Bai T, Dou HJ, Zhao WX *et al.* An experimental study of text representation methods for cross-site purchase preference prediction using the social text data. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(4): 828–842 July 2017. DOI 10.1007/s11390-017-1763-6

An Experimental Study of Text Representation Methods for Cross-Site Purchase Preference Prediction Using the Social Text Data

Ting Bai [1,2], *Student Member, CCF*, Hong-Jian Dou [1,2], Wayne Xin Zhao [1,2,3,*], *Member, CCF, ACM, IEEE*
Ding-Yi Yang [1], and Ji-Rong Wen [1,2], *Member, CCF, ACM, IEEE*

[1] *School of Information, Renmin University of China, Beijing 100872, China*
[2] *Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing 100872, China*
[3] *Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China*

E-mail: {baiting, hongjiandou}@ruc.edu.cn; batmanfly@gmail.com; ydy@ruc.edu.cn; jirong.wen@gmail.com

Received December 21, 2016; revised May 24, 2017.

Abstract Nowadays, many e-commerce websites allow users to login with their existing social networking accounts. When a new user comes to an e-commerce website, it is interesting to study whether the information from external social media platforms can be utilized to alleviate the cold-start problem. In this paper, we focus on a specific task on cross-site information sharing, i.e., leveraging the text posted by a user on the social media platform (termed as social text) to infer his/her purchase preference of product categories on an e-commerce platform. To solve the task, a key problem is how to effectively represent the social text in a way that its information can be utilized on the e-commerce platform. We study two major kinds of text representation methods for predicting cross-site purchase preference, including shallow textual features and deep textual features learned by deep neural network models. We conduct extensive experiments on a large linked dataset, and our experimental results indicate that it is promising to utilize the social text for predicting purchase preference. Specially, the deep neural network approach has shown a more powerful predictive ability when the number of categories becomes large.

Keywords social media, e-commerce website, purchase preference, deep neural network

1 Introduction

Recent years have witnessed the great success of e-commerce websites, where users can make purchases on a diverse range of products, such as books, food, and courses. To provide users with better service, it is essential to effectively understand and model users' purchase preference. However, one of the major obstacles to tackle this task is that many users are with very few purchases or even without any purchases (i.e., new users), so called cold-start problem[1]. Cold-start recommendation is ubiquitous in various e-commerce websites, which would largely affect the online experiences of new users if it was not effectively solved.

Cold-start recommendation problems can be categorized into two kinds, new-user recommendation and new-item recommendation, in which we do not have enough history information for new users or new items respectively. Traditional collaborative filtering methods, which rely on the user-item interactions, cannot work well on both cases[2]. To solve the cold-start problem, various methods have been proposed[1,3-5] in the literature. A major solution is to utilize the side infor-

Regular Paper
Special Issue on Deep Learning
This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 61502502 and 61502501, the National Basic Research 973 Program of China under Grant No. 2014CB340403 and the Beijing Natural Science Foundation under Grant No. 4162032. Ting Bai was supported by the Outstanding Innovative Talents Cultivation Funded Programs 2016 of Renmin Univertity of China. Wayne Xin Zhao was also supported by the Opening Project of Guangdong Province Key Laboratory of Big Data Analysis and Processing under Grant No. 2017001.
*Corresponding Author

mation from users or items for alleviating the cold-start problem. Specially, most of the existing studies focus on solving the new-item cold-start recommendation. Compared with new items, it is often more difficult to obtain and model the side information of new users.

The focus of this paper is new-user cold-start recommendation. Our work is inspired by an important observation: many online service platforms allow users to login using their existing account information from mainstream social media websites such as Facebook①, Twitter②, Weibo③ or WeChat④. Such a login mechanism is usually called the third-party social login. Social login provides users with a more flexible, general way to engage in multiple online platforms. Hence e-commerce websites can utilize social login to better understand users with external social account information, and make more effective product promotion. Indeed, cross-site user modeling for product promotion has already been utilized in industry. For example, in China, WeChat enables a user to buy products directly by visiting a built-in product navigation page. As another instance, Taobao product adverts can be directly delivered to Weibo users with a personalized advertising board.

The connection between social media and e-commerce websites enhances the sharing of users' information, and the information from social media websites can be leveraged to alleviate the cold-start problem in the e-commerce websites[1,3,6]. Although several efforts have been made[1,3,5-6] to address the cold-start problem on e-commerce platforms by incorporating social media information, most of them extract handcrafted features from social media platforms in a straightforward way. To the best of our knowledge, it is still not well studied how to effectively utilize the heterogenous social media information for product recommendation.

In this paper, we focus on exploring effective ways to utilize the heterogenous information from social media platforms to solve the new-user cold-start problem. We leverage the text posted by a user on social media platform (termed as social text) to infer the user's purchase preference over product categories on the e-commerce platform. Text data is one of the most fundamental data types on social media platforms, and is usually more prevalent and easier to be obtained than the other types of data, e.g., gender and age. More importantly, the social text contains rich comments, discussions and opinions, which is an important resource to reveal users' purchase interests[4,6]. Hence, it is of practical values to explore whether the social text is useful to infer users' purchase behaviors and how to effectively utilize the social text for our prediction task of users' purchase preference.

Specially, we study the cross-site prediction task of users' purchase preference in a full cold-start setting, i.e., only the social text of a user is available for prediction and no historical purchase records are used. We perform an experimental study of the social text representation methods in cross-site purchase preference prediction task. To be more specific, we study two major kinds of text representation methods for predicting cross-site purchase preference. For the first kind, we mainly consider shallow textual features, including term-based surface semantic representations and shallow neural features. However, these shallow and superficial features may not be effective enough to be used in cross-site tasks[7], since these features typically model the syntactic context of words and cannot capture the script beyond the surface form of the language, e.g., sentiment[8-9]. Inspired by the recent success of deep learning techniques in NLP (natural language processing) tasks, e.g., text classification[10], sentiment classification[8], semantic parsing[11] and sentence modeling[12], we further propose to learn deep textual features using multiple deep neural network models in our cross-site prediction task of users' purchase preference.

In order to carry out this study, we select Weibo (the largest Chinese microblogging service) as the social media platform and JingDong (one of the largest Chinese B2C e-commerce websites) as the e-commerce platform. A large dataset is constructed for our task, containing a total of 11 334 linked users across these two platforms. In our work, we leverage users' social text to predict their purchase preference which is characterized as the distribution over the product categories on JingDong. We first quantitatively analyze the correlation between a user's social text on Weibo and product descriptions on JingDong. Our finding is that the more similar a user's social text is to the descriptions of products from a certain category, the more likely the user

①https://www.facebook.com, May 2017.

②http://twitter.com, May 2017.

③www.weibo.com, May 2017.

④www.wechat.com, May 2017.

is to purchase products from the category. To further model the effectiveness of the social text, we formally formulate the prediction task of users' purchase preference as a category ranking problem only using the social text. We extensively examine a large set of text representation methods for the current task, including both shallow and deep text representation methods.

Our contributions are summarized as follows. 1) We take the initiative to study how the social text can be utilized for the cross-site purchase prediction on a large linked dataset, and our experimental results have indicated that the social text is indeed useful to predict the purchase preference on the e-commerce platform. To the best of our knowledge, few studies address this task in a systematic and comprehensive way. 2) We quantitatively analyze the correlations between users' social text and purchase behavior, and we find that if the social text posted by a user is more similar to the product descriptions of a category, he/she is more likely to make purchases. 3) We propose two approaches to represent the social text for predicting users' purchase preference, and extensive experiments demonstrate the powerful predictive ability of our deep neural network models, especially when the number of product categories becomes large.

The rest of the paper is organized as follows. Section 2 describes the construction of data collection and quantitative analysis. In Section 3, we first present the problem statement, and then propose two kinds of prediction approaches. Section 4 presents the experimental results and analysis. Section 5 reviews relevant previous work and Section 6 concludes our work.

2 Data Collection and Analysis

Our task requires a dataset of linked users from both an e-commerce website and an online social networking website. We first construct the dataset, and then we make a quantitative analysis of the correlations between the social text and product descriptions.

2.1 Dataset Construction

We use the shared dataset in [6] which contains 11 334 linked users between Weibo and JingDong platforms. The statistics of the linked dataset are summarized in Table 1. For privacy consideration, Weibo IDs and JingDong IDs of all linked users are replaced by anonymized unique IDs.

Weibo Data. We only utilize the published microblogging text of linked users. Other types of information (e.g., gender and age) are discarded since they are not our focus in this paper. The average number of microblogging text posted by each linked user is 41.

Table 1. Statistics of the Linked Dataset

	Statisties
Number of users	11 334
Number of products	246 416
Average number of tweets	41
Average number of products	52

JingDong Data. The linked dataset contains 246 416 products, and the average number of products each linked user purchases is 52. To characterize the category of a product, we follow the navigation page on JingDong website.

As shown in Fig.1, it organizes the products in a three-level category taxonomy. For example, the first-level category "Sports & Outdoors" contains the second-level categories like "Sports Clothing & Footwear", "Cycling", and "Golf". Each second-level category is further decomposed into the third-level categories like "Cycling Wear" and "Golf Apparel". In our task, we do not consider the associations between categories from two different levels. Instead, we perform the prediction task in three category levels separately. Each product is associated with three category labels in different levels. We summarize the basic statistics of JingDong data (i.e., the number of categories, the average number of products, and the average number of users who purchase products in some category) in three category levels. The statistics are shown in Table 2.

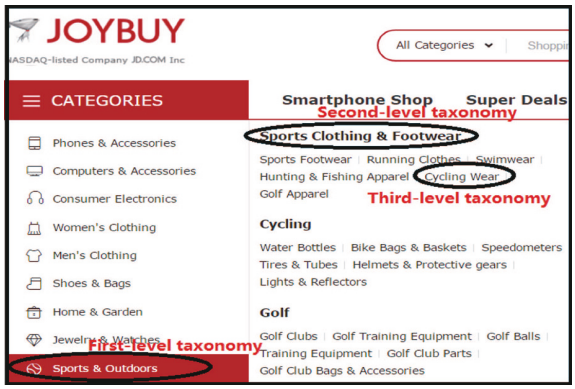


Fig.1. Three-level product category taxonomy on JingDong.

Table 2. Statistics of JingDong Data in Three Category Levels

Level	Number of Categories	Average Number of Products	Average Number of Users
First-level	12	20 535	5 492
Second-level	27	9 126	3 292
Third-level	166	1 484	1 024

2.2 Quantitative Analysis of the Correlations Between the Social Text and Product Descriptions

The social text contains rich comments, discussions and opinions, which is an important resource to reveal users' purchase interests[4,6]. Hence, we hypothesize that the social text information of a user is useful in predicting his/her purchase preference. There has been a lack of quantitative analysis of the correlations between a user's social text and purchase behavior on large datasets. To fulfill this purpose, we quantitatively analyze this problem in two different aspects, i.e., products and users respectively.

To understand whether there exist some correlations between the social text posted by a user and the products he/she purchased, we compute the semantic similarity between the social text and product descriptions. For each user, we aggregate all his/her microblogs into a document, called user document. For each product category, we aggregate the descriptions of all products in the category into a document, called product document. We represent these two kinds of documents using standard Vector Space Model with tf-idf weighting. Then, we can compute the similarity between a user and a category. We can make the following observations.

1) Products are more likely to be purchased by the users whose social text is more similar to the product descriptions. Given a category, we apply the above similarity method to identify the most similar 100 users (called similar users for short) and the least similar 100 users (called dissimilar users for short). We would like to check whether the similar users are more likely to have a higher purchase preference degree than the dissimilar users in the given category. Given a user, the purchase preference degree on a category is calculated as the normalized probability by dividing the purchase frequency in a category by the total purchase frequency of the user. For ease of analysis, we further discretize users into two bins based on the normalization purchase probability: $[0, 0.5]$ (low purchase preference), $(0.5, 1]$ (high purchase preference), and present the results based on the first-level categories in Fig.2. For each category, we present the purchase probability distributions by two groups, i.e., similar (left bar) and dissimilar (right bar) user groups.

It can be observed that in most product categories, the amount of users with high purchase preference in the similar user group is larger than that in the dissimilar user group. Interestingly, the gaps in the categories of "Kids & Baby" and "Pet Supplies" between two user groups are the most significant. It indicates that if the tweets of a user are closely related to the topics of "Kids & Baby" and "Pet Supplies", he/she is likely to make more purchases from these two categories.

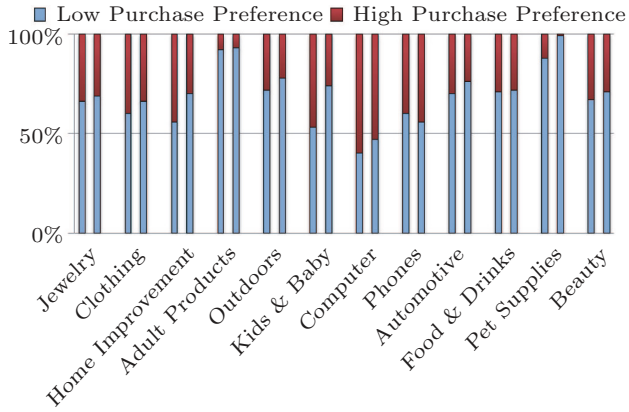


Fig.2. Purchase preference distributions of 12 first-level categories. The purchase preference is divided into two bins, i.e., high and low. Given a category, the left and the right bars correspond to the similar and the dissimilar user groups in terms of text similarity respectively.

2) Users are more likely to purchase products from the category in which product descriptions are more similar to their posted social text. Our previous analysis focuses on the aspect of products, and we find that products are more likely to be purchased by the users whose social text is more similar to the product descriptions. We further analyze the problem in the aspect of users. Given a user, we apply the similar method to compute the most similar product category (called top category for short) and the least similar product category (called bottom category for short) in which product descriptions are the most similar or the least similar to the user's social text. We would like to check whether the user is more likely to purchase products in the top category than in the bottom category.

We first conduct the analysis on the top category. Given a user, we rank product categories descendingly by the purchase frequencies, and obtain the ranking position of his/her top category. Since we have 12 first-level product categories, we set up 12 user bins corresponding to the ranking position of the top category. We put a user into a bin according to the ranking position of his/her top category. Finally, we calculate the number of users in each bin, and derive the distribution of users over the 12 user bins. Similarly, we can perform the analysis on the bottom category. In this way, we obtain two distributions of users according to either

top category ranking or bottom category ranking. The user distributions of the top and the bottom categories on the 12 first-level categories are shown in Fig.3.

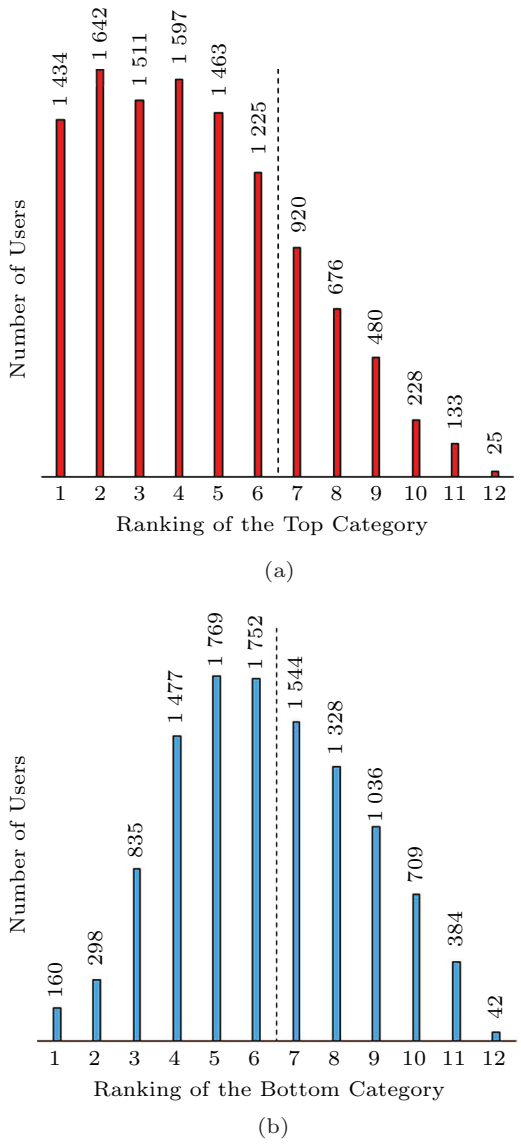


Fig.3. User distributions of the top and the bottom categories on the 12 first-level categories. (a) Top category. (b) Bottom category.

It can be observed that 79% users are classified into the top six positions according to their top category ranking. As a comparison, the number of users only accounts for 40% in the top six positions according to their bottom category ranking. The two different distribution patterns in ranking top and bottom categories indicate that a user is more likely to have a higher purchase preference of the products that his/her social text is more similar to.

The above findings from both aspects of products and users show that if the social text posted by a user is more similar to the product descriptions in a category, then the user is more likely to purchase products from this category. Indeed, the previous study[5] had found that users are likely to expose the commercial intents in their tweets.

3 Predicting Users' Purchase Preference Based on the Social Text

So far, we have shown that there indeed exist some correlations between a user's social text and the descriptions of products that he/she purchased on an e-commerce platform. The next practical question is whether given the text information posted by a user on social media, we can leverage it for predicting his/her purchase preference on the e-commerce platform. Such a cross-site application has commercial values for solving the cold-start problem on e-commerce platforms. In what follows, we first formally define the prediction task of users' purchase preference based on the social text, and then present two major approaches to represent users' social text. The notations in our prediction task are presented in Table 3.

Table 3. Notations in Prediction Models

Notation	Description
$\mathcal{U}^{(\mathrm{L})}$	Linked user set
$\mathcal{C}$	Set of product categories
c_k	Product category, such that $c_k \in \mathcal{C}$
d_{u}	User document
d_{p}	Product document
$\mathcal{D}_{\mathrm{U}}$	User dictionary of d_{u}
$\mathcal{D}_{\mathrm{P}}$	Product dictionary of d_{p}
$\mathcal{D}_{\mathrm{C}}$	Common dictionary, $\mathcal{D}_{\mathrm{C}} = \mathcal{D}_{\mathrm{U}} \cap \mathcal{D}_{\mathrm{P}}$
$\boldsymbol{p}_{\mathrm{u}}$	Vector of preference degree
$\hat{\boldsymbol{p}}_{\mathrm{u}}$	Vector of predicted preference degree
$\boldsymbol{v}_i$	Embedding of the i-th word in the input sentence of neural network models

3.1 Problem Statement

Our task is to predict the purchase preference of product categories given the social text of a user. To solve this task, we assume that a linked user set $\mathcal{U}^{(\mathrm{L})}$ is available. Given a user $u \in \mathcal{U}^{(\mathrm{L})}$, we can obtain both his/her social text data and purchase history. As mentioned above, the social text of a user u is all the microblogs he/she posted, and it can be aggregated into a single document d_{u}, called user document. And let $\mathcal{C}$ denote the set of product categories in an e-commerce website.

We first define the purchase preference of user u as the purchase distribution over $\mathcal{C}$.

$$p_{u,c_k} = \frac{purc(u, c_k)}{\sum_{c \in \mathcal{C}} purc(u, c)},$$

where $purc(u, c_k)$ is the number of products purchased by user u from category $c_k \in \mathcal{C}$. Then, we would like to learn a prediction function f which takes the user document d_{u} as input and returns a predicted purchase preference vector $\hat{\boldsymbol{p}}_{\mathrm{u}}$. It is formally defined as follows:

$$\hat{\boldsymbol{p}}_{\mathrm{u}} = f(d_{\mathrm{u}}),$$

where $\hat{\boldsymbol{p}}_{\mathrm{u}}$ is a $|\mathcal{C}|$-dimensional vector, where each element $\hat{p}_{u,c_k}$ is the user's predicted preference degree of category c_k. With $\hat{\boldsymbol{p}}_{\mathrm{u}}$, we can produce a predicted ranking list of the categories for user u. In our work, we mainly focus on the relative order between categories; hence the task becomes a category ranking problem[5].

Since we learn from the social text of a user for category ranking prediction, it is vital to effectively represent the original social text. In our work, we consider two major approaches: one extracts textual features using traditional text representation models or shallow neural network models, and the other learns effective text representations using deep neural network models. In what follows, we introduce the two prediction approaches: a traditional classification approach with shallow textual features and a neural network approach with deep textual features respectively.

3.2 Traditional Classification Approach with Shallow Textual Features

To solve the above prediction problem, traditional machine learning algorithms such as logistic regression (LR), naive Bayes (NB), and support vector machine (SVM) have been studied in [3]. We use SVM model which achieves the best performance in [3] for our purchase preference prediction task. For each category c, we build an SVM classification model. The input of the SVM model is the feature vector of a user extracted or learned from the text data. The SVM model learns the feature weights based on the following objective function for classification:

$$\text{minimize } \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + C \sum \xi_c \qquad (1)$$

$$\text{subject to } y_{\mathrm{u}}(\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{v}_{\mathrm{u}} + b) \geqslant 1 - \xi_c, \xi_c \geqslant 0,$$

where y_{u} is the preference label of a user u in category c, $\boldsymbol{v}_{\mathrm{u}}$ is the user feature vector represented by the following text representation methods, and $\boldsymbol{w}$ is the feature weight vector learned from the training model. Following [3], if a user u has bought at least one product from category c, we set $y_{\mathrm{u}} = 1$; otherwise we set $y_{\mathrm{u}} = 0$. For training, positive examples are users who buy at least one item in the category, and an equal number of random negative examples are provided. During testing, for each user in the test set, SVM returns a confidence score[13] which we used for ranking. We use SVM$^{\mathrm{light}}$ with a radial basic function (RBF) kernel[3]. By minimizing the objective function in (1), we learn parameters by grid search on a subset of the training set. Our focus is to design and compare different text representation methods, i.e., how to construct the feature vector $\boldsymbol{v}_{\mathrm{u}}$ for a user u.

Vector Space Model (*VSM*)[14]. The vector space model is an algebraic model to represent text as vectors of identifiers. It is widely used in information retrieval and relevancy ranking. We use the social text of all users to generate the user dictionary $\mathcal{D}_{\mathrm{U}}$ which contains 41 744 words. A user feature vector $\boldsymbol{v}_{\mathrm{u}} \in \mathbb{R}^{|\mathcal{D}_{\mathrm{U}}|}$ can be considered as a representation of all words in user document d_{u} over the dictionary $\mathcal{D}_{\mathrm{U}}$. The weighting of a word in d_{u} is determined by term frequency-inverse document frequency (i.e., tf-idf) model[15].

Topic Model (*TM*)[16]. Following the same idea from [17], we use the standard LDA (Latent Dirichlet Allocation) to obtain the topic distributions of each user document d_{u}. In LDA, we can learn a topic distribution for each user document, called user-topic distribution. A user's feature vector can be represented as the user-topic distribution. We set the number of topics T to 100 in our experiment, which largely reduces the number of dimensionality to work with. The topic model generates condense semantic units, and it is easier to interpret and understand.

Word Embedding Model (*WEM*)[18]. WEM is a shallow neural network model. It encodes a word into a latent vector and ensures similar words to be close in the latent space. We employ the CBOW model[18] implemented by tool word2vec to learn the distributed representation of words. The core idea of CBOW is using the surrounding words in a context to predict

[5] Our task setting is similar to that in [2], which also predicts users' purchase preference using social media data. A major difference is that multiple data signals are considered as the input in [2], while we assume only the social text is available. Another difference is that very simple text features are used in [2], while we develope more effective text representation methods for the current task.

a target word. Formally, given a sequence of words $\mathcal{S}_{\mathrm{w}} = \{w_1, w_2, ..., w_n\}$ in training data, and a word w_t in $\mathcal{S}_{\mathrm{w}}$, the surrounding words in a content of w_t are denoted as $context(w_t)$. The target of this model is to maximize the average log probability, and it is defined as:

$$\frac{1}{n} \sum_{w_t \in \mathcal{S}_{\mathrm{w}}} \log p(w_t | context(w_t)).$$

After training, the words with similar semantic meaning are likewise close in the low dimensional vector space. Finally, we average the vectors of all words in a user document d_{u} and normalize the vector as a user's feature vector $\boldsymbol{v}_{\mathrm{u}}$. The word embedding model addresses the problem of classic text representation approaches which fail to capture words' contextual semantics[19-20].

