Large-Scale Item Categorization in e-Commerce Using Multiple Recurrent Neural Networks

Jung-Woo Ha*
NAVER LABS
Seongnam, 13561, Korea
jungwoo.ha
@navercorp.com

Hyuna Pyo*
NAVER LABS
Seongnam, 13561, Korea
hyuna.pyo
@navercorp.com

Jeonghee Kim[†]
NAVER LABS
Seongnam, 13561, Korea
jeonghee.kim
@navercorp.com

ABSTRACT

Precise item categorization is a key issue in e-commerce domains. However, it still remains a challenging problem due to data size, category skewness, and noisy metadata. Here, we demonstrate a successful report on a deep learning-based item categorization method, i.e., deep categorization network (DeepCN), in an e-commerce website. DeepCN is an end-to-end model using multiple recurrent neural networks (RNNs) dedicated to metadata attributes for generating features from text metadata and fully connected layers for classifying item categories from the generated features. The categorization errors are propagated back through the fully connected layers to the RNNs for weight update in the learning process. This deep learning-based approach allows diverse attributes to be integrated into a common representation, thus overcoming sparsity and scalability problems. We evaluate DeepCN on large-scale real-world data including more than 94 million items with approximately 4,100 leaf categories from a Korean e-commerce website. Experiment results show our method improves the categorization accuracy compared to the model using single RNN as well as a standard classification model using unigram-based bag-of-words. Furthermore, we investigate how much the model parameters and the used attributes influence categorization performances.

CCS Concepts

• Computing methodologies → Neural networks • Computing methodologies → Supervised learning by classification • Information systems → Online shopping.

.Keywords

large-scale item categorization; recurrent neural networks; deep learning; e-commerce

1. INTRODUCTION

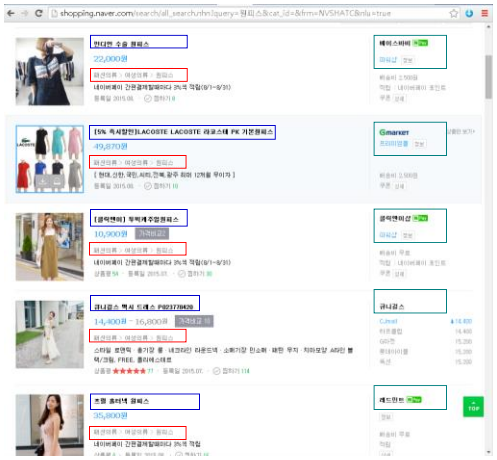
Recent advances in web and mobile technologies have dramatically increased the e-commerce markets. Many new items are registered in e-commerce websites such as *eBay*, *Amazon*, and *NAVER shopping* every day. Each item is represented by metadata such as its title, category, image, price and so on, and most of them are given manually by human sellers. Dissimilar to attributes including title and price that can only be provided by humans, the item categories can be automatically inferred from the metadata. While automatic item categorization can reduce time and economic costs, the accuracy of the categorization has large influences on customer satisfaction and revenue of e-commerce sites [19, 20]. Therefore, precisely categorizing items emerges as a significant issue in e-commerce domains.

Item categorization is a text classification problem because most metadata of items are represented as textual features. Although many models using support vector machines (SVMs) [14], naïve Bayes classifiers [1], decision trees [6], and latent Dirichlet allocations (LDAs) [3, 18] have been successfully applied to text classification, they are difficult to be applied to item categorization due to scalability, sparsity and skewness. Item categorization is mapping several millions of items to class labels, corresponding to several thousands of leaf categories. Moreover, the distribution of leaf categories is usually severely skewed. Furthermore, many new items continue to be listed in an online commerce site on-the-fly every day. Although a seller inputs metadata on items, the data often includes noisy information. Therefore, precise item categorization still remains a challenging problem in online marketplace domains. Shen et al. proposed a hierarchical item categorization method for large items [19-21]. However, it may be inefficient because it does not use a flat-classifier but conducts two levels of classification. Since the method uses a unigram-based approach for learning from text descriptions, in addition, it is difficult to avoid the sparsity problem and to capture the full meaning of given word sequences [2]. Taxonomy-based methods have also showed competitive performances in category integration [17]. However, they require taxonomy of item categories as prior knowledge. Neural network-based approaches also have been successfully applied to text classification such as sentiment analysis [4, 12]. However, the studies on deep learning-based large-scale item categorization over several tens of million goods are still rare to the best of our belief.

Here we propose a flat categorization model for large-scale items from their metadata using deep learning [13] without prior knowledge, called deep categorization network (DeepCN). The proposed DeepCN uses an end-to-end deep network consisting of multiple recurrent neural networks (RNNs) for feature construction [7, 16] and fully connected layers [22] for item categorization. Each RNN is dedicated to each input metadata attribute composing the item information. For example, the RNNs of DeepCN includes Name-RNN, ShoppingMAll-RNN, and so on.

KDD'16, August 13-17, 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-4232-2/16/08…$15.00 .
DOI: http://dx.doi.org/10.1145/2939672.2939678

[*]: Both authors have equally contributed to this work.
[†]: Corresponding author



shopping.naver.com/search/all_search.nhn?query=원피스&cat_id=&frm=NVSHATC&nlu=true

언디안 수술 원피스
22,000원
패션의류 > 여성의류 > 원피스
네이버페이 간편결제할때마다 3%씩 적립(8/1~8/31)
등록일 2015.08. · 찜하기 8
베이스비비 · 파워샵 정보 · 배송비 2,500원 · 적립 네이버페이 포인트 · 쿠폰 상세

[5% 즉시할인]LACOSTE LACOSTE 라코스테 PK 기본원피스
49,870원
패션의류 > 여성의류 > 원피스
[현대,신한,국민,시티,전북,광주 최대 12개월 무이자]
등록일 2015.08. · 찜하기 10
Gmarket · 프리미엄몰 정보 · 배송비 2,500원 · 쿠폰 상세

[클릭앤미] 투픽캐주얼원피스
10,900원 가격비교
패션의류 > 여성의류 > 원피스
네이버페이 간편결제할때마다 3%씩 적립(8/1~8/31)
상품평 54 · 등록일 2015.07. · 찜하기 30
클릭앤미샵 · 파워샵 정보 · 배송비 무료 · 적립 네이버페이 포인트 · 쿠폰 상세

큐니걸스 맥시 드레스 P023778420
14,400원 ~ 16,800원 가격비교
패션의류 > 여성의류 > 원피스
스타일 로맨틱 · 총기장 롱 · 네크라인 라운드넥 · 소매기장 민소매 · 패턴 무지 · 치마모양 A라인 블랙/크림, FREE, 폴리에스터
상품평 ★★★★★ 77 · 등록일 2015.07. · 찜하기 114
큐니걸스 · CJmall 14,400 · 15,200

프릴 홀터넥 원피스
35,800원
패션의류 > 여성의류 > 원피스
네이버페이 간편결제할때마다 3%씩 적립
레드민트 · 배송비 무료 · 적립

Figure 1. Examples of items listed in "NAVER Shopping" website. Red, blue, and green boxes denote the categories, the names, and the shopping malls of items

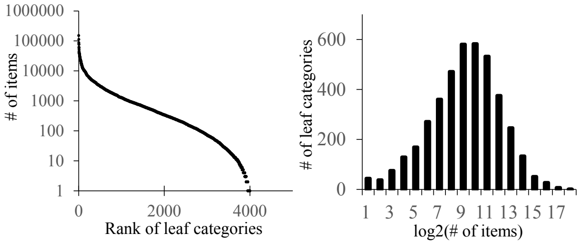
Dedicating an RNN to each attribute prevents the ambiguity emerging by concatenating the attribute word sequences. Also, the dedication approach keeps the word sequence from being too long. The RNNs generate real-value vectors characterizing the description semantics from given metadata through word embedding. The use of RNNs allows the input node size of the fully connected layers to be fixed regardless of the word sequence length. Also, it enables the model to learn the full meaning of text descriptions while bag of words (BoW) methods using *n*-grams use very restricted sequential information. The features generated by the RNNs move through the multiple fully connected layers and a softmax layer into the output layer correspond to leaf categories. The cost function is defined as the errors of the output layer which are propagated through the fully connected layers to the RNNs to update the weights of the model. This end-to-end structure not only allows the word vectors to characterize the category-sensitive semantics but also renders a pretraining process such as word2vec [15] unnecessary.

We evaluate the proposed DeepCN on large-scale industrial data including more than 94 million items dealt with in a Korean famous online marketplace, "*NAVER Shopping*[1]". The number of leaf categories is approximately 4,100. Experimental results show that our method outperforms a neural categorization model using single RNN as well as the Bayesian networks using bag of words based on unigram. Furthermore, we investigate the effects of each metadata attribute on categorization as well as model parameters for high-level categories.

2. ITEM DATA IN E-COMMERCE

2.1 Item Categorization in Online Marketplaces

Item categorization is defined as a problem to classify the leaf category of items from their metadata. In general, item categories are hierarchical and for convenience we separate them into leaf category and high-level category. In this study, the target classified is the leaf category of an item.



(a) Item-Rank (b) Category-Item

Figure 2. Distributions of the item data used in this study. The data are very large-scale over 94 million items and severely skewed with respect to leaf category.

In most online marketplace sites, sellers register an item by manually feeding metadata as shown in Figure 1. However, these metadata are likely to include lots of intentional or unintentional noisy information. For example, the name of socks is "*Socks, stockings, leggings, ankle socks. Thank you so much. We will do our best*!!" There exist many cases where the leaf category of an item is not related to a high-level category given by a seller and a high-level category registered by a seller is not consistent to the category terms of the online shopping system. Simple unigram-based approaches are not sufficient to precisely categorize items from these metadata.

Categorizing items belongs to text classification because metadata are in general represented by textual description. However, it is more challenging, compared to traditional text classification problems. In e-commerce domains, the data distribution has a long tail, which means that there are many leaf categories including a few items only. The data used in this study also show a long tail distribution according to Figure 2. It is well known that the leaf categories in a long-tail position are difficult to be categorized correctly because it is a severely imbalanced data problem. In addition, many new items continue to be registered every day. Furthermore, the number of leaf categories is over several thousand. Therefore, although a model might initially show good performance, its accuracy could decrease as e-commerce sites add new items with time.

2.2 Online Item Metadata

Item metadata include various attributes such as specification, name, hierarchical category information, image, brand, and so on. However, sellers often register data directly related to sale or search only and omit the rest of them. In this study, therefore, we use six essential attributes including item name, brand name, high-level (HL) category given by sellers, sellers' shopping mall id, maker, and image signature. An image signature is represented by nominal value characterizing the color and the edge patterns of an image. This allows image signatures to be used as a symbolic attribute. Therefore, all the attributes are characterized as textual word sequences or nominal values.

Formally, an item d consists of its leaf category label y and an attribute vector $\mathbf{x}$ represented with a collection of six attributes:

$$d = \{\mathbf{x}, y\} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(6)}, y\}, \qquad (1)$$

[1] http://shopping.naver.com

Table 1. Description of item metadata attributes

Var	Attributes	Values	Example
$\mathbf{x}^{(1)}$	Item name	Word sequence	Stylish wallet
$\mathbf{x}^{(2)}$	Brand name	Word sequence	Louis quatorze
$\mathbf{x}^{(3)}$	High-level category	Word sequence	Miscellaneous goods / Women goods
$\mathbf{x}^{(4)}$	Shopping Mall ID	Nominal	A023012
$\mathbf{x}^{(5)}$	Maker	Word sequence	Louis quatorze
$\mathbf{x}^{(6)}$	Image signature	Nominal	38720307
y	Leaf category	Nominal	3423(Women's wallet)

Shopping mall id is users' unique id, which is automatically generated when a user signs in NAVER Shopping.

$$\mathbf{x} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(6)}\}\,. \tag{2}$$

Table 1 explains the metadata description in detail. For preprocessing, we eliminate all the symbols that are not characters or numbers from the metadata. In addition, we treat all nominal values as a textual word. Then, the *i*-th metadata attribute of an item is defined as the sequence of textual words as follows:

$$\mathbf{x}^{(i)} = \{x_1^{(i)} x_2^{(i)} ... x_n^{(i)}\}\,. \tag{3}$$

3. DEEP LEARNING-BASED ITEM CATEGORIZATION

Deep categorization network (DeepCN) is an end-to-end deep learning model for categorizing large-scale items from their textual and nominal metadata. DeepCN uses multiple RNNs and fully connected layers for learning, thus integrating the processes of metadata attribute-specific feature construction and category classification. The given textual data are transformed into real-valued vectors which precisely distinguish categories by learning.

3.1 Deep Categorization Networks

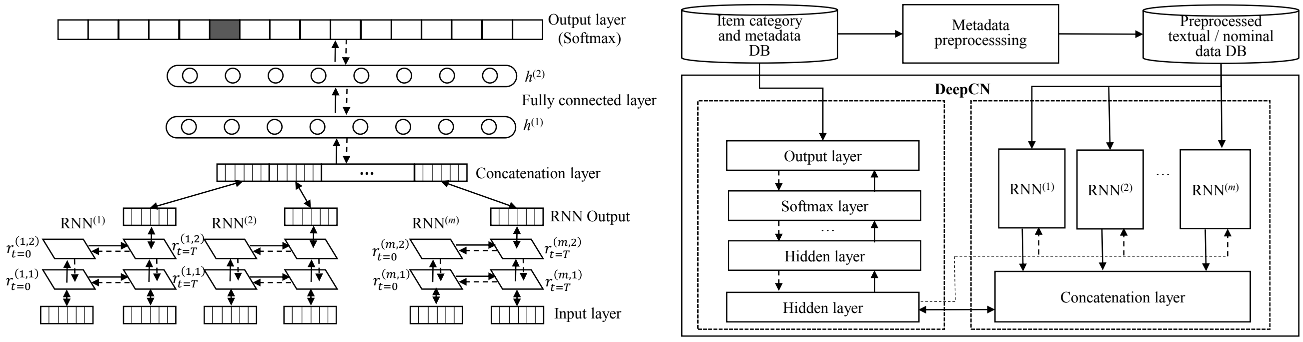
DeepCN consists of multiple RNNs, fully connected layers, one softmax and an output layer. The RNNs generate real-valued feature vectors from given metadata represented by word sequences. The generated vectors characterize the semantics of a word sequence. Each RNN is dedicated to one attribute composing metadata, and thus DeepCN includes *m* RNNs when item metadata consist of *m* attributes. This dedication approach keeps the accuracy from being low by too long word sequence length and the ambiguity emerging by concatenating attribute words. All the output vectors generated from the RNNs are concatenated into one vector, which moves to the fully connected layer. The nodes in the output layer correspond to leaf categories and each output node value is represented the probability of the corresponding leaf category where the given item belongs through fully connected and softmax layers. Figure 3(a) presents the structure of a DeepCN model.

DeepCN is formulated with traditional terms of neural networks. Let ${}^{R}_{m}\mathbf{h}_t^{(n)}$ and ${}^{F}\mathbf{h}^{(a)}$ denote the *n*-th layer at time *t* of the *m*-th RNN and the *a*-th fully connected layer. *R* and *F* mean RNN and fully connected, respectively. Like the preceding, ${}^{R}_{m}W^{kn}$ and ${}^{F}W^{kn}$ denote the weight matrix between the *k*-th and the *n*-th layers of the *m*-th RNN and the fully connected layers, respectively. ${}^{R}_{m}\mathbf{h}_t^{(n)}$ is defined with a function whose parameters are the hidden layer vector at *t*-1, ${}^{R}_{m}\mathbf{h}_{t-1}^{(n)}$, and the hidden layer vector of (*n*-1)-th layer at time *t*, ${}^{R}_{m}\mathbf{h}_t^{(n-1)}$:

$${}^{R}_{m}\mathbf{h}_t^{(n)} = {}^{R}_{m}f^{(n)}\left({}^{R}_{m}W^{(n-1)n}\,{}^{R}_{m}\mathbf{h}_t^{(n-1)} + {}^{R}_{m}W^{nn}\,{}^{R}_{m}\mathbf{h}_{t-1}^{(n)} + {}^{R}_{m}\mathbf{b}^{(n)}\right), \tag{4}$$

$${}^{R}_{m}\mathbf{h}_t^{(1)} = {}^{R}_{m}f^{(1)}\left({}^{R}_{m}W^{x1}\mathbf{x} + {}^{R}_{m}W^{11}\,{}^{R}_{m}\mathbf{h}_{t-1}^{(1)} + {}^{R}_{m}\mathbf{b}^{(1)}\right), \tag{5}$$

where ${}^{R}f^{(n)}$ and ${}^{R}\mathbf{b}^{(n)}$ denote the nonlinear activation function and the bias vector of ${}^{R}\mathbf{h}_t^{(n)}$. As defined in (5), the first layer uses the input layer as the (*n*-1)-th layer and **x** denotes the input vector with represented with one hot encoding, which means the index of



(a) Structure of DeepCN

(b) Overall learning flow of DeepCN

Figure 3. Structure (a) and overall learning flow (b) of DeepCN. In (a), the model consists of *m* attribute-specific RNNs with two fully connected layers and a softmax layer for the output layer. Solid and dashed lines denote the flows of data and errors during the learning process, respectively.

the *t*-th word in a given metadata sequence is only one and the rests are zero. The size of **x** is equal to the vocabulary size. The output vectors of all the attribute-RNNs are concatenated into one vector **u**, which is given to the fully connected layers:

$$\mathbf{u} = {}^{R}_{1}\mathbf{h}^{(n)}_{T_1} \circ \cdots \circ {}^{R}_{m}\mathbf{h}^{(n)}_{T_m}, \tag{6}$$

where T_i denotes the length of the word sequence of the *i*-th attribute.

The fully connected layer is defined as:

$${}^{F}\mathbf{h}^{(a)} = {}^{F}f^{(a)}\left({}^{F}W^{a(a-1)}\,{}^{F}\mathbf{h}^{(a-1)} + {}^{F}\mathbf{b}^{(a)}\right), \tag{7}$$

$${}^{F}\mathbf{h}^{(1)} = {}^{F}f^{(1)}\left({}^{F}W^{21}\mathbf{u} + {}^{F}\mathbf{b}^{(1)}\right). \tag{8}$$

The hyperbolic tangent function is used for the activation functions of both the RNNs and the fully connected layers in this study since it in general provides better performance in RNN learning than the sigmoid function [10].

For representing the probability for each category, we use the softmax function in the output node of DeepCN:

$$P(y_k \mid {}^{F}\mathbf{h}^{(l)}) = \frac{\exp\left(\mathbf{w}_k\,{}^{F}\mathbf{h}^{(l)}\right)}{\sum_{y\in Y}\exp\left(\mathbf{w}_y\,{}^{F}\mathbf{h}^{(l)}\right)}, \tag{9}$$

where $\mathbf{w}_k$ means the weight vector between the *k*-th output node and the *l*-th fully connected layer nodes, and *l* denotes the number of fully connected layers.

Therefore, given an item metadata represented with textual word sequences, DeepCN calculates the category probability vector of the item, and classifies the category with the maximum probability as the item category.

3.2 DeepCN Learning

The objective function is defined as the categorization errors on the given item metadata dataset, which is minimized by the training process. Figure 3(b) illustrates the learning flow of DeepCN.

Formally, the categorization error is formulated as follows:

$$E = \frac{1}{2}\sum_{n=1}^{N}\left\{y^{(n)} - \hat{y}^{(n)}\right\}^2 \equiv \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\right\|^2, \tag{10}$$

where $\mathbf{y}^{(n)}$ and $\hat{\mathbf{y}}^{(n)}$ denote the one-hot-encoding vector of the real category of the *n*-th item and the calculated softmax probability vector, respectively. The errors are propagated from the output layer into the fully connected hidden layers and the weights of the hidden layers are updated as follows:

$$\mathbf{w}_i = \mathbf{w}_i - \eta\frac{\partial E}{\partial \mathbf{w}_i} \text{ and } \frac{\partial E}{\partial \mathbf{w}_i} = \delta_i\mathbf{x}, \tag{11}$$

$$\delta_i = \begin{cases} \left(y_i^{(n)} - \hat{y}_i^{(n)}\right)\left(1 - \tanh^2(net_i)\right), \text{ if } i \in \mathbf{o} \\ \left(\sum_{j\in J}\delta_j w_{ij}\right)\left(1 - \tanh^2(net_i)\right), \text{ if } i \in \mathbf{h} \end{cases}, \tag{12}$$

where **o** and **h** denote the node set of an output and an inner hidden layer. η is the learning rate and net_{i} denotes the net value of the *i*-th node, the weighted sum of the lower layer node values. The backpropagated error gradients are separated into the top hidden node errors of each attribute-RNN. After that, all the weights of the RNNs are updated by backpropagation through time (BPTT) [23]:

$${}^{R}_{m}\mathbf{w}^{kk}_{i} = {}^{R}_{m}\mathbf{w}^{kk}_{i} - \eta\sum_{t=1}^{T}\delta_i(t)\,{}^{R}_{m}\mathbf{h}^{(k)}_{t-1}, \tag{13}$$

$$\mathbf{w}^{(n-1)n}_{i} = \mathbf{w}^{(n-1)n}_{i} - \eta\sum_{t=1}^{T}\delta_i(t)\,{}^{R}\mathbf{h}^{(n-1)}_{t}. \tag{14}$$

Therefore, this error propagation allows the word vector representation to capture the information on the category as well as the relationships between words.

Figure 4 shows the algorithm of learning DeepCN. The word sequences of item metadata are given to each attribute-RNN. The

Algorithm 1: Learning of DeepCN

D' / D^{te}: Minibatch trainng dataset / test dataset
M: The number of RNNs (= # of used metadata attributes)
U: The set of concatenated vectors
E: Categorization error
${}^{R}E_m$: Backpropagated error of the *m*-th RNN
V_m: Word sequence embedding vector set by the *m*-th RNN
S: The number of separated minibatch datasets
${}^{F}\theta^{i}$ / ${}^{R}\theta^{i}_{m}$: Model parameters of FC layers and the *m*-th RNN at the *i*-th iteration
MAXITER: The number of maximum iteration for learning

```
( ${}^{R}\theta^{0}_{1}$ ,…, ${}^{R}\theta^{0}_{M}$ , ${}^{F}\theta^{0}$ ) ← Initialize($M$)
for $i$ =1 to $MAXITER$
  for $j$ =1 to $S$
    $D'$ ← GetMiniBatch($D$, |$D'$|);
    for $m$=1 to $M$
        $V_m$ ← DeepCN_RNN_Forward($D'$, ${}^{R}\theta^{i-1}_{m}$ );
    endfor
    U ← Concatenate($V_1$,…,$V_M$)
    $E$ ← DeepCN_FC_Softmax_Forward($U$, ${}^{F}\theta^{i-1}$ );
    ( ${}^{F}\theta^{i}$ , ${}^{R}E$ ) ← DeepCN_ FC_Softmax_Backward($E$, ${}^{F}\theta^{i-1}$ );
    for $m$=1 to $M$
      ${}^{R}E_m$ ← SeparateError( ${}^{R}E$ , $m$)
      ${}^{R}\theta^{i}_{m}$ ← DeepCN_RNN_Backward( ${}^{R}E_m$ , ${}^{R}\theta^{i-1}_{m}$ );
    endfor
  endfor
  EvaluateDeepCN($D^{te}$, ${}^{R}\theta^{0}_{1}$ ,…, ${}^{R}\theta^{0}_{M}$ , ${}^{F}\theta^{0}$ )
endfor
```

Figure 4. Algorithm of learning DeepCN

Table 2. Composition of item data with respect to high level categories

High-level category	# of leaf categories	Training (8)	Validation (2)	Test (1)	Total	Data size / leaf category
Fashion clothing	103	7,201,594	1,800,399	900,200	9,902,193	**96,138**
Miscellaneous goods	255	12,798,623	3,199,656	1,599,829	17,598,108	**69,012**
Cosmetic / Beauty	156	2,595,834	648,959	324,480	3,569,273	22,880
Digital & home electronic	642	13,341,844	3,335,462	1,667,731	18,345,037	28,575
Furniture / Interior	335	5,159,216	1,289,805	644,903	7,093,924	21,176
Childbirth / Infant care	473	5,162,732	1,290,684	645,342	7,098,758	15,008
Foods	432	1,610,087	402,522	201,261	2,213,870	**5,125**
Sports / Leisure	459	5,255,248	1,313,813	656,907	7,225,968	15,743
Life / Health	1,115	15,104,282	3,776,071	1,888,036	20,768,389	18,626
Trip / Culture	61	748,506	187,127	93,564	1,029,197	16,872
Tax free goods	85	25,196	6,299	3,150	34,645	**408**
Total	4,116	69,003,162	17,250,797	8,625,403	94,879,362	23,051

Bold faces denote high-level categories with a very skewed data size per leaf category. Values in parenthesis of table header mean the ratios of data separated for learning deepCN.

real-valued vectors characterizing the word sequences are generated by the forward mechanism of an RNN. The generated vectors are concatenated into one vector for each item, which is forwarded into the fully connected hidden layers and then is transformed into the probability vector of the leaf categories with the softmax layer. After that, the categorization error is propagated through the fully connected layers into the RNNs and the weights are updated by the BPTT method.

4. EXPERIMENTAL RESULTS

4.1 Data and Parameter Setup

We evaluated the proposed DeepCN on a very large-scale dataset including over 94.8 million items with 4,116 leaf and 11 high-level categories. These items are currently registered in a Korean famous online e-commerce website, "*NAVER Shopping*." The high-level categories of the items are manually given by sellers. The item leaf categories are manually classified by many human experts employed at the website and these were used as the ground truth category for learning and evaluating the models.

The data are randomly separated into training, validation, and test sets, whose ratios are 8/11, 2/11, and 1/11, respectively. Table 2 shows the composition of the data in detail. As shown in Table 2 and Figure 2, we find that the high-level categories are severely imbalanced in both the number of leaf categories and the number of items per leaf category. As preprocessing, we eliminated rare words and sentence symbols including parenthesis, quotation, and period from the metadata. The resulting vocabulary size is 2,836,443.

The fixed model parameters for experiments are learning rate, momentum, and minibatch size, which are 0.001, 0.9 and 100, respectively. These values are selected by many experimental experiences and we focus on main parameters such as the number of hidden layers, hidden node size, and the used metadata attributes in this study. DeepCN was implemented based on CUDA 6.4 using a single GPU, Titan Black with 6GB VRAM.

4.2 Performance Measure and Comparison

We define a new measure, relative accuracy, instead of using conventional accuracy term due to a legal issue in this study. Relative accuracy of a model θ for given data D $\tilde{\psi}(D;\theta)$ is defined as the ratio of an estimated accuracy $\psi(D;\theta)$ to basis accuracy $\bar{\psi}$:

$$\tilde{\psi}(D;\theta) = \frac{\psi(D;\theta)}{\bar{\psi}} . \qquad (15)$$

The basis accuracy is the accuracy of the model using all metadata attributes (DCN-6R) and can be flexibly defined for target categories.

The global mean accuracy can be computed by averaging accuracies for all the leaf categories of DCN-6R on the test dataset. For example, when an estimated accuracy of a model is 0.8 and the global mean accuracy is 0.9, the relative accuracy is 0.889 (0.8/0.9). The relative accuracy is enough to show the improvements by using the proposed DeepCN even if we cannot use the conventional absolute accuracy.

We compare our method to one deep learning-based method and one conventional bag-of-words (BoW) approach. The first is a model using the architecture same as the proposed method except using only single RNN for item metadata (DCN-1R). In this setting, all the word sequences of the item metadata are concatenated into one long sequence, which is given as an input data instance. Comparing DCN-1R, we validate the effects of the approach dedicating an RNN to each metadata attribute. The second is the Bayesian networks using BoW based on uni-gram as the input feature (BN_BoW). In this setting, all the words appearing in the metadata of an item are converted into a TF/IDF vector, which is given as an input. The maximum number of parent nodes of the Bayesian network is set to 2. Comparing BN_BoW, we confirm the effects of word-embedding by deep neural networks as the data representation.

Table 3: Relative accuracies of three methods

High-level category	DCN-6R	DCN-1R	BN_BoW
Fashion clothing	**1.004**	0.984	0.696
Miscellaneous goods	**0.895**	0.866	0.443
Cosmetic / Beauty	**1.011**	0.976	0.823
Digital & home electronic	**1.108**	1.091	0.852
Furniture / Interior	**0.983**	0.952	0.665
Childbirth / Infant care	**0.956**	0.926	0.740
Food	**1.033**	1.005	0.791
Sports / Leisure	**1.016**	0.985	0.713
Life / Health	**0.992**	0.963	0.628
Trip / Culture	**1.197**	1.193	0.930
Tax free goods	**1.027**	1.008	0.101
Total	**1.000***	0.974	0.676

* denotes the basis accuracy for the relative accuracy.

4.3 Categorization Performance

Table 3 presents the relative accuracies of DeepCN compared to the DeepCN using single RNN and the Bayesian networks using BoW. The values are the categorization accuracies of the leaf categories for each high-level category, and are averaged on 10 times of experiments. The basis accuracy for relative accuracy in Table 3 is the conventional mean accuracy, which denotes the ratio of the number of correct categorization to the size of all test data. DCN-6R has six RNNs with two layers and two fully connected hidden layers. DCN-1R use one RNN and the same number of fully connected layers. The word vector size, the hidden node size of the RNNs is 100 and the fully connected layer node size is 600. As shown in Table 3, DCN-6R shows better categorization accuracy as 2.6% and 32.4% on average compared to DCN-1R and BoW-based Bayesian network. This indicates the categorization accuracies are improved by the dedicated RNN approach as well as word-embedding representation using neural networks.

We investigate the improvement of the proposed method in the level of high-level categories. Table 4 shows the decreases of categorization accuracies of two compared methods with respect to high-level categories. Unlike Table 3, the basis accuracy for each high-level category is defined as the mean accuracy of the high-level category. As shown in Table 4, DCN-1R shows even accuracy decreases from 1% to 3.5% for all high-level categories.

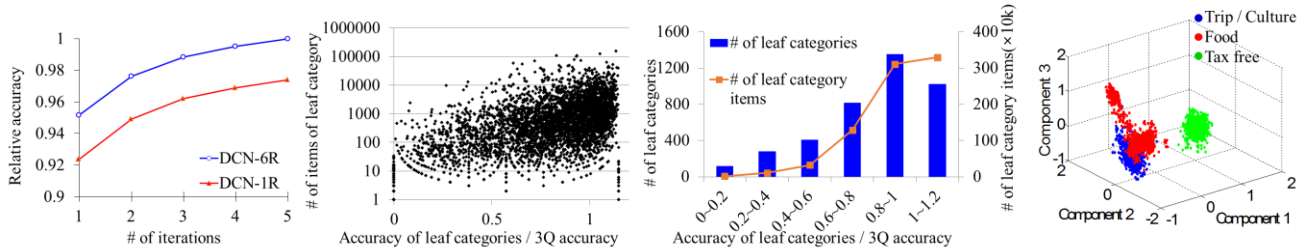
Table 4: Accuracy decrements of two methods for each high-level category

High-level category	DCN-1R	BN_BoW
Fashion clothing	0.979	0.693
Miscellaneous goods	0.968	**0.495**
Cosmetic / Beauty	0.966	0.814
Digital & home electronic	0.985	0.769
Furniture / Interior	0.968	0.676
Childbirth / Infant care	0.968	0.774
Food	0.974	0.766
Sports / Leisure	0.969	0.702
Life / Health	0.971	0.633
Trip / Culture	0.996	0.777
Tax free goods	0.982	**0.099**
Average on HL categories	0.975	0.654

Bold face denotes the large decrease of categorization accuracy.

This indicates the dedicated RNN approach is effective for most categories. Unlike the results of DCN-1R, the accuracy decrements of BN_BoW show large disparity. In particular, large decreases appear in *miscellaneous goods* and *tax free items*. In general, *tax free items* are difficult to be discriminated from other categories because they are same to other category items but are dealt with in tax free shop only. In addition, we can imagine that low accuracy of *tax free goods* may be caused by their smaller number of data compared to other categories as shown in Table 2. *Miscellaneous goods* are usually more general than other categories. From Table 4, word-embedding methods by a deep learning approach can resolve these problems which are difficult for conventional BoW-based models.

Figures 5(a) shows the learning curves of DCN-6R and DCN-1R. Two curves show the similar pattern while the learning proceeds and the accuracy differences between two methods are steadily kept. Figures 5(b) and (c) present the distributions of leaf categories with respect to accuracy. In two figures, x-axis values are relative accuracies or accuracy ranges whose basis accuracy is the third quartile of all leaf category accuracies of DCN-6R. We used this 3Q-normalzied relative accuracy in order to investigate the accuracy patterns for the sizes of leaf categories. 1.0 in x-axis



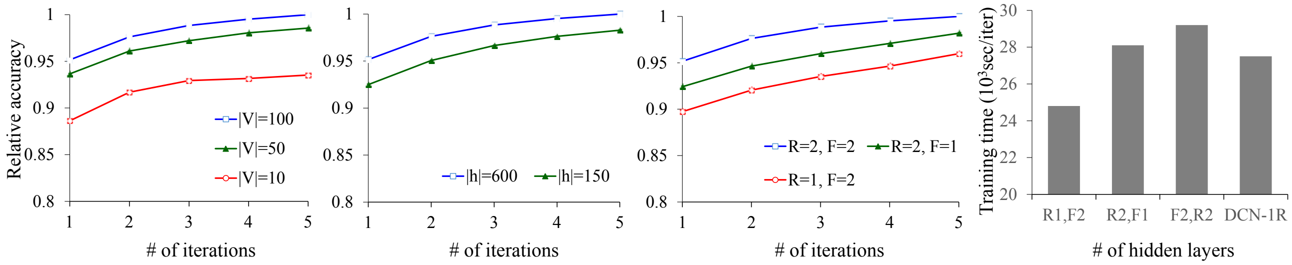
(a) Learning curve

(b) Distribution of item number of leaf category for accuracy

(c) Distribution of the number of leaf categories and leaf category items for accuracy

(d) PCA plot of word-embedding vectors of metadata

Figure 5. Learning curves of DCN-6R and DCN-1R (a), distribution of the numbers of items contained in leaf categories for a normalized categorization accuracy (b), distribution of the numbers of leaf categories (bar) and items contained in leaf categories (line) for a normalized accuracy (c), and three dimensional PCA plot of the concatenated output vectors of the RNNs for three high-level categories (d). The normalized accuracy of a leaf category is the ratio of the accuracy of the leaf category to the third quartile accuracy for all leaf categories of DCN-6R.



(a) Word vector size (b) Number of hidden nodes (c) Number of hidden layers (d) Training time

Figure 6. Relative accuracies on three model parameters such as word vector size (a), hidden node sizes of the fully connected layers (b), numbers of hidden layers (c), and training time for the number of hidden layers (d). |*V*| and |*h*| are the sizes of the word vector and hidden nodes of the fully connected layers. R*n* and F*n* denote the numbers of hidden layers of the RNNs and the fully connected layers of DeepCN, respectively.

denotes the accuracy equal to the third quartile accuracy. As shown in Figure 5(b), DeepCN in general provides better accuracies for leaf categories containing more items over 10,000 goods. It is known that goods in a long tail are usually difficult to be categorized correctly. Interestingly, we find that DeepCN shows high accuracy for many long tail leaf categories including less than 10 items from figure 5(b) despite low accuracy of many other leaf categories belonging to a long tail. Figure 5(c) shows the distributions of the number of leaf categories and leaf category items for normalized accuracy, respectively. From Figures 5(c), we can confirm that the accuracy variances are not large because the proportion below 0.6 compared to the third quartile accuracy is less than 20% with respect to the number of leaf categories. Considering Figure 5(b) and the distribution of the number of leaf category items in Figure 5(c), we can find that DeepCN provides better accuracy for leaf categories including more items.

Figure 5(d) shows the three-dimensional PCA plot of concatenated word-embedding vectors of item metadata extracted from the learned RNNs for three different high-level categories such as *trip/culture*, *food*, and *tax-free goods*. Similar to many RNN-based language models, item metadata are represented by DeepCN as real-valued vectors whose dimension size is equal to the number of hidden nodes of a top RNN layer. These vectors characterize the semantic and syntactic information of the metadata of an item to discriminate its category more precisely. We can find that the metadata vectors are separately scattered in a three-dimensional space for each high-level category. This indicates that DeepCN generates a category-specific metadata representation, thus enhancing the classification accuracies compared to conventional BoW-based methods.

4.4 Effects of Parameters

We investigate how much model parameters influence categorization accuracy and training time. In this study, we focus on three parameters directly related to model architectures such as word-embedding vector size, the number of hidden layers, and hidden node size of the fully connected layers. The word vector size is equivalent to the hidden node size of the RNNs.

Figure 6 presents the effects of the parameters on the categorization performance, including the word vector size (a), the number of hidden nodes in fully connected layers (b), and the number of hidden layers (c). As shown in Figures 6(a) and (b), larger word vector size and hidden node size provide better categorization accuracy. This is consistent to the results of many deep learning-based language models [14]. In particular, too a small embedding-vector size such as |*V*|=10 makes the accuracy worsen because it is not enough for discriminating categories. Figures 6(c) and (d) shows the effects of the hidden layer numbers on the accuracy and learning time, respectively. As shown in Figures 6(c) and (d), more number of hidden layers provides better accuracy. In particular, comparing the results of R2F1 model and DCN-1R to that of R1F2, we indicate that the number of RNN layers and RNNs has larger influence on accuracy and time cost that the number of fully connected layers.

4.5 Effects of Attributes

Table 5 shows the relative accuracy of using subsets of the metadata attributes. The basis accuracy is same as that of Table 4. The results in Table 5 are from a model using two RNN hidden layers and two fully connected layers (R2F2) with |*V*|=100 and |**h**|=600. The values in Table 5 are computed by the same way to the results in Table 4. As shown in Table 5, on average, the model using all the metadata attributes shows higher accuracy as 2.3% compared to that of model excluding image signatures. This result

Table 5: Relative accuracy for used metadata attributes

High-level category	Ex Image	Ex Image+ Sm_id
Fashion clothing	0.983	0.901
Miscellaneous goods	**0.958**	**0.870**
Cosmetic / Beauty	0.981	0.908
Digital & home electronic	0.975	0.929
Furniture / Interior	0.965	0.885
Childbirth / Infant care	0.970	0.882
Food	0.980	0.890
Sports / Leisure	0.977	0.888
Life / Health	0.966	0.884
Trip / Culture	0.997	0.974
Tax free goods	0.994	**0.814**
Total	0.977	0.893

Ex denotes excluded attributes. Bold face denotes the large decrease of categorization accuracy.

indicates that image signatures slightly improve the categorization accuracy for all high-level categories. When shopping mall id is excluded in addition to image signature, the mean relative accuracy considerably decreases from 0.977 to 0.893. Interestingly, we find that the accuracy on *tax-free goods* dramatically decreases while those on other high-level categories slightly worsen. This result indicates shopping mall id is a critical attribute for discriminating tax-free goods. Considering the results in Table 4 together, this result indicates that DeepCN effectively uses shopping mall id attribute for categorization unlike BoW-based methods.

5. CONCLUDING REMARKS

Item categorization is a significant but challenging problem in e-commerce domains. We demonstrated a deep learning-based model for categorizing large-scale items reaching 100 million goods from textual metadata in a real-world online shopping website, i.e., a deep categorization network (DeepCN). The proposed DeepCN is an end-to-end model integrating multiple RNNs dedicated to each metadata attribute for feature construction, two fully connected layers, and one softmax layer for item categorization. The categorization errors are propagated back through the fully connected layers to the RNNs to update the weights. This structure enables the model to categorize items using the semantics of the word sequences from the metadata, thus overcoming simple bag of words-based text classification models using an *n*-gram approach. Also, the proposed method to dedicate each RNN to an attribute composing item metadata not only prohibits the ambiguity emerging by concatenating semantically heterogeneous word sequences describing item attributes and but also keeps the length of word sequences from being too long.

We evaluated DeepCN on large-scale real-world data including 94.9 million items with 4,116 leaf categories from a Korean e-commerce site, *NAVER Shopping*. Experimental results show the proposed DeepCN slightly outperforms the model using single RNN for all metadata attributes and dramatically improves the categorization accuracy compared to conventional BoW-based models using the Bayesian networks. Also, DeepCN can precisely categorize leaf categories in a long tail position, and this means our method can be applied to problems of classifying imbalanced data. Furthermore, we investigated the effects of the key model parameters such as the numbers of hidden layers and hidden nodes. We found that the number of RNN layers is a critical parameter for model performances including categorization accuracy and learning time. In addition, we confirmed the effects of the metadata attributes including image signatures and shopping mall ids on categorization. Furthermore, DeepCN is a general model that can be applied to various text classifications such as sentiment analysis and document classification even if it is applied to large-scale item categorization in this study.

DeepCN has some directions improved despite its encouraging results. Although DeepCN provides relatively good accuracies for more than tens of leaf categories in a long tail, in particular, the performances for more long-tail leaf categories are still not satisfactory. It can be improved by using advanced RNNs such as long short-term memories (LSTMs) [8] and gated recurrent units (GRUs) [5]. The use of the convolutional neural network features [9, 11] for item images would improve the performance instead of image signature. Moreover, the model will be advanced to deal with the problem where new leaf categories continue to increase with time.

6. ACKNOWLEDGMENTS

This work was partly supported by the Creative Industrial Technology Development Program (10053249, Development of Personalized Healthcare System Exploiting User Life-log and Open Government Data for Business Service Model Proof on Whole Life Cycle Care) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

7. REFERENCES

[1] Agrawal, R. and Srikant, R. 2001. On Integrating Catalogs. In *Proceedings of the 10th International Conference on World Wide Web* (*WWW 2001*), 603-612.

[2] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3, 1137-1155.

[3] Blei, D. M., Ng, A. Y. and Jordan, M. I. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3, 993-1022.

[4] Cao, Z., Li, S., Liu, Y., Li, W., and Ji, H. 2015. A Novel Neural Topic Model and Its Supervised Extension. In *Proccedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (*AAAI* 2015), 2210-2216.

[5] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. 2015. Gated Feedback Recurrent Neural Networks. *arXiv preprint arXiv*:1502.02367.

[6] Dalal, M. K. and Zaveri, M. A. 2011. Automatic Text Classification: a Technical Review. *International Journal of Computer Applications*, 28, 2, 37-40.

[7] Graves, A. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks*, 385. Heidelberg: Springer.

[8] Graves, A. and Schmidhuber, J. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18, 5, 602-610.

[9] He, K. Zhang, X., Ren, S., and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR 2016*).

[10] Jozefowicz, R., Zaremba, W., and Sutskever, I. 2015. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning* (*ICML-15*), 2342-2350.

[11] Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 1097-1105.

[12] Lai, S., Xu, L., Liu, K., and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence* (*AAAI 2015*), 2267-2273.

[13] LeCun, Y., Bengio, Y., and Hinton, G. 2015. Deep Learning, *Nature*, 521, 436-444.

[14] Lee, L. H., Wan, C. H., Rajkumar, R., and Isa, D. 2012. An Enhanced Support Vector Machine Classification

Framework by Using Euclidean Distance Function for Text Document Categorization. *Applied Intelligence*, 37, 1, 80-99.

[15] Mikolov, T. Sutskever, I. Chen, K., Corrado, G., and Dean, J. 2013. Distributed Representation of Words and Phrases and Their Compositionality, In *Advances in Neural Information Processing Systems* (*NIPS 2013*), 3111-3119.

[16] Mikolov, T., Kombrink, S., Burget, L., Cernocky, J. H., and Khudanpur, S. 2011. Extensions of Recurrent Neural Network Language Model. In *Proceedings of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP* 2011), 5528-5531.

[17] Papadimitriou, P., Tsaparas, P., Fuxman, A., and Getoor, L. 2013. TACI: Taxonomy-Aware Catalog Integration. *Knowledge and Data Engineering, IEEE Transactions on*, 25, 7, 1643-1655.

[18] Paul, M. and Dredze, M. 2012. Factorial LDA: Sparse Multi-Dimensional Text models. In *Advances in Neural Information Processing Systems* (*NIPS 2012*), 2582-2590.

[19] Shen, D., Ruvini, J.-D., Mukherjee, R., and Sundaresan, N. 2012. A Study of Smoothing Algorithms for Item Categorization on e-Commerce sites. *Neurocomputing*, 92, 54-60.

[20] Shen, D., Ruvini, J.-D., Sarwar, B. 2012. Large-scale Item Categorization for e-Commerce. In *Proceedings of International Conference on Information and Knowledge Management* 2012 (*CIKM*'12), 595-604.

[21] Shen, D., Ruvini, J.-D., Somaiya, M. and Sundaresan, N. 2011. Item Categorization in the e-Commerce Domain. In *Proceedings of International Conference on Information and Knowledge Management* 2011 (*CIKM*'11), 1921-1924.

[22] Suykens, J. A., Vandewalle, J. P., and de Moor, B. L. 2012. *Artificial Neural Networks for Modelling and Control of Non-Linear systems*. Springer Science & Business Media.

[23] Werbos, P. J. 1990. Backpropagation through Time: What It Does And How to Do It. In *Proceedings of the IEEE*, 78, 10, 1550-1560.