Personalized Review Recommendation based on Users' Aspect Sentiment

CHUNLI HUANG, Hunan University, China
WENJUN JIANG *, Hunan University, China
JIE WU, Temple University, USA
GUOJUN WANG, Guangzhou University, China

Product reviews play an important role in guiding users' purchase decision-making in e-commerce platforms. However, it is challenging for users to find helpful reviews that meet their preferences and experiences among an overwhelming amount of reviews. Some works have been done to recommend helpful reviews to users, either from personalized or non-personalized views. While some existing models recommend similar users' reviews for a target user, they either neglect the target user's aspect preferences or the user-product interactions for measuring user similarity. Moreover, those models predict review helpfulness at the review-level (a review is taken as a whole); few of them consider the aspect-level. To address the above issues, we propose an aspect sentiment similarity-based personalized review recommendation model ($A2SPR$), which quantifies review helpfulness and recommends reviews that are customized for each individual. We analyze users' aspect preferences from reviews and improve user similarity with users' fine-grained sentiment and product relevance. Furthermore, we redefine the review helpfulness score at the aspect level, which indicates the review's reference value for users' purchase decisions. Finally, we recommend the top k helpful reviews for individuals based on the review helpfulness score. To validate the performance of the proposed model, eight baselines are developed and compared. Experimental results show that our model performs better than those baselines in both the coverage and precision.

CCS Concepts: • **Information systems** → **Information systems applications**; *Data mining*;

Additional Key Words and Phrases: product relevance, aspect level, sentiment analysis, personalized review recommendation

ACM Reference Format:
ChunLi Huang, Wenjun Jiang, Jie Wu, and Guojun Wang. 2020. Personalized Review Recommendation based on Users' Aspect Sentiment. *ACM Trans. Internet Technol.* xx, x, Article xxx (May 2020), 26 pages. https://doi.org/10.1145/1122445.1122456

*Wenjun Jiang is the corresponding author

Authors' addresses: ChunLi Huang, chunli_hnu@163.com, Hunan University, Changsha, Hunan, China, 410082; Wenjun Jiang, Hunan University, Changsha, China, jiangwenjun@hnu.edu.cn; Jie Wu, Temple University, USA; Guojun Wang, Guangzhou University, Guangzhou, China, csgjwang@gzhu.edu.cn.

1533-5399/2020/5-ARTxxx $15.00
https://doi.org/10.1145/1122445.1122456

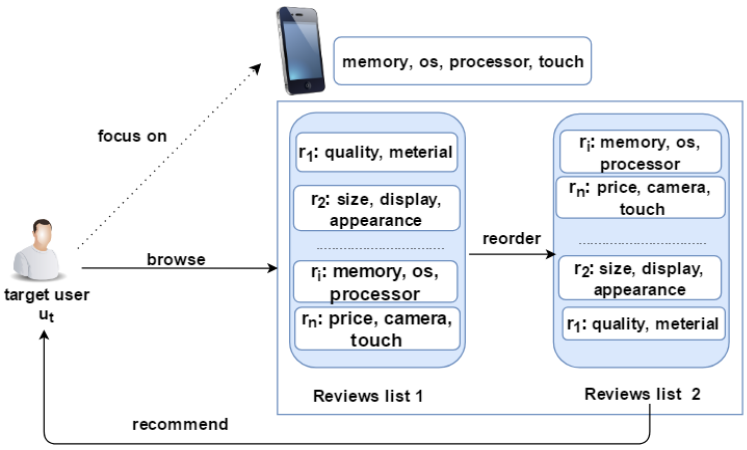


Fig. 1. The scenario of aspect-based review recommendation. The target user cares more on four aspects: "memory", "OS", "processor" and "touch" of a mobile phone. Other methods may display Review list 1 to him while our method recommends Review list 2, in which the reviews containing the most relevant aspects are ranked first.

1 INTRODUCTION

There are many reviews left by users after they make purchases on e-commerce shopping websites including *Amazon*[1], *TripAdvisor*[2], *Taobao*[3] and *JD*[4], and so on. Product reviews play an important role in guiding users' purchase decision-making in e-commerce platforms. However, it is challenging for users to find helpful reviews that meet their preferences and experiences among an overwhelming amount of reviews. Most users usually focus on a few specific product aspects, e.g., "quality", "appearance", "price", and so on. Both the reviews' qualities and users' preferences vary greatly, leading to the challenge of finding the most helpful reviews for individuals. In this paper, we strive to recommend helpful reviews with a find-grained and personalized approach at the aspect level, so as to help users understand products better and make decisions efficiently.

Existing e-commerce systems such as *Taobao*, *JD* and *Amazon* usually classify product reviews by different dimensions, e.g., "quality", "price", etc., but they recommend reviews with no differentiation to individuals. Meanwhile, most of the current researches in literature are devoted to non-personalized recommendations, e.g., [22, 23, 36, 37, 47], which neglect the users' preferences. There are only a few works on the personalized review recommendations, e.g., [1, 31, 32, 42, 44]. Online users expect to read reviews that contain aspects they are concerned about, which will save their time reading reviews and help them make purchase decisions.

We use Fig. 1 to illustrate the scenario of aspect-based review recommendation that our work tries to achieve. The target user cares more about four aspects: "memory", "OS", "processor" and "touch" of a mobile phone. Other methods may display Review list 1 which does not consider individuals' preferences, while we are trying to analyze the target user's aspect preferences and recommend reviews that contain more aspects he is interested in, as shown by Review list 2.

[1]https://www.amazon.com/
[2]https://www.tripadvisor.cn/
[3]https://www.taobao.com/
[4]https://www.jd.com/

For the personalized review recommendation, existing works usually learn latent factors from raters, reviews and products to select helpful reviews [31, 32]. The others focus on exploiting the interactions of raters, reviewers and reviews to measure a review's helpfulness and make recommendation [1, 44], or recommend reviews based on user similarity [42]. Although these efforts have achieved good results, three challenges are still open: (1) Existing works in review recommendation usually neglect the users' aspect preferences on an item. (2) When measuring user similarity, they usually consider users' overall ratings. Moreover, they neglect the fine-grained sentiment of users and the relevance of users' previously purchased products and the target product. (3) They usually measure review helpfulness in a coarse-grained manner, i.e., at the review-level (reviews are treated as a whole) rather than the aspect-level.

Our motivation. Keeping the above challenges in mind, we try to: (1) analyze users' aspect preferences from reviews and find user groups who have similar aspect preferences, (2) improve user similarity with users' fine-grained sentiment and product relevance, and (3) measure review helpfulness in a fine-grained manner, i.e., at the aspect-level.