Cross-Domain Vector Space Model (*CD-VSM*). Since it has been shown that there may exist semantic correlations between the social text and product descriptions in Subsection 2.2, we consider applying a simple cross-site semantic matching method. To emphasize the product-related information in the social text, we propose to construct a common dictionary $\mathcal{D}_{\mathrm{C}} = \mathcal{D}_{\mathrm{U}} \cap \mathcal{D}_{\mathrm{P}}$ which consists of user dictionary $\mathcal{D}_{\mathrm{U}}$ on the social media platform and product dictionary $\mathcal{D}_{\mathrm{P}}$. $\mathcal{D}_{\mathrm{C}}$ contains 8 018 words. A user's feature vector of the social text $\boldsymbol{v}_{\mathrm{u}} \in \mathbb{R}^{|\mathcal{D}_{\mathrm{C}}|}$ can be represented as the distribution of all words in the user document d_{u} over the common dictionary $\mathcal{D}_{\mathrm{C}}$. The weighting of a word in d_{u} is determined by term frequency-inverse document frequency (i.e., tf-idf) model[15].

Product Category Similarity Model (*PCSM*). Since we predict users' purchase preference of product categories, we utilize the information of product category to design the social text features of a user. Similar to the VSM model, we use the common dictionary $\mathcal{D}_{\mathrm{C}}$ to construct the feature vectors of a user's text $\boldsymbol{v}_{\mathrm{u}}$ and a product category $\boldsymbol{v}_{c_i}$. Then we compute the similarity score $s(\boldsymbol{v}_{\mathrm{u}}, \boldsymbol{v}_{c_i})$ between $\boldsymbol{v}_{\mathrm{u}}$ and $\boldsymbol{v}_{c_i}$ by the inner product of the two vectors $s(\boldsymbol{v}_{\mathrm{u}}, \boldsymbol{v}_{c_i}) = \boldsymbol{v}_{\mathrm{u}} \cdot \boldsymbol{v}_{c_i}$. The features in this model are defined as:

$$\boldsymbol{v}_{\mathrm{s}} = \{s(\boldsymbol{v}_{\mathrm{u}}, \boldsymbol{v}_{c_i}) | i = 1, ..., |\mathcal{C}|\}.$$

The vector dimension is equal to the number of categories.

3.3 Neural Network Approach with Deep Textual Features

It may be not effective enough if the above shallow textual features are directly used for classification task[7], since these features typically model the syntactic context of words and cannot capture the script beyond the surface form of the language, e.g., sentiment[8-9]. Inspired by the recent success of deep learning techniques in many NLP tasks[8,10-12], we further propose to learn deep textual features using multiple deep neural network models in our cross-site prediction task of users' purchase preference.

The input of deep neural network models is a sentence represented as $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n\}$, where n is the sentence length, and $\boldsymbol{v}_i$ is the d-dimensional word embedding of the i-th word in the sentence. The output of deep neural network models is the vector $\hat{\boldsymbol{p}}_{\mathrm{u}}$. Each dimension of $\hat{\boldsymbol{p}}_{\mathrm{u}}$ is the purchase probability of each product category. During training, we minimize the objective function Z, which is defined as:

$$Z = -\frac{1}{n} \sum_{k=1}^{|\mathcal{C}|} (p_{u,c_k} \ln \hat{p}_{u,c_k} + (1 - p_{u,c_k}) \ln(1 - \hat{p}_{u,c_k})),$$

where p_{u,c_k} is a user's real purchase preference of product category c_k, and $\hat{p}_{u,c_k}$ is the predicted purchase preference of category c_k. Recent researches[21-23] show unsupervised pre-training is helpful to neural network to converge to a better local minima. We use CBOW model[18] to initialize input embedding vector, of which the dimension L is set to 100 in our experiment.

The four deep neural network models are shown in Fig.4.

Convolutional Neural Network (*CNN*). In recent years, a convolutional neural network (CNN) model[21] has subsequently achieved excellent results in many NLP tasks[11-12]. The architecture of the CNN model is shown in Fig.4(a). In the training process, a convolution operation uses a filter $\boldsymbol{w}$ which has a window of h words to produce a new feature. For example, a feature f_i generated from a window h of words sequence $\boldsymbol{x}_{i:i+h-1}$ is defined as follows:

$$f_i = g(\boldsymbol{w} \cdot \boldsymbol{x}_{i:i+h-1} + b),$$

where b is a bias term and g is a non-linear function such as hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ to produce a feature map $\boldsymbol{f} = \{f_1, f_2, ..., f_{n-h+1}\}$. Then a max-pooling operation[24] is applied on the feature map to capture the most important feature by taking the maximum value $\hat{f} = \max\{\boldsymbol{f}\}$. Finally, we use a softmax function to generate the probability distribution over labels.

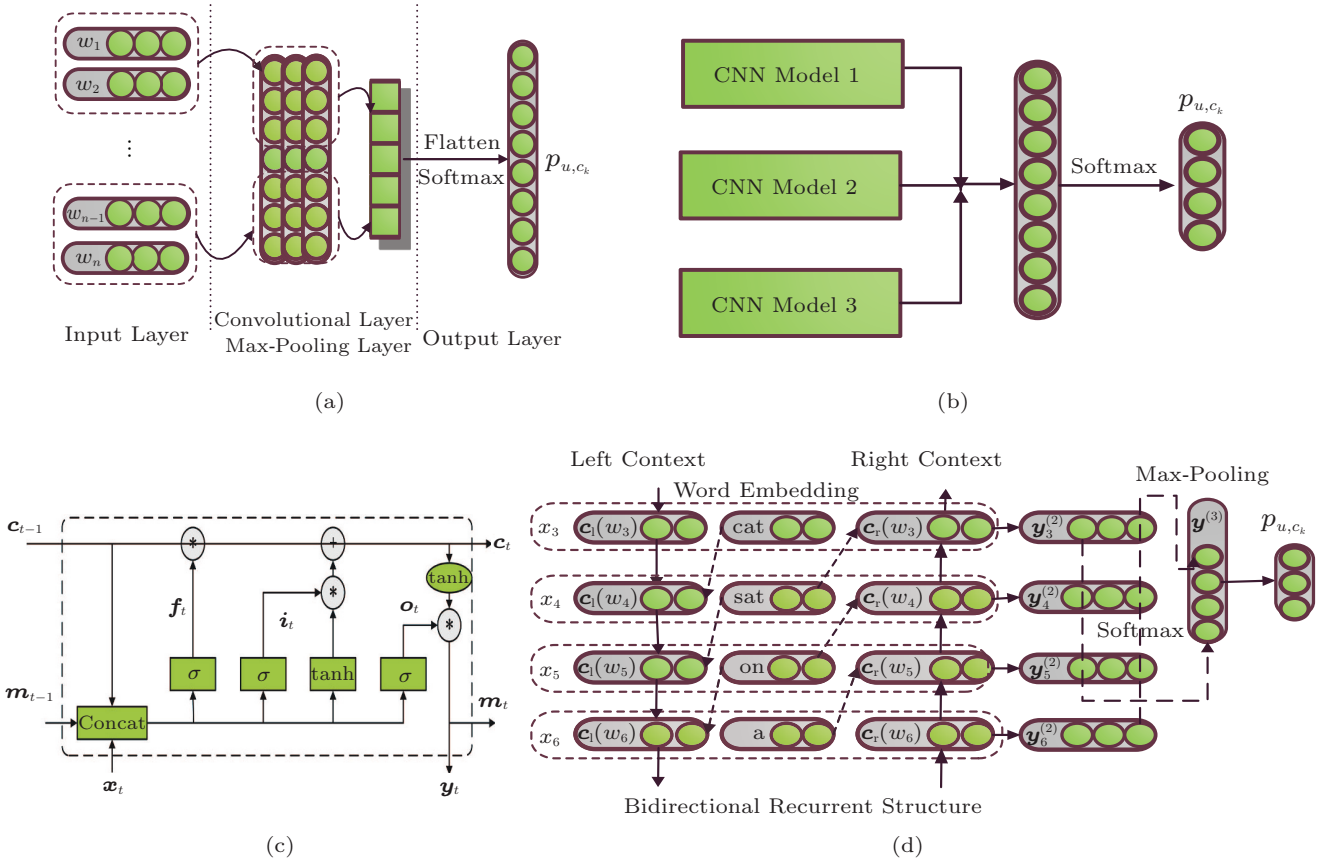


Fig.4. Deep neural network models. (a) CNN. (b) HCNN. (c) LSTM. (d) BRNN.

Hybrid Convolutional Neural Network (*HCNN*). The above simple CNN model uses a specific convolutional kernel with a fixed window[24-25]. However, it is difficult to determine the window size. A small window size may result in the loss of some critical information, whereas large windows result in an enormous parameter space[10]. To address this problem, inspired by the model architecture in [25], we design a hybrid convolutional neural network (HCNN) with different kernel lengths, as shown in Fig.4(b). The model captures the context information with three different window sizes, and then merges features from the three models together. HCNN generates more features from different convolutional layers than CNN and it also reduces the influence of improper window size.

Long Short-Term Memory (*LSTM*) *Model.* A recurrent neural network (RNN) is a class of neural network for processing sequential data and it uses internal memory to process arbitrary sequences of inputs. However, when learning long-term dependencies, the gradients propagated over many stages tend to vanish. To solve this problem, the long short-term memory (LSTM) model[26] introduces self-loops to produce paths where the gradients can flow long durations. The architecture is depicted in Fig.4(c). Let $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ be the input and the output signal for an LSTM network at time t respectively, and $\boldsymbol{c}_t$ and $\boldsymbol{m}_t$ be the cell activation and the output activation respectively. The forward pass from $\boldsymbol{x}_t$ to $\boldsymbol{y}_t$ follows the equations[27]:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_{\mathrm{ix}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{im}}\boldsymbol{m}_{t-1} + \boldsymbol{W}_{\mathrm{ic}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\mathrm{i}}),\\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_{\mathrm{fx}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{fm}}\boldsymbol{m}_{t-1} + \boldsymbol{W}_{\mathrm{fc}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\mathrm{f}}),\\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_{\mathrm{ox}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{om}}\boldsymbol{m}_{t-1} + \boldsymbol{W}_{\mathrm{oc}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\mathrm{o}}),\\
\boldsymbol{m}_t &= \boldsymbol{o}_t \odot h(\boldsymbol{c}_t),\\
\boldsymbol{y}_t &= \phi(\boldsymbol{W}_{\mathrm{ym}}\boldsymbol{m}_t + \boldsymbol{b}_{\mathrm{y}}),\\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot g(\boldsymbol{W}_{\mathrm{cx}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{cm}}\boldsymbol{m}_{t-1} + \boldsymbol{b}_{\mathrm{c}}),
\end{aligned}$$

where $\boldsymbol{W}_{\mathrm{cx}}$, $\boldsymbol{W}_{\mathrm{ix}}$, $\boldsymbol{W}_{\mathrm{fx}}$ and $\boldsymbol{W}_{\mathrm{ox}}$ are the weights of the LSTM inputs, $\boldsymbol{W}_{\mathrm{cm}}$, $\boldsymbol{W}_{\mathrm{im}}$, $\boldsymbol{W}_{\mathrm{fm}}$ and $\boldsymbol{W}_{\mathrm{om}}$ are the weights of the LSTM activations, $\boldsymbol{b}_{\mathrm{c}}$, $\boldsymbol{b}_{\mathrm{i}}$, $\boldsymbol{b}_{\mathrm{f}}$ and $\boldsymbol{b}_{\mathrm{o}}$ are the biases, and $\boldsymbol{W}_{\mathrm{ic}}$, $\boldsymbol{W}_{\mathrm{fc}}$, $\boldsymbol{W}_{\mathrm{oc}}$ are diagonal peephole connections from the cell to the gate signals. $\boldsymbol{W}_{\mathrm{ym}}$ and $\boldsymbol{b}_{\mathrm{y}}$ are the weights and biases in the final output layer. σ, g and h are nonlinear functions. The sigmoid function is used for σ and the tanh function is used for g

and h. $\odot$ is elementwise product of the vectors and ϕ is the softmax function in the output layer.

Bidirectional Recurrent Neural Network (*BRNN*). In RNN, the state at time t only captures information from the past $x^{(1)}, ..., x^{(t-1)}$ and the present input $x^{(t)}$. However, when processing the sequence text, the semantics of the current word is correlated with the whole input sequence. The bidirectional recurrent neural network (BRNN)[28] combines a RNN that moves forward from the start of the sequence with another RNN that moves backward from the end of the sequence through time beginning. BRNN can capture the contextual information to the greatest extent possible when learning word representations. The model structure is shown in Fig.4(d). We define $\boldsymbol{c}_{\mathrm{l}}(w_i)$ as the left context of word w_i and $\boldsymbol{c}_{\mathrm{r}}(w_i)$ as the right context of word w_i. Both $\boldsymbol{c}_{\mathrm{l}}(w_i)$ and $\boldsymbol{c}_{\mathrm{r}}(w_i)$ are dense vectors with $|c|$ real value elements. They are calculated as follows:

$$\boldsymbol{c}_{\mathrm{l}}(w_i) = f(\boldsymbol{W}^{\mathrm{l}}\boldsymbol{c}_{\mathrm{l}}(w_{i-1}) + \boldsymbol{W}^{\mathrm{sl}}\boldsymbol{e}(w_{i-1})),$$
$$\boldsymbol{c}_{\mathrm{r}}(w_i) = f(\boldsymbol{W}^{\mathrm{l}}\boldsymbol{c}_{\mathrm{r}}(w_{i+1}) + \boldsymbol{W}^{\mathrm{sl}}\boldsymbol{e}(w_{i+1})),$$

where $\boldsymbol{e}(w_{i-1})$ is the word embedding of word w_{i-1}, and $\boldsymbol{c}_{\mathrm{l}}(w_{i-1})$ is the left-side context of the previous word w_{i-1}. $\boldsymbol{W}^{\mathrm{l}}$ is a matrix that transforms the hidden layer into the next hidden layer, and $\boldsymbol{W}^{\mathrm{sl}}$ is a matrix that is used to combine the semantics of current word with the next word's left context. f is a non-linear activation function. The right-side context $\boldsymbol{c}_{\mathrm{r}}(w_i)$ is calculated in a similar manner. Using the contextual vector $\boldsymbol{c}_{\mathrm{l}}(w_i)$ and $\boldsymbol{c}_{\mathrm{r}}(w_i)$, we define the representation of word w_i as $\boldsymbol{x}_i$, which is defined as:

$$\boldsymbol{x}_i = (\boldsymbol{c}_{\mathrm{l}}(w_i); \boldsymbol{e}(w_i); \boldsymbol{c}_{\mathrm{r}}(w_i)).$$

After we obtain the representation $\boldsymbol{x}_i$ of the word w_i, we apply a linear transformation together with the tanh activation function to $\boldsymbol{x}_i$ and send the result to the next layer.

$$\boldsymbol{y}_i^{(2)} = \tanh(\boldsymbol{W}^{(2)}\boldsymbol{x}_i + \boldsymbol{b}^{(2)}),$$

where $\boldsymbol{y}_i^{(2)}$ is a latent semantic vector and $\boldsymbol{b}^{(2)}$ is the bias vector. We finally build a max-pooling layer upon the bidirectional recurrent structure, which automatically captures the key component in the text. BRNN captures the semantics of all the left and the right side contexts. It may be more able to disambiguate the meaning of the word w_i compared with unidirection RNN and CNN.

Parameter Settings. We implement our models in PYTHON using the library KERAS⑥. We take 10% of users in the training data as the validation set to optimize parameters in our models and report the detail parameter settings with which the model achieves the best performance. The parameter settings are shown in Table 4.

Table 4. Detailed Parameter Settings in Our Deep Neural Network Models

Model	Parameter Setting
CNN	Conv1D (nb filter=128, filter length=4, activation="tanh"), Max-Pooling1D (pool length=5), Dropout (p=0.5), Dense (output dim=128, activation="tanh")
HCNN	Conv1D filer length in each CNN model (filter length=3, 4, 5); other parameters are the same as those in CNN model
LSTM	LSTM (output dim=64), Dense (output dim=128, activation="tanh")
BRNN	Bidirectional (LSTM(output dim=64)), MaxPooling1D (pool length=140), Dense (output dim=128, activation="tanh")

Note: For all the models, the pre-trained input embedding vector dimension L=100; the output vector is generated by Dense (output dim= $|\mathcal{C}|$, activation="sigmoid").

4 Experiments

In this section, we conduct experiments to compare the performance of different social text representation models on our prediction task of users' purchase preference.

4.1 Methods to Compare

Gold Standard. We regard the real purchase preference of a user as the gold standard in our experiments.

Baseline Methods. A reasonable system is ranking product categories according to their popularity, i.e., the sales volume in the training of the category. We also compare our deep neural network models with SVM, TM, WEM, CD-VSM, PCSM which use shallow textual features to represent the social text. We make the comparison of different text representation models in Table 5.

4.2 Evaluation Metrics

Given a product, each candidate method will produce an ordered list of product categories. Hence, we adopt two ranking based metrics to evaluate the predicting results, which are outlined below.

⑥https://keras.io/, May 2017.

Table 5. Comparison of Different Text Representation Models

Model	Characteristics
Popularity	It simply counts the total purchase frequency of all the users
VSM	It utilizes the tf-idf method to weight the terms
TM	It captures the latent topic semantic distribution of a document
WEM	Semantical information of words in the social text is captured
CD-VSM	Product descriptions are considered.
PCSM	It computes the similarity between the social text and product
CNN	It better captures the local structures of the input data
HCNN	More contextual information is captured by different kernel lengths
LSTM	It is beneficial to capture the semantics of long text
BRNN	It reserves a larger range of the word ordering

Normalized Discounted Cumulative Gain (*NDCG*). For each user, we define discounted cumulative gain (DCG)[29] at position k as:

$$DCG@k = \sum_{i=1}^{k} \frac{w(i)}{\log(i+1)},$$

where $w(i)$ is the relevance weight of the product category at position i. We set $w(i)$ as a user's actual purchase preference p_{u,c_i}. We further define IDCG[30] (i.e., ideal DCG) at position k as the DCG of the list ranked by real purchase preference. Then, we compute NDCG at position k as follows:

$$NDCG@k = \frac{DCG@k}{IDCG@k}.$$

Precision at Rank k ($P@k$). Given a predicted list of product categories, we define $P@k$ as:

$$P@k = \frac{|\mathcal{P}_k \cap \mathcal{T}_k|}{k},$$

where $\mathcal{P}_k$ and $\mathcal{T}_k$ denote the sets of product categories from the predicted ranked list and actual purchase records for the top k product categories respectively.

Note that although we are solving a ranking problem, we do not use any ranking correlation coefficient for our evaluation (e.g., Spearman or Kendall Tau). In our cases, we are not interested in computing how similar two rankings are as a whole, but just in how good an algorithm is in catching the correct categories as early as possible. In this case, NDCG and precision at rank k are more reliable measures.

We evaluate our ranking models using 10-fold cross validation in order to reliably compute statistical significance values. For each fold we use 90% of the users for training and 10% for testing. We compute the above measures for each fold by averaging the measures over all testing users.

4.3 Experimental Results and Analysis

We present the results on the prediction of users' purchase preference in Table 6. It is interesting to see that the number of categories has influence on the prediction performance. Specially, the deep neural network models have shown a more powerful predictive ability when the number of categories becomes large. Our findings are as follows.

• In the prediction of the first-level categories, our deep neural network models show advantages over popularity and SVM models on $NDCG@5$. PCSM has

Table 6. Performance Comparison on the Results of Predicting Purchase Preference

Model		First-Level				Second-Level				Third-Level			
		$P@1$	$P@5$	$NDCG@1$	$NDCG@5$	$P@1$	$P@5$	$NDCG@1$	$NDCG@5$	$P@1$	$P@5$	$NDCG@1$	$NDCG@5$
Popularity		0.469	**0.680**	0.664	0.728	0.527	0.539	0.690	0.657	0.428	0.312	0.589	0.481
SVM	VSM	0.461	0.620	0.656	0.719	0.519	0.485	0.678	0.650	0.368	0.273	0.515	0.434
	TM	0.483	0.615	0.673	0.713	0.537	0.465	0.698	0.625	0.415	0.237	0.571	0.421
	WEM	0.476	0.617	0.668	0.726	0.538	0.520	0.696	0.666	0.394	0.288	0.551	0.458
	CD-VSM	0.470	0.620	0.667	0.725	0.535	0.486	0.687	0.658	0.369	0.271	0.521	0.440
	PCSM	0.496	0.636	**0.685**	0.728	**0.553**	0.479	**0.713**	0.647	0.411	0.259	0.571	0.442
DL	CNN	**0.497**	0.634	0.674	0.731*	0.540	0.541	0.703	0.672*	0.436*	0.378*	0.598*	0.523*
	HCNN	0.463	0.621	0.656	0.724	0.542	0.536	0.704	0.674*	0.439*	0.377*	0.592	0.522*
	LSTM	0.487	0.631	0.675	0.731*	0.540	**0.558***	0.700	**0.683***	0.436*	0.378*	0.598*	0.523*
	BRNN	0.488	0.649	0.676	**0.733***	0.546	0.556*	0.711	0.661	**0.441***	**0.378***	**0.600***	**0.523***

Note: Symbol $*$ indicates the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the popularity and SVM models.

a matched performance with deep neural network approach. All models except VSM perform better than popularity at $P@1$ and $NDCG@1$; however, popularity is still a strong baseline as more categories evaluated (i.e., $k = 5$). It may be caused by the imbalance of sales volume in each category when products are classified into 12 categories.

• In the prediction of the second-level categories, our deep neural network approach starts showing significant advantages over the popularity and traditional classification approach SVM with one exception model PCSM on $P@1$ and $NDCG@1$. PCSM constructs features by directly computing the similarity between the social text and product categories, and thus it may be effective to make predictions.

• As the number of the third-level categories increases to 166, our deep neural network approach shows significant advantages over the popularity and the traditional classification approach SVM. The traditional classification approach with shallow textual features performs the worst. It implies that the shallow features cannot effectively capture the useful information in the social text to make more accurate purchase predictions, while the deep textual features learned by neural network models do the best. BRNN which combines the advantage of recurrent neural model and convolutional neural model performs the best within deep neural network models.

In addition, it can be observed that the prediction performance is the highest in the second-level categories instead of in the first-level categories. It may be caused by the imbalance of sales volume in each category in the first-level categories. We rank product categories descendingly by their sales volume, e.g., in the top three categories, the sales volume accounts for a total proportion of 46%, while in the least three categories, sales volume only accounts for a total proportion of 5%. Thus, predictions may heavily skew to the categories with large sales volume, while in the prediction of the second-level categories, the product category with large sales volume is subdivided into different subclasses, which relieves the imbalance of sales volume in product categories, and the subdivided categories are more matched to the purchase preference of the majority of users. It makes the performance higher in the second-level categories than in the first-level categories. In the prediction of the third-level categories, the overall performance of models is the worst, because it is more difficult to make accurate prediction among the large number of categories.

The above experiments have shown the performance of different models in predicting users' purchase preference. We further use an example to present the prediction results. We randomly choose five microblogs posted by a user on the social media platform. The five microblogs are as follows.

1) The makeup in Hong Kong is really cheap.

2) Mountain hiking is a good way to lose weight!

3) Celebration for my baby's completion of his first month of life. But it is difficult to give a name for my baby, any suggestions?

4) I don't know why he is unwilling to call me. Disappointed.

5) I purchased "KONKA XQB50-5001 automatic washing machine" in JingDong Mall, just carefully write the review: can't drying.

We choose BRNN to represent the deep neural network approach and make predictions in the first-level categories. The top five predicted product categories in each model are shown in Table 7.

Table 7. Top Five Predicted Product Categories

Model	Category
Gold standard	Kids & Baby, Clothing, Beauty, Automotive, Computer
Popularity	Computer, Phones, Home Improvement, Beauty, Clothing
VSM	Computer, Phones, Home Improvement, Automotive, Clothing
TM	Kids & Baby, Automotive, Computer, Food & Drinks, Beauty
WEM	Computer, Phones, Home Improvement, Kids & Baby, Automotive
CD-VS	Computer, Home Improvement, Kids & Baby, Phones, Clothing
PCSM	Kids & Baby, Automotive, Food & Drinks, Phones, Computer
BRNN	Kids & Baby, Beauty, Clothing, Computer, Automotive

From the social text of the user, we can assume that the user is more likely to be a woman who has a baby. The top three product categories: "Kids & Baby", "Clothing" and "Beauty" in gold standard illustrate that the social text is promising to reflect the purchase preference of a user. Among different prediction models, it can be observed that our deep neural network model BRNN is more matched to the user's real purchase preference.

In the second set of experiments, we further examine the impact of the amount of training data on the results of product category prediction. By fixing the test data

at 10%, we vary the remaining 90% of training data at five different splits: $\{20\%, 40\%, 60\%, 80\%, 100\%\}$. We choose BRNN to represent the deep neural network approach. The results are presented in Fig.5.

Overall, we observe that all the methods suffer from performance drop with the decrement of the training data. Nevertheless, we also see that with less training data, the traditional classification approach with shallow textual features (i.e., PCSM, TP, WEM) performs worse, while deep neural network model HRNN performs the best with relatively little training data.

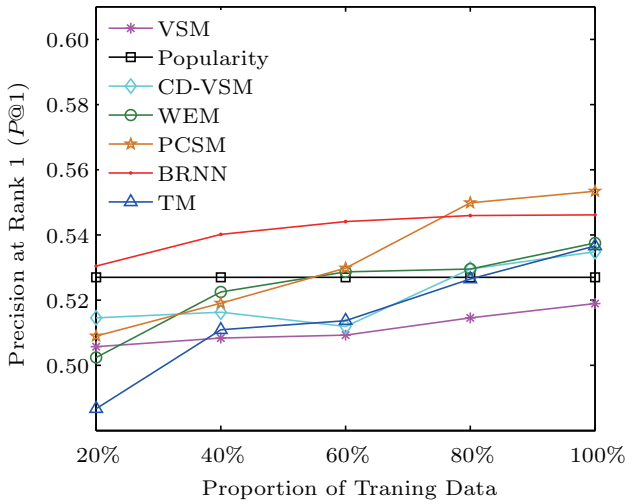


Fig.5. Varying the size of training data.

We also vary the number of dimensions for input embedding vector in deep neural network models (i.e., L), and report the results in Fig.6.

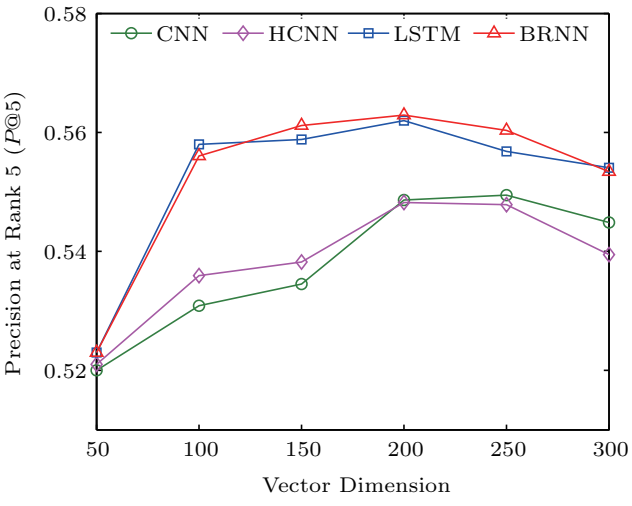


Fig.6. Varying the number of embedding dimensions.

It can be observed the number of embedding dimensions should be set to neither too large (e.g., no less than 250) nor too small (e.g., no more than 50). In our experiments, we set the embedding dimensionality L to 200.

5 Related Work

Our current work is mainly related to the following three lines of research.

Mining Social Networking and e-Commerce Websites. As the boundaries between e-commerce and social media have become increasingly blurred, some studies have investigated how social network influences users' purchase behavior. Zhang and Pennacchiotti[1,3] empirically demonstrated that users' social media profiles can be used to recommend branded product in e-commerce website. Zhao *et al.*[31] recommended products from the e-commerce website to users at the social networking website in "cold start" situations. Guo *et al.*[32] studied the trading dynamics on the e-commerce network TaoBao, and they found that users are more likely to purchase from the sellers that their friends in the network have already bought from. Bhatt *et al.*[33] studied the "peer pressure" in social media which affects the purchase behavior: if the user's friends widely adopt a product, the user is more likely to buy it. Zhou *et al.*[34] focused on product adoption probability prediction in the context of large social networks. The above researches show that social information is closely related to users' purchase behavior. Similar conclusions are also drawn in [6,35-36].

Cross-Domain Information Utilization. The key technique of cross-domain information utilization is transfer learning[37-38]. The aim is to learn transfer knowledge from the source domain, and further apply it in a target domain. Zhao *et al.*[31] used the linked users across social networking websites and e-commerce websites as a bridge to map users' social networking features to another feature representation for product recommendation. Li *et al.*[39] attempted to transfer user-item rating patterns from an auxiliary matrix in another domain to the target domain through Codebooks. Singh *et al.*[40] proposed a collective matrix factorization model to estimate the relations of multiple entities by factorizing several matrices. Zhao *et al.*[41] and Hu *et al.*[42] extended transfer learning to active learning and triadic factorization for cross-domain recommendation respectively.

Text Representation and Deep Neural Networks. Text representation is one of the fundamental problems

in text mining and information retrieval. It aims to numerically represent the unstructured text documents to make them mathematically computable. The traditional methods (e.g., vector space model and topic model) for feature representation often ignore the contextual information or word order in text and remain unsatisfactory for capturing the semantics of the words. To address the problem, Mikolov *et al.*[18-19] proposed a word embedding method to learn meaningful syntactic and semantic regularities which help to capture the semantic representation of text. Recently, deep neural network models have been widely used in many NLP tasks and have achieved promising results. For example, convolutional neural network models (CNNs)[12,21] have been widely adopted in sentiment analysis[9,43], semantic parsing[11], and sentence classification. As another type of deep neural network models, recurrent neural network models (RNNs) have been proved to be more effective than CNNs in many tasks[28,44-45]. Besides NLP tasks, neural network models have been applied to various applications in other fields, such as information representation in social media. Chen and Ku[46] proposed a user-topic-comment neural network, and Dong *et al.*[47] used an adaptive layer in a recursive neural network for target-dependent twitter sentiment analysis. Moreover, as attention mechanism has been successfully applied to deep neural network models, several recent studies achieve good results in text comprehension task[48-51]. The deep neural network architectures in these studies use an attention mechanism which allows them to highlight places in the document that might be relevant to answering the question. The deep neural network models with an attention mechanism can automatically focus on the words that have decisive effect on classification, semantic analysis, and other text processing tasks.

6 Conclusions

In this paper, we utilized user text representations learned from social media platforms to predict users' purchase preference on e-commerce platforms. We first quantitatively analysed the correlations between a user's social text posted on the social media platform and descriptions of products purchased on the e-commerce platform. We found that 1) products are more likely to be purchased by the users whose social text is more similar to the product descriptions; 2) users are more likely to purchase products from the category in which product descriptions are most similar to the social text of users. Furthermore, we proposed two major kinds of text representation approaches for predicting cross-site purchase preference. Our experimental results indicated that it is promising to utilize the social text for predicting purchase preference. Specially, the deep textual features learned from neural network approach can effectively represent the useful information in the social text, which shows a more powerful predictive ability when the number of categories becomes large. Our work tried to fill the semantic gap between heterogeneous online platforms and have practical values in the industry.

Currently, our major focus is to test the performance of various text representation models using the social text data. In the future, we consider incorporating the text information from the e-commerce platform to develop a more effective text semantic model, for example, using attention mechanisms in deep neural network structures to improve the prediction performance. The attention mechanism has been successfully applied to deep neural network models, and makes it possible to automatically focus on the words that have decisive effect on classification, semantic analysis, and other text processing tasks. As we aim to learn from the user's social text for predicting cross-site purchase preference, it could be useful to incorporate the attention mechanism to emphasize the words that are more relevant to the product information.

Apart from the social text information, another direction for future research is to leverage more information from social media platforms for predicting users' purchase preference on an e-commerce platform. We will extend our feature sets by including other types of features, such as following the information of users on social media platforms. We will also develop more effective models which can better characterize various kinds of available side information.

As an initiative work, our current prediction task is on the level of categories, and we will consider characterizing the purchase preference directly in the product level. In this way, we will be able to build a personalized cross-site recommender system for product recommendation, which will be the focus of our future work.

References

[1] Zhang Y, Pennacchiotti M. Recommending branded products from social media. In *Proc. the 7th ACM Conference on Recommender Systems*, Oct. 2013, pp.77-84.

[2] Ricci F, Rokach L, Shapira B. Introduction to Recommender Systems Handbook. Springer US, 2011, pp.1-35.

[3] Zhang Y, Pennacchiotti M. Predicting purchase behaviors from social media. In *Proc. the 22nd International Conference on World Wide Web*, May 2013, pp.1521-1532.

[4] Wang J, Zhao W X, He Y, Li X. Leveraging product adopter information from online reviews for product recommendation. In *Proc. the 9th International AAAI Conference on Web and Social Media*, May 2015, pp.464-472.

[5] Wang J, Cong G, Zhao W X, Li X. Mining user intents in Twitter: A semi-supervised approach to inferring intent categories for tweets. In *Proc. the 29th AAAI Conference on Artificial Intelligence*, Jan. 2015, pp.318-324.

[6] Zhao W X, Guo Y, He Y, Jiang H, Wu Y, Li X. We know what you want to buy: A demographic-based system for product recommendation on microblogs. In *Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2014, pp.1935-1944.

[7] Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014, pp.1555-1565.

[8] Cao Z, Li W, Li S, Wei F, Li Y. AttSum: Joint learning of focusing and summarization with neural attention. In *Proc. COLING*, Dec. 2016, pp.547-556.

[9] Gui L, Xu R, He Y, Lu Q, Wei Z. Intersubjectivity and sentiment: From language to knowledge. In *Proc. the 25th International Joint Conference on Artificial Intelligence*, July 2016, pp.2789-2795.

[10] Lai S, Xu L, Liu K, Zhao J. Recurrent convolutional neural networks for text classification. In *Proc. the 29th International AAAI Conference on Web and Social Media*, Jan. 2015, pp.2267-2273.

[11] Yih W, He X, Meek C. Semantic parsing for single-relation question answering. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014, pp.643-648.

[12] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014, pp.655-665.

[13] Basili R. Learning to classify text using support vector machines: Methods, theory, and algorithms by Thorsten Joachims. *Computational Linguistics*, 2003, 29(4): 655-661.

[14] Salton G, Wong A, Yang C S. A vector space model for automatic indexing. *Commun. ACM*, 1975, 18(11): 613-620.

[15] Robertson S. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 2004, 60(5): 503-520.

[16] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation. In *Proc. NIPS*, Dec. 2001, pp.601-608.

[17] Seroussi Y, Bohnert F, Zukerman I. Personalised rating prediction for new users using latent factor models. In *Proc. the 22nd ACM Conference on Hypertext and Hypermedia*, June 2011, pp.47-56.

[18] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In *Proc. International Conference on Learning Representations*, May 2013.

[19] Mikolov T, Sutskever I, Chen K, Corrado G S, Dean J. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, Dec. 2013, pp.3111-3119.

[20] Le Q, Mikolov T. Distributed representations of sentences and documents. In *Proc. the 31st International Conference on Machine Learning*, June 2014, pp.1188-1196.

[21] Kim Y. Convolutional neural networks for sentence classification. In *Proc. the Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1746-1751.

[22] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks. *Science*, 2006, 313(5786): 504-507.

[23] Erhan D, Bengio Y, Courville A, Manzagol P A, Vincent P, Bengio S. Why does unsupervised pretraining help deep learning? *Journal of Machine Learning Research*, 2010, 11: 625-660.

[24] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011, 12: 2493-2537.

[25] Kalchbrenner N, Blunsom P. Recurrent convolutional neural networks for discourse compositionality. In *Proc. the Workshop on Continuous Vector Space Models and Their Compositionality*, Aug. 2013, pp.119-126.

[26] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735-1780.

[27] Gers F A, Schraudolph N N, Schmidhuber J. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 2002, 3: 115-143.

[28] Zhang S, Zheng D, Hu X, Yang M. Bidirectional long short-term memory networks for relation classification. In *Proc. the 29th Pacific Asia Conference on Language, Information and Computation*, Oct.30-Nov.1, 2015.

[29] Jarvelin K, Kekalainen J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 2002, 20(4): 422-446.

[30] Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G N. Learning to rank using gradient descent. In *Proc. the 22nd Annual International Conference on Machine Learning*, Aug. 2005, pp.89-96.

[31] Zhao W X, Li S, He Y, Chang E Y, Wen J, Li X. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(5): 1147-1159.

[32] Guo S, Wang M, Leskovec J. The role of social networks in online shopping: Information passing, price of trust, and consumer choice. In *Proc. the 12th ACM Conference on Electronic Commerce*, June 2011, pp.157-166.

[33] Bhatt R, Chaoji V, Parekh R. Predicting product adoption in large-scale social networks. In *Proc. the 19th ACM International Conference on Information and Knowledge Management*, Oct. 2010, pp.1039-1048.

[34] Zhou F, Ji Y, Jiao R J. Predicting product adoption in large social networks for demand estimation. In *Proc. ISPECE*, Sept. 2014, pp.890-899.

[35] Hill S, Provost F, Volinsky C. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 2006, 21(2): 256-276.

[36] Iyengar R, Han S, Gupta S. Do friends influence purchases in a social network? *SSRN ELectronic Journal*, 2009.

[37] Pan S J, Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010, 22(10): 1345-1359.

[38] Pan W, Xiang E W, Yang Q. Transfer learning in collaborative filtering with uncertain ratings. In *Proc. the 26th AAAI Conference on Artificial Intelligence*, July 2012, pp.662-668.

[39] Li B, Yang Q, Xue X. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *Proc. the 21st International Joint Conference on Artificial Intelligence*, July 2009, pp.2052-2057.

[40] Singh A P, Gordon G J. Relational learning via collective matrix factorization. In *Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2008, pp.650-658.

[41] Zhao L, Pan S J, Xiang E W, Zhong E, Lu Z, Yang Q. Active transfer learning for cross-system recommendation. In *Proc. the 27th AAAI Conference on Artificial Intelligence*, July 2013.

[42] Hu L, Cao J, Xu G, Cao L, Gu Z, Zhu C. Personalized recommendation via cross-domain triadic factorization. In *Proc. the 22nd International Conference on World Wide Web*, May 2013, pp.595-606.

[43] Cao Y, Xu R, Chen T. Combining convolutional neural network and support vector machine for sentiment classification. In *Proc. the 4th National Conference on Social Media Processing*, Nov. 2015, pp.144-155.

[44] Miwa M, Bansal M. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, Aug. 2016.

[45] Xu Y, Jia R, Mou L, Li G, Chen Y, Lu Y, Jin Z. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proc. the 26th International Conference on Computational Linguistics*, Dec. 2016, pp.1461-1470.

[46] Chen W F, Ku L W. UTCNN: A deep learning model of stance classification on social media text. In *Proc. the 26th International Conference on Computational Linguistics*, Dec. 2016, pp.1635-1645.

[47] Dong L, Wei F, Zhou M, Xu K. Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis. In *Proc. the 28th AAAI Conference on Artificial Intelligence*, July 2014, pp.1537-1543.

[48] Hermann K M, Kocisky T, Grefenstette E, Espeholt L, Kay W, Suleyman M, Blunsom P. Teaching machines to read and comprehend. In *Proc. NIPS*, Dec. 2015, pp.1693-1701.

[49] Kobayashi S, Tian R, Okazaki N, Inui K. Dynamic entity representation with max-pooling improves machine reading. In *Proc. the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, June 2016, pp.850-855.

[50] Chen D, Bolton J, Manning C D. A thorough examination of the CNN/daily mail reading comprehension task. In *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, Aug. 2016.

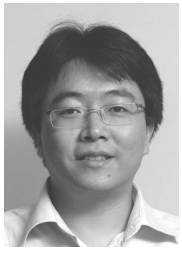
[51] Wang L, Cao Z, de Melo G, Liu Z. Relation classification via multi-level attention CNNs. In *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, Aug. 2016.

Ting Bai is currently a Ph.D. student at the School of Information, Renmin University of China, Beijing. She is working in Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing. Her research mainly focuses on social content analysis, recommender systems and deep learning, especially on commercial intent detection and human behavior analysis.

Hong-Jian Dou is currently a graduated student at the School of Information, Renmin University of China, Beijing. He is working in Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing. His research mainly focuses on natural language processing, deep learning and data mining.

Wayne Xin Zhao is currently an assistant professor at the School of Information, Renmin University of China, Beijing. He received his Ph.D. degree in computer science from Peking University, Beijing, in 2014. His research interests are web text mining and natural language processing. He has published several referred papers in top conferences including ACL, EMNLP, COLING, CIKM, SIGIR and SIGKDD.

Ding-Yi Yang is an undergraduate at the School of Information, Renmin University of China, Beijing. Her major is computer science and technology, and she is currently working in Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing.



Ji-Rong Wen is a professor at the School of Information, Renmin University of China, Beijing. He has published extensively on prestigious international conferences/journals and served as program committee members or chairs in many international conferences. He was the chair of the "WWW in China" track of the 17th World Wide Web Conference. He is currently the associate editor of ACM Transactions on Information Systems (TOIS).