detection, seller tagging and catalogs.

Rajyashree Mukherjee joined eBay in April 2006. She is currently working in the Merchandising Applied Research group focusing on Similar Items Recommendation. Prior to that she was involved in many projects as part of the Catalogs and Classification group like Seller Tagging, Community Created Catalogs, Product Autotagging, etc. Rajyashree has a masters degree in computer science from the University of California, Irvine and another masters degree in mathematics and computing from the Indian Institute of Technology, Kharagpur.

Neel Sundaresan is a senior director and head of eBay Research Labs. He has been with eBay since 2005. Prior to joining eBay was a founder and CTO of a startup focused on multi-attribute fuzzy search and network CRM. Prior to this he was the head of the eMerging Internet Technologies group at the IBM Almaden Research Center. There he built the first XML-based Search Engine. He was one of the early leaders in building XML technologies including schema-aware compression algorithms, application component generators and pattern-match systems and compilers. He built the first RDF reference implementation as a W3C standard recommendation. He led research work in other areas like domain specific search engines, multi-modal interfaces and assistive technologies, semantic transcoding, web mining, query systems, and classification for semi-structured data. Prior to this he worked on C++ compiler and runtime systems for massively parallel machines and for shared memory systems and also on retargetable compilers, program translators and generators. He has over 50 research publications and several patents to his credit. He has been a frequent speaker at several national and international technology conferences. He has advised 2 PhD and several Masters dissertations. He has a degree in mathematics and a masters in computer science and engineering from the Indian Institute of Technology, Mumbai India and a PhD in computer science from Indiana University, Bloomington. His dissertation was on Modeling Control and Dynamic Data Parallelism in Object-oriented Languages.