We propose a personalized review recommendation method based on the users' aspect sentiment. First, we concentrate on capturing users' preferences on the aspects of items and measuring the aspect sentiment similarity between users. Second, we consider the relevances between the target product (i.e., the product that users want to buy) and the products that users have purchased, as an additional condition for selecting similar users accurately [4]. Finally, we measure the review's helpfulness score at the aspect-level, which is a fine-grained manner. It is worth noting that, our work is more suitable for the hot products that have many reviews, for which it is difficult to read all reviews and thus it is more urgent for review recommendation. Our main contributions are summarized as follows:

1. We formulate a personalized review recommendation problem based on users' aspect sentiment, so as to recommend the most helpful reviews to individuals. As far as we know, this is the first work that considers aspects and aspect sentiments for personalized review recommendation.

2. We propose an aspect sentiment similarity-based personalized review recommendation model ($A2SPR$). The model analyzes product relevance and users' aspect sentiment similarity, and take both of them as user-product interactions for calculating user similarity. It also redefines the reviews' helpfulness score to recommend top k helpful reviews for individuals.

3. We conduct an extensive data analysis and comparative experiments with real world review datasets. We define four new metrics and implement two groups of eight baselines, covering both non-personalized recommendations and personalized recommendations. Experimental results show that our method achieves a better aspect coverage and sentiment precision than baselines.

2 RELATED WORK

In this section, we briefly review the related work from two aspects of review analysis and review recommendation. Product reviews provide a huge amount of information for both sellers and buyers. For sellers, these reviews can effectively help them improve their products and services, so as to generate more purchases and gain more profits. For buyers, product reviews can help them make better purchase decisions. All of them need review analysis. Review ranking and recommendation also rely on review analysis.

2.1 Review analysis

The review selection, review summarization, review quality evaluation, review helpfulness prediction and review spam detection, etc., are different components of review analysis.

Review selection. The goal is to select a small set of reviews that can represent the total comments. Tsaparas et al. [46] present a subset of reviews that cover many different item aspects. However, the proportion of opinions in the corpus was not considered. Lappas et al. [23] select a subset of reviews which not only cover more product aspects but have a consistent sentiment distribution with an entire review set. The work done by Muhmmad et al. [2] is a personalized review selection.

Review summarization. The main task of review summarization is to extract and classify reviews with different semantic expressions on different product aspects, so as to generate a summary review. Nguyen et al. [34] use snippets of the entire review to form a new review. Guy et al. [12] extract tips from user-generated comments and rank them, which is similar to the review summary. Ding et al. [8] propose an abstractive method to integrate two attention mechanisms with the Encoder-decoder framework, so as to generate the review summary for supervised scenarios. Jiang et al. [14] further propose an integrated review summary generation framework for both supervised and unsupervised scenarios.

Review helpfulness prediction. It measures how helpful the reviews are to users' purchase decision-making. Ocampo Diaz et al. [35] show that the review helpfulness modeling and prediction is a task to learn the factors that determine the helpfulness. Tang et al. [44] combine text semantics with the interactions of reviewers and raters to predict review helpfulness. Saumya et al. [40] use a two-layered convolutional neural network model to predict the best helpful product reviews. Moreover, Lu et al. [26] integrate the feature of the reviewers and the social network to improve the review quality prediction.

Review spam detection. It mainly detects the review's credibility and whether the reviews may mislead consumers. This work has two subtasks: detecting review text's credibility and detecting whether the reviewer is a spammer. Jindal et al. [21] divide spam reviews into three types: untruthful opinions, reviews on brands only, and non-reviews. Lim et al. [25] define four different spam behavior patterns to detect spammers based on the reviews and ratings. Shehnepoor et al. [41] utilize the spam features to transform the spam detection into a classification problem. Minnich et al. [30] develop a systematic approach to evaluate reviews extracted from multiple review sites and examine the credibility of cross-domain reviews. Li et al. [24] discover reviewers' posting time dynamics and behavioral rules, and proposed two models to detect individual spammer and spammer groups. Jiang et al. [15, 16] propose a flow-based trust model to evaluate personalized trust and discuss graph-based trust evaluation models for enhancing user experience.

Review recommendation also involves selecting representative reviews, detecting reviews credibility, and making proper recommendation. Details are as follows.

2.2 Review recommendation

The two terms "review recommendation" and "review ranking" will be used interchangeably in this paper. Review recommendation can be seen as a separate task or as an extension task of review analysis. Its goal is to recommend reviews that are helpful to users' purchase decision-making. Momeni et al. [33] investigate the existing methods for ranking user-generated content. Krestel et al. [22] propose several review ranking strategies according to the users' needs, e.g., paying attention to specific review topics or different aspects emotions.

Wang et al. [47] consider the time dynamics of online reviews and design a time-aware review consistency ranking model.

Most of the review recommendation researches are oriented towards buyers, to help buyers better understand the item's characteristics and make correct purchase decisions. While Prado et al. [37] are oriented towards sellers and help sellers better grasp the customer's satisfaction with the items. Review recommendations usually require relevant subtasks of review analysis as a basis (e.g., review quality evaluation, review helpfulness prediction, review selection, etc.). There are two approaches: non-personalized recommendation and personalized recommendation. The former makes the same recommendation for all users, while the latter makes different recommendations for different users.

2.2.1 Non-personalized review recommendation. Non-personalized review recommendation does not take users' preferences into account and the recommendation result is the same review list shown to the public. It mainly makes recommendations by the review quality evaluation and helpfulness prediction.

Some works produce non-personalized recommendations based on review quality evaluation. Lappas et al. [23] utilize an iterative algorithm to select a subset of reviews that can represent all reviews. Paul et al. [36] recommend representative reviews with higher quality to the public, which improves the work in [23].

Some works produce non-personalized recommendations based on review helpfulness prediction. Yang et al. [51] predict the review helpfulness by examining the structural features and semantic features in review texts. Yang et al. [50] increase reviews' aspect features to predict helpfulness based on the work [51]. It confirms that reviews including product function, usability, and quality could achieve higher helpfulness scores.

2.2.2 Personalized review recommendation. Personalized review recommendation integrates individual preferences on the basis of non-personalized review recommendation [22, 23, 36, 37]. The recommended results vary with individuals, because the review quality and helpfulness are different for different individuals.

Some works generate personalized recommendations based on personalized review quality evaluation. Moghaddam et al. [31] analyze the review as a whole and apply the latent factor model to make personalized review quality prediction. Some other works generate personalized recommendations based on personalized review helpfulness prediction. Moghaddam et al. [32] analyze review helpfulness at the review-level. Tang et al. [44] analyze four different social contexts and combine them with review texts to predict review helpfulness in a personalized way. However, both of them neglect the users' fine-grained aspect preferences for products, and they predict review helpfulness in a coarse-grained way.

There are also some review recommendation methods based on user similarity. It is quite different from the above two manners. However, it is a common method in personalized item recommendation, in which finding user groups with high similarity is the critical task. Suresh et al. [42] utilize users' ratings to find similar users, then they recommend similar users' reviews. The users' aspect preferences and user-product interactions are neglected when measuring the users' similarity.

As we know, individuals have different preferences. The non-personalized review recommendations neglect users' preferences and recommend all users with the same review list. Maroun et al. [27] think that understanding the user's preferences and recommending reviews in a personalized way will make the recommendation more accurate. There are some researches dedicated to personalized review recommendations. But they pay less attention to users' aspect-level preferences and user-product interactions, which is the major focus of our

Table 1. The notations.

Notation	Explanation
$\mathcal{U}$	a set of users
$\mathcal{P}$	a set of products (items)
R^p	a set of reviews about product p
R_u	a set of reviews written by user u
R_u^p	the user u's review about product p
$\mathcal{P}_{u_t}$	product set that user u_t bought
$\mathcal{P}_j^{rel}$	related-product set of product p_j
$\mathcal{U}_i^{sim}$	similar user set of u_i
$\mathcal{A}^p$	the standard aspect set of product p
$\mathcal{A}_u^p$	the product p's aspects in u's review about p
$\mathcal{E}_u^a$	u's sentiment score about aspect a
$\mathcal{A}_u^{p^+}$	aspect set with positive sentiment in u's review about p
$\mathcal{A}_u^{p^-}$	aspect set with negative sentiment in u's review about p
$\mathcal{A}_{U_i^{sim}}^p$	aspect set contained in u_i's similar users' reviews about p
$\mathcal{A}_{U_i^{sim}}^{p^+}$	aspect set with positive sentiment in u_i's similar users' reviews about p
$\mathcal{A}_{U_i^{sim}}^{p^-}$	aspect set with negative sentiment in u_i's similar users' reviews about p

work. In this paper, we analyze reviews at the aspect-level to find users' aspect preferences and we consider user-product interactions to model user similarity.

3 PROBLEM DEFINITION

In this section, we define the problem we solve in this paper. Notations used in this paper are listed in Table 1.

3.1 System settings and Basic concepts

We first describe the system settings and basic concepts. A review system consists of three different types of entities: a set $\mathcal{P}=\{p_1, p_2,\ldots, p_m\}$ of m products; a set $\mathcal{U}=\{u_1, u_2,\ldots, u_n\}$ of n users that register on e-commerce websites; a set $\mathcal{R}=\{R^1, R^2,\ldots, R^m\}$ of reviews over the m products. The target user $u_t \in \mathcal{U}$ represents the user who wants to buy the product. The target product $p_t \in \mathcal{P}$ is the product that u_t wants to buy. Formally, we have the following definitions.

DEFINITION 1 (**Review network**). We define a review network $\mathcal{G}=\{\mathcal{P}, \mathcal{U}, \mathcal{R}\}$, in which users $\mathcal{U}$ connect with the products $\mathcal{P}$ by reviews $\mathcal{R}$. The set of reviews $\mathcal{R}$ have another form, i.e., $\mathcal{R}=\{R_1, R_2,\ldots, R_n\}$ represents reviews written by n users. u_t's similar user group is represented by $\mathcal{U}_t^{sim}$. p_t's related product set are represented by $\mathcal{P}_t^{rel}$. Product set $\mathcal{P}_{u_t}$ refers to the products reviewed by u_t. R^p is the reviews on product p and R_u represents the reviews written by user u. R_u^p denotes u's review on p.

DEFINITION 2 (**Product aspects (or attributes)**). Product aspects (or attributes) represent some characteristics that define a particular product and will affect a consumer's purchase decision.

Product aspects (or attributes) have been widely studied in literature, e.g., [5, 20, 36, 37, 42, 49]. To make it simple, we consistently use product aspects or aspects in the following parts. Here we use an example to illustrate the concept: in a review about a phone "I don't like the appearance of it", the "appearance" can be taken as a product aspect. In this paper, we use $\mathcal{A}^p$ to denote the standard set of aspects about p, which is defined in the aspect dictionary. Moreover, we use $\mathcal{A}_u^p=\{a_1, a_2,\ldots, a_s\}$ to represent product p's aspects mentioned in user u's review.

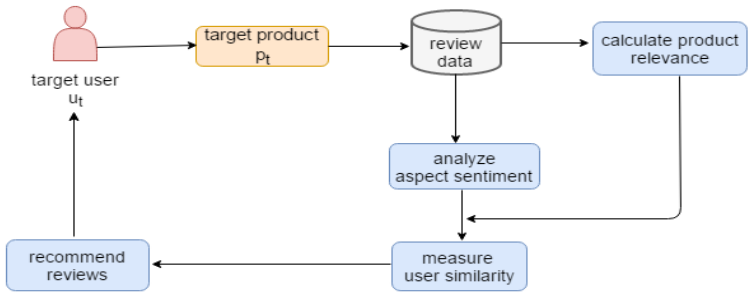


Fig. 2. Overview of the proposed A2SPR model.

DEFINITION 3 (**Aspect sentiment**). In the reviews, a user's opinion on product aspects can be seen as his aspect sentiments. For example, in a review about a phone "I don't like the appearance of it.", "don't like" is the aspect sentiment on "appearance". We use $\mathcal{E}_u^a$ to denote the user u's sentiment about aspect a, which can be quantified as a numerical value.

3.2 Problem Statement

Given the input data $(u_t, \mathcal{G}, p_t)$, our task is to recommend u_t top k helpful reviews that meet his aspect preferences and experiences. The recommendation model $\mathcal{F}$ is defined as a mapping from $(u_t, \mathcal{G}, p_t)$ to a review set R, i.e., $\mathcal{F} : (u_t, \mathcal{G}, p_t) \longrightarrow R$, where $R \subseteq R^{p_t}$.

In order to recommend proper reviews to u_t, we need to: (1) extract aspects from reviews and analyze his aspect preferences, (2) find similar users who have fine-grained preferences with him, (3) calculate reviews' helpfulness score according to user similarity, and (4) recommend top k reviews based on reviews' helpfulness score, considering u_t's aspect preferences.

3.3 Solution Overview

Our main idea is to find a user group $\mathcal{U}_t^{sim}$ who have similar aspect preferences and experiences with u_t on $\mathcal{P}_t^{rel}$. Then, we redefine the review helpfulness score and recommend the most helpful reviews to the target user u_t, according to the review's helpfulness score. Fig. 2 shows the framework of our proposed $A2SPR$ model. It has four steps: (1) calculating relevance between products, (2) analyzing aspect-sentiment in reviews, (3) calculating user similarity and (4) recommending top k helpful reviews. The details are as follows:

1. For the product set $\mathcal{P}_{u_t}$ that u_t has purchased, we construct a product-associated graph to calculate the relevance between p_t and products in $\mathcal{P}_{u_t}$, and obtain the p_t's related products $\mathcal{P}_t^{rel}$.

2. We analyze the aspect sentiments of the common reviewers who reviewed at least one product in $\mathcal{P}_t^{rel}$ and also reviewed p_t, and calculate the aspect sentiment similarity between the target user and each of the common reviewers.

3. We explore the user-product interactions to measure user similarity, in which the product relevance serves as the confidence weight of user aspect sentiment similarity.

4. We measure reviews' helpfulness at the fine-grained aspect level and recommend u_t the most helpful reviews based on the review helpfulness score.

3.4 Preliminary: aspect extraction and sentiment analysis

In this section, we briefly introduce the preliminary of our work, i.e., aspect extraction (i.e., keyword extraction) and sentiment analysis.

For the aspect extraction, the common methods are TF-IDF [39] based on word frequency and Text-Rank [29] which is a graph-based method. Previously, Bing et al. [5] develop an unsupervised method to extract the product aspects. Paul et al. [36] extract the aspects and aspect sentiments by using the double propagation method [38]. Wang et al. [48] assume that each aspect is described by only a few keywords (i.e., some similar words) and design an algorithm to obtain more related words for each aspect. It is a big challenge to extract the aspects in an unsupervised manner. If we use the supervised methods, lots of data needs to be manually tagged, which is particularly time consuming and labor intensive. Moreover, the aspects extracted by most methods need to merge the similar aspects (i.e., merging aspect synonymy). The aspect extraction involves natural language processing. According to our survey on keyword extraction, keyword extraction by machine cannot reach 100% accuracy. In this paper, in order to extract aspects as comprehensively as possible, we conduct aspect extraction based on the identified standard set of aspects, which is similar to [36].

Sentiment analysis is also known as opinion extraction or opinion mining. There are mainly two approaches in literature: the dictionary-based methods and the machine learning methods. (1) The dictionary-based methods (e.g., [43, 45]) extract opinion words from the text and calculate the opinion tendency according to the sentiment dictionary. The effect depends on the consummation of sentiment lexicon. $HowNet$[5] is commonly used as a sentiment dictionary for analyzing Chinese texts and $SentiWordNet$ [3] is commonly used to analyze English texts. (2) The machine learning-based methods (e.g., [6, 9]) select sentiment words as feature words using Logistic Regression, etc., for classification. The effect depends on the selection of training texts and the correct sentiment annotations. Most of the machine learning methods require manual annotation data. Moreover, the judgement of sentiment requires rich professional knowledge. Therefore, the most common sentiment analysis is based on sentiment dictionary, which we will use in our model.

4 DATA ANALYSIS

In this section, we first introduce the review dataset we use in this paper and the preprocessing. Then, we analyze the aspect distributions at three levels with respect to the review, user, and item, respectively. Finally, we check the existence of common reviewers, which is a key factor for measuring user similarity.

4.1 Dataset and pre-processing

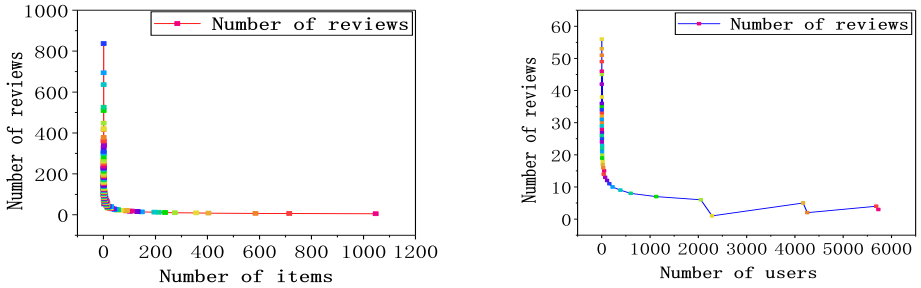
In this paper, we use a publicly available review dataset about "cell phone and accessories" and its meta-data set in *Amazon*, which is provided by McAuley et al. [28] and He et al. [13]. Note that all the duplicate item reviews are removed. A record in the data set includes product ID, user ID, review text, overall rating, review time, and review helpfulness (the ratio of up-votes to the total votes). The meta-data set contains the product ID, product category, product description and co-purchasing links. We combine the product review dataset and its meta-data set by product ID to form a new dataset. The review data about "screen protector", "basic case", "data cable", "travel charger" and "phone charm" are selected for our data analysis and experiments.

Table 2 shows the statistical details of the dataset. It lists the product categories, the number of products, users and reviews, and the standard aspect words defined in our aspect dictionary. Fig. 3(a) and Fig. 3(b) show the number of reviews for items and users respectively. It can be seen from Fig. 3(a) that a small number of items have a lot of reviews and most

[5]http://www.keenage.com/

Table 2. Statistical details of the dataset about "cell phone and accessories".

Category	Products	Users	Reviews	Standard aspect words
"screen protector"	865	11,197	16,364	"protector", "material", "quality", " fit", "price", "feature", "look", "size", "thin", "design", "brand", "protection", "touch", "put", "delivery", "service"
"data cable"	415	5915	7627	"cable", "material", "weight", "compatibility", "quality", "price", " size", "length", "brand", "experience", "delivery", "package", "service"
"phone charm"	65	809	1108	"charm", "material", "quality", "price", "feature", "size", "weight", "look", "color", "style", "brand", "cleaning", "shape", "delivery", "package", "service"
"travel charger"	364	6440	8568	"charger", "material", "quality", "price", "safety", "feature", "appearance", "size", "weight", "color", "brand", "compatibility", "protection", "heat", "delivery", "service"
"basic case"	4382	25,273	76,822	"case", "material", "quality", "price", "feature", "look", "size", "weight", "color", "style", "brand","protection", "feel", "fit", "delivery", "service"



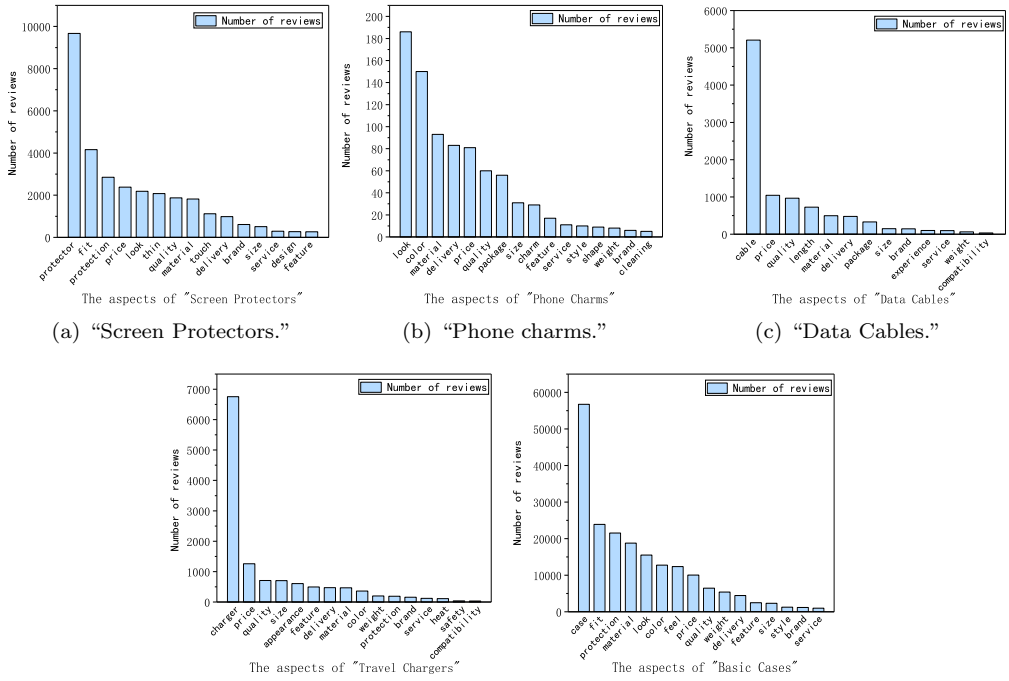
(a) Number of reviews for items (b) Number of reviews for users

Fig. 3. The number of reviews of users and items.

items have only a few reviews. For users' reviews, Fig. 3(b) shows that a few users write lots of product reviews while most users have very few reviews.

We conduct two main steps for preprocessing: one is to train the review texts using word2vec[6] and the other is to extract the aspect words and phrases contained in reviews. After training by word2vec, each word in the reviews is mapped to a d-dimensional vector. For extracting the product aspects involved in the reviews, we create a standard aspect word dictionary for each product category, which is inspired by [36]. We first segment the reviews, then we calculate the semantic similarity between the keyword in the dictionary and each of the words in the reviews. When their similarity is maximum and is greater than a threshold (e.g., 0.55), we take the word as an aspect. The details of extracting aspects will

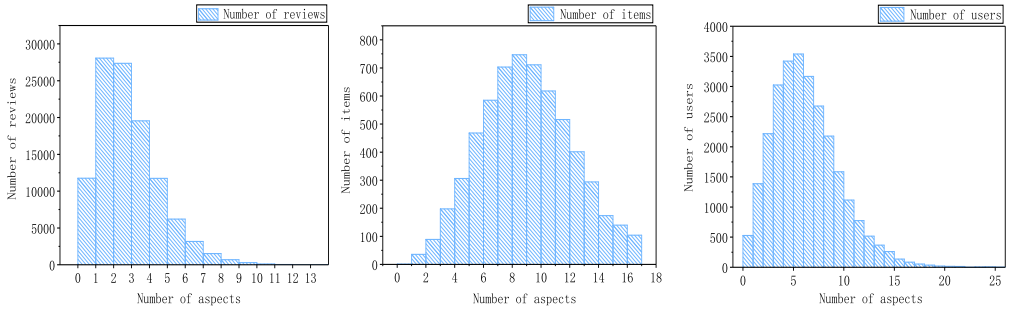
[6]https://code.google.com/archive/p/word2vec



(a) "Screen Protectors." (b) "Phone charms." (c) "Data Cables."

(d) "Travel Chargers." (e) "Basic Cases."

Fig. 4. The aspects distribution in different categories of product reviews.



(a) review level (b) item level (c) user level

Fig. 5. Statistics for the distribution of aspects from three levels.

be introduced in Section 5.2. For the phrase extraction, we take the aspect as a center word, and then we extract 5 words before and after the center word as a phrase.

4.2 Data statistics

4.2.1 Statistics for aspect distribution. We make distribution statistics for aspect words in each category of product reviews, as is shown in Fig. 4. From Fig. 4, we can see that the "price", "quality", "material" are contained in different categories of product reviews. It indicates that the general aspects of the products are more often mentioned, e.g., "protector", "cable", "charger" and "case". In addition, the aspects that represent the common features of

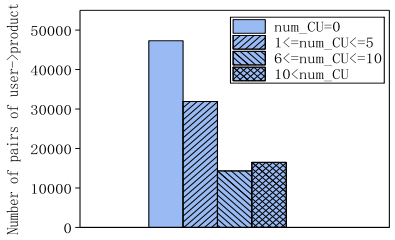


Fig. 6. Statistics for the number of common user (num_CU) between the target product p_t and its related-products.

products are also of interest to most users, e.g., "price", "quality", "material", "look" and "protection". Meanwhile, some specific product aspects are rarely mentioned, e.g., "service".

We also analyze the distribution of aspect words from three levels: the review level, the item level and the user level. The review level counts the number of aspects contained in a review. The item level counts the number of aspects contained in the reviews of an item. The user level counts the number of aspects contained in the reviews of a user. The distribution of aspect words is shown in Fig. 5. From the statistics, we find that: at the review level, more than 80% reviews contain less than 4 aspects; at the item level, 50% of items' reviews contain 5-11 aspects; at the user level, 70% of users' reviews contain 1-9 aspects.

Through the analysis of product aspects and how frequently they appear in the reviews, we find that: (1) the aspects contained in a review are quite sparse; (2) users have different concerns about a product, so they mention different product aspects in their reviews; (3) many users focus on the general aspects of the products, and a small number of users concentrate on the special aspects. Based on the above observations, we conclude that different users have different aspect preferences for the products. Therefore, in our review recommendation, we will distinguish the differences of users' aspect preferences and exploit it for personalized review recommendation.

4.2.2 Statistics for common users between p_t and its related-products. We recommend reviews written by the target user u_t's similar user group. To find similar users more accurately, we look for similar users from the user groups who have purchased target product p_t and its related products. Fig. 6 shows the distribution of the number of common reviewers (num_CU) for a user and product pair. We find that among 110,000 purchase records, 15% of user-product pairs have more than 10 common reviewers, 13% of user-product pairs have 6-10 common reviewers, 29% of user-product pairs have 1-5 common reviewers, and 43% of user-product pairs have no common reviewers.

Through analysis, we find that: (1) there do exist common reviewers between target product and its related products, which allows us to analyze the user preference similarity among these common reviewers; (2) the popular products with lots of reviews usually have a large number of common reviewers between its related products. Our model exploits common reviewers for selecting similar users. Less common reviewers will lead to less similar users. Therefore, our work is more suitable for hot products which usually have many reviews.

5 A2SPR: MODEL DETAILS

In this section, we describe our $A2SPR$ model in details: (1) constructing the product-associated graph to calculate product relevance, (2) analyzing users' aspects sentiment

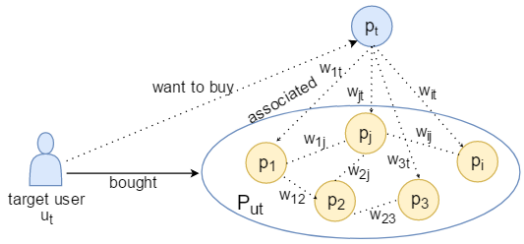


Fig. 7. The details of product associations between p_t and $\mathcal{P}_{u_t}$.

contained in their reviews, (3) calculating the similarity between users, and (4) recommending helpful reviews to users according to their personalized aspect preferences.

5.1 Calculating the relevance between products

In order to find u_t's similar user group $\mathcal{U}_t^{sim}$ more accurately, we first look for a related products set $\mathcal{P}_t^{rel}$ of the product p_t. We select the product set $\mathcal{P}_{u_t}$ that the target user u_t has reviewed, and then we calculate the relevance between p_t and p, where $p \in \mathcal{P}_{u_t}$. Finally, we get the related products set $\mathcal{P}_t^{rel}$ of p_t and their relevance. Fig. 7 illustrates the details of product relevance between p_t and $\mathcal{P}_{u_t}$.

We construct a product-associated graph G_p=$\{N, E\}$, which is similar to [19]. $N = \{p \in (\mathcal{P}_{u_t} \cup \{p_t\})\}$ are product nodes and $E = \{(p_i, p_j) \mid p_i, p_j \in (\mathcal{P}_{u_t} \cup \{p_t\}), i \neq j\}$ are the edges. There is an edge between p_i and p_j when there is at least one common reviewers between them. The number of common reviewers of the two nodes is taken as the weight of the edge. We use w_{xy} to represent the weight of edge(x, y). The more common reviewers two items have, the more relevant they are.

Besides the direct common reviewers, we also consider the indirect common reviewers. We take the same way as in [7], which is calculated with $\sum_{z \in N(x) \cap N(y)} w_{xz} + w_{zy}$. Combining the direct and indirect common reviewers, the integrated weight $W(x, y)$, which represents the number of all possible common reviewers of x and y, is calculated as follows:

$$W(x, y) = w_{xy} + \sum_{z \in N(x) \cap N(y)} w_{xz} + w_{zy}, \tag{1}$$

where x and y are two product nodes; $N(x)$ and $N(y)$ represent the neighbors of node x and y respectively; w_{xy} represents the number of direct common reviewers of node x and node y.

We use the integrated weight $W(x, y)$ to measure the relevance of two products. By normalization, we obtain the product relevance between the target product p_t and p_i, and their relevance score $R(p_t, p_i)$, as shown in Eq. (2).

$$R(p_t, p_i) = \frac{2W(p_t, p_i)}{\sum_{j=1}^{m} \sum_{k=1}^{m} w_{jk}}, \tag{2}$$

where $W(p_t, p_i)$ represents the number of all possible common reviewers (i.e., direct and indirect common reviewers) of p_t and p_i, as is shown in Eq. (1). w_{jk} represents the number of common reviewers of node j and node k, and m represents the total number of nodes in the graph. $R(p_t, p_i)$ is used to calculate the relevance between p_t and p_i. The details of calculating product relevance is shown as in Algorithm 1.

In Algorithm 1, lines 1-2 construct the product-associated graph. Lines 3 and 5 acquire the neighbors of product nodes p_t and p_i respectively. Lines 4-7 calculate the number of all possible common reviewers of p_t and p_i. Line 8 calculates the product relevance score. For

Algorithm 1 Calculating the relevance

Input: the target user u_t and the target product p_t
Output: the relevance score between p_t and other products
1: Select products set P_{u_t} that u_t has reviewed.
2: Construct the product-associated graph G_p=$\{N, E\}$: the product nodes N= $\{p \in \mathcal{P}_{u_t} \cup \{p_t\}\}$, E=$\{(p_i, p_j) \mid p_i, p_j \in \mathcal{P}_{u_t} \cup \{p_t\}, i \neq j\}$ if there exsit common reviewers between p_i and p_j. The weight on (p_i,p_j) are the number of common reviewers of two nodes.
3: Let p_t_adj be $\{p_s \mid p_s \in \mathcal{P}_{u_t}\}$, where p_t and p_s have at least one common reviewers.
4: **for** p_i in P_{u_t} **do**
5: Let p_i_adj be $\{p_j \mid p_j \in \mathcal{P}_{u_t} \cup \{p_t\}\}$, where p_i and p_j have at least one common reviewers.
6: $com\_adj \leftarrow p_t\_adj \cap p_i\_adj$.
7: Calculate $W(p_t, p_i)$ with Eq. (1).
8: Calculate $R(p_t, p_i)$ with Eq. (2).
9: **end for**
10: Return p_t's related-products and their relevance score

the product-associated graph with m product nodes, lines 3 and 5 take the time complexity of $O(m)$. The total time complexity of Algorithm 1 is $O(m^2)$.

5.2 Analyzing aspect-sentiment in reviews

In order to evaluate the users' fine-grained opinions about a product, it is necessary to utilize aspect-based sentiment analysis. There are two important steps: one is aspect extraction and the other is sentiment analysis. Details are as follows.

5.2.1 The aspects and phrases extraction. Our model relies on the aspects for review recommendation. To make the aspect extraction as comprehensive as possible, we identify a standard set of aspects by using the product catalogues from *Flipkart*[7], similar to the way in [36]. *Flipkart* is a famous e-commerce platform in India, and there are filters by standard aspects in it. In fact, the filters are quite general in many e-commerce platforms (e.g., *Amazon*, etc.), because they share the same (or similar) product categories, and each category has the same (or similar) product attributes or aspects. Based on the standard aspects, we create a product aspect dictionary and use the aspect $a \in \mathcal{A}^p$ to match and extract words in p's reviews, where $\mathcal{A}^p$ is a standard set of aspects of product p. Besides, we train the review texts using word2vec to obtain the word vector representation.

After word vector representation, we extract the aspects from reviews. We first segment the reviews and tag the words in reviews using the parts of speech (POS) tagger from NLTK[8], which is a Python package for natural language processing. The pre-trained word vectors are used to calculate the semantic similarity between a standard aspect word a and each of the words in a review, where $a \in \mathcal{A}^p$.

For each aspect a, we select the word that has the maximum semantic similarity with a as the candidate, if their semantic similarity is larger than a threshold e, we will take the word as an aspect. Note that there may be different words representing the same aspect. Therefore, we need to merge aspect synonyms. In our model, we replace the aspect word in

[7]http://www.flipkart.com/
[8]http://www.nltk.org

Table 3. Aspects and its corresponding sentiment scores in the review.

Review	Aspect	Sentiment words	Score
This phone case has super cute colors and I love that it is a soft case so it bounces when you drop it.Arrived quickly.	"color"	"super cute"	1.56
	"case"	"super cute", "soft"	1.25
	"material"	"love", "soft"	1.12

reviews with the similar aspect word in the standard aspect dictionary, so as to keep the same aspect having the same name. For the phrase extraction, we take the aspect word as a center word and extract 5 words before and after the aspect word as a phrase.

Performance Analysis. Aspect extraction is the key basis of our model. We construct the standard aspect dictionary, and we check each word in reviews to make our extraction as comprehensive as possible. Moreover, we can extract both the explicit and implicit aspect words well. For example, from the sentences "It is a soft case, and it more like a little pink", we can extract the aspects "color", "material" and "case", which have been defined in product aspect dictionary. The implicit aspects "color" and "material" are extracted, because "pink" and "soft" have higher semantic similarity with "color" and "material", respectively.

5.2.2 The aspect sentiment analysis. We conduct the sentiment analysis on the extracted aspects and phrases. Since our review data is not emotionally labeled, we choose the sentiment dictionary-based approach to quantify sentiment scores. We take $SentiWordNet3.0$ as the sentiment dictionary, which is a lexical resource for opinion mining. From the perspective of linguistics, adjectives, verb and adverbs are generally emotional in the text. Therefore, we select the adjective, verbs and adverbial words in the phrases, and use the sentiment dictionary $SentiWordNet3.0$ to measure their sentiment scores.

The sentiment scores calculated by $Sentiwordnet3.0$ can be classified into positive, neutral or negative sentiment, respectively. In our work, we map those scores to the range of [0,2]. The scores in the range of [0,1) indicate negative sentiment, while that in the range of (1,2] indicate positive sentiment and "1" is taken as the neutral sentiment. In a real product review system, most of users only mention a few aspects in their reviews, which make the aspect sentiments rather sparse. To save space, the aspect words and aspect sentiments are stored in a $json$ (JavaScript Object Notation) file with the form of "key->value".

Table 3 displays the aspects extraction and sentiment analysis about a review of a phone case. In this table, the "color", "case" and "material" are aspects which are extracted from the review. Among them, "material" is extracted, because the word "soft" have a higher semantic similarity with "material". The phrases in the review are underlined and the sentiment words contained in phrases are used to calculate the aspect sentiment scores.

5.3 Measuring the similarity between users

The users with the same opinions and preferences for the same products are considered as similar user groups. In the recommendation system, we usually recommend the item p to user A, when the item p is preferred by user B and the user B has the similar preferences with user A. In our work, we capture the users' fine-grained preference similarity and take the product relevance into account when calculating user similarity.

Fig. 8 shows the process of finding u_t's similar user group. In order to find the similar users more accurately, we first look for the common reviewers who reviewed at least one

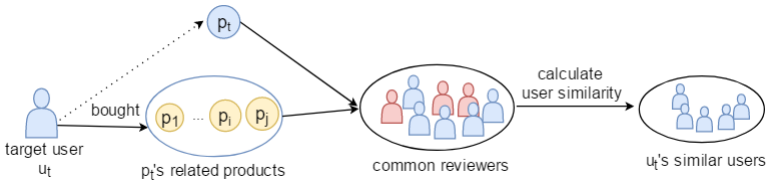


Fig. 8. The process of finding u_t's similar user group.

product in $\mathcal{P}_t^{rel}$ and also reviewed p_t. Then, u_t's similar user group will be constructed by selecting similar users from those common reviewers.

There are various ways to measure the similarity of users and items. Cosine similarity is widely used in collaborative filtering algorithms [11], and we use it to calculate users' aspect sentiment similarity. For the product p_i that u_t and u_j have reviewed together, we use the cosine similarity to analyze the aspect sentiment similarity between u_t and u_j, as shown in Eq. (3).

$$S^{p_i}(u_t, u_j) = \begin{cases} \frac{\sum_{a \in \mathcal{A}^{p_i}} \mathcal{E}^a_{u_t} . \mathcal{E}^a_{u_j}}{\sqrt{\sum_{a \in \mathcal{A}^{p_i}} \mathcal{E}^{a^2}_{u_t}} \sqrt{\sum_{a \in \mathcal{A}^{p_i}} \mathcal{E}^{a^2}_{u_j}}} & \text{if } u_t \text{ and } u_j \text{ reviewed } p_i \text{ together} \\ 0 & \text{else,} \end{cases} \tag{3}$$

where $\mathcal{A}^{p_i}$ represents the p_i's standard aspect set which is defined in aspect dictionary. $\mathcal{E}^a_{u_t}$ is u_t's aspect sentiment score about aspect a and $\mathcal{E}^a_{u_j}$ is that of u_j.

In the calculation of users' similarity, we consider the users' aspect sentiment similarity and product relevance. No matter how many reviews a user writes about the related-products, we use the product relevance scores to weight the users' aspect sentiment similarity, and then take the average as the user similarity. We use Eq. (4) to calculate the similarity between the target user u_t and u_j.

$$S(u_t, u_j) = \sum_{i=0}^{m} R(p_t, p_i) \frac{S^{p_i}(u_t, u_j)}{m}, \tag{4}$$

where $S^{p_i}(u_t, u_j)$ indicates the aspect sentiment similarity between u_t and u_j about product p_i. $R(p_t, p_i)$ represents the relevance of product p_t and p_i. m represents the number of products that u_t and u_j reviewed together.

5.4 Personalized reviews recommendation

We measure reviews' helpfulness in the fine-grained aspect level, which refers to the review's helpfulness for a user's purchase decision-making. User similarity is one of the important factors to make personalized recommendations.

If a user is more similar to the target user u_t, the user's reviews are relatively more helpful to u_t's purchase decisions. On the other hand, the more product aspects are mentioned in a review, the more u_t will know about the product, and the more valuable the review is to help u_t's purchase decision-making. Therefore, we also consider the number of product aspects contained in similar users' reviews when recommending reviews. We calculate the helpfulness score as shown in Eq. (5), and we recommend top k helpful reviews based on the helpfulness score.

$$H(u_t, u_j) = \frac{\mid \mathcal{A}^{p_t}_{u_j} \mid}{l} S(u_t, u_j), \tag{5}$$

where $H(u_t, u_j)$ represents the helpfulness of u_j's review for u_t's purchase decision-making. l is the number of aspects in aspect dictionary. $\mid \mathcal{A}^{p_t}_{u_j} \mid$ represents the number of item aspects contained in u_j's review. To avoid too much impact on recommendations, we divide $\mid \mathcal{A}^{p_t}_{u_j} \mid$

by l, so as to normalize it to the range of [0,1]. $S(u_t, u_j)$ is the similarity between user u_t and u_j, which is calculated in Eq. (4).

5.5 Case Study

In order to make our work more clear, we conduct a case study and provide some recommendation results achieved by our model. It recommends the top three helpful reviews about p_t for the target user u_t, as shown in Table 4. We also display u_t's review on p_t in Table 5.

Table 4. The recommended reviews about product "B009GSB1KU" for u_t.

similar users	recommended reviews	aspect-sentiment	aspect: sentiment score
"A13JCLM HMOBSC8" (sim=0.283)	Cute, simple case for everyday use. Bright pink color. Case goes on an off easily. Doesn't do much for protection but that's not really what I was looking for. Great if you like to change up your case often. (helpfulness score=0.07)	"case - simple" "case-easily" "color - bright pink" "protection - doesn't do much"	"case: 1.06" "color: 1.0" "protection: 0.65"
"A3R4KAK 2O0OUC6" (sim=0.327)	I love the case but its not quite magenta. It's more like a lite pink. I still like it just wish it was a brighter pink. (helpfulness score=0.041)	"case - love" "color - pink "color - brighter pink"	"case: 1.0" "color: 1.013"
"A311P2 OBNO5JWF" (sim=0.243)	I dd. not like this item. The side buttons for volume are covered and too hard to touch them. But, for $1.82 what would i expect. I will not buy from hong kong again. (helpfulness score=0.015)	"case (item) - not like"	"case: 0.375"

In the two tables, the aspects and phrases contained in the reviews are underlined. Meanwhile, the aspect-sentiment pairs (i.e., aspect words and those adjectives, verbs and adverbs describing aspects) are listed. Among them, those aspects are extracted by utilizing standard aspect words defined in the aspect dictionary, which has been described in Section 5.2, and the sentiment words are selected from the extracted phrases that are related to those aspects. The aspect sentiment scores are calculated by the tool of $SentiWordNet3.0$ (i.e., numerical values representing sentiments). Moreover, the user similarities between u_t and his similar users, and review helpfulness scores are displayed in Table 4.

Next, we illustrate the calculation of user similarity by an example. For the product "B0093QER4C" which was reviewed together by u_t and user "A13JCLMHMOBSC8", the product relevance between "B0093QER4C" and p_t calculated by our model is 0.327. After normalization, the aspect sentiment similarity of u_t and "A13JCLMHMOBSC8" is 0.865. Therefore, the user similarity between u_t and "A13JCLMHMOBSC8" is calculated as $(0.327 * 0.865)/1 = 0.283$, where 1 is the number of related products that the two users reviewed together. The review helpfulness score of "A13JCLMHMOBSC8" is calculated as $0.283 * (4/16) = 0.07$, where 4 is the number of aspects on review of "A13JCLMHMOBSC8" and 16 is the length of standard aspect dictionary about "basic case".

The target product p_t "B009GSB1KU" is a phone case. The "material", "case", "color", etc., are its aspects. From Table 5, we know that u_t mentions the "color", "case" and "material" about p_t, and he has positive sentiment on them (i.e., the aspect sentiment scores are greater than 1). We believe that the mentioned aspects reveal u_t's aspect preferences, and the aspect sentiments represent u_t's experiences about p_t. If the recommended reviews satisfy u_t's aspect preferences and experiences, they should contain the aspects that u_t

Table 5. The u_t's review about product "B009GSB1KU".

u_t	review	aspect-sentiment	aspect: sentiment score
"A101577718CA TXNFEYBQR"	This phone case has super cute colors and I love that it is a soft case so it bounces when you drop it.Arrived quickly.	"case - super cute" "color - super cute" "case -soft" "material - love, soft"	"case: 1.25 " "color: 1.56" "material: 1.12"

mentioned, and they should have the similar sentiment on these aspects. As we can see in Table 4, the reviews recommended by our model contain the aspects that u_t is interested in, e.g., "case" and "color", and some of them have similar aspect sentiments with u_t regarding p_t (i.e., the sentiment on the "case" and "color" are positive). Therefore, we can say that the reviews recommended by our model can better satisfy users' preferences and experiences.

It is worth noting that in the recommended reviews of Table 4, user "A13JCLMHMOB SC8" thought that the target product doesn't do much for protection. Meanwhile, the aspect sentiment score on "protection" calculated by $SentiWordNet3.0$ is 0.65 (less than 1), which indicates a negative sentiment. In real life, "protection" could be an important aspect of the target product. However, since we recommend helpful reviews to the target user in a personalized way by considering his aspect preferences, if the target user is not interested in certain aspects, our model will not take those aspects into consideration. Table 5 displays the real product review posted by the target user, in which he didn't mention the protection of the target product. Thus, the "protection" aspect will not be considered when recommending reviews to the target user. On the other hand, the more aspects contained in the recommended reviews, the greater the reference value is for those reviews. Therefore, "protection" contained in our recommended reviews will make the target user understand the target product better and it will not affect the performance of our model.

6 EXPERIMENTS

In this section, we conduct experiments on the review dataset about "cell phone and accessories" on *Amazon* to verify the performance of our proposed model.

6.1 Evaluation method and metrics

We use the leave-one-out method to test the performance of our work. In our recommendation, we first mask u_t's review on p_t; then we use u_t's reviews on other products to find similar users who have similar aspect preferences with him, and recommend the top k helpful reviews on p_t written by u_t's similar users. The aspects and aspect sentiments contained in u_t's review about p_t can reveal u_t's aspect preferences and experiences about p_t. In order to verify whether the reviews we recommend to the target user u_t satisfy his preferences and experiences, we take u_t's review on p_t as the ground truth. Then, we compare the recommended reviews with the masked review.

We define four metrics to evaluate the performance, i.e., the aspect coverage, the sentiment precision, the aspect coverage error and the global error. The details are as follows.

6.1.1 The Aspect Coverage. The aspect coverage refers to the proportion of aspects involved in both the recommended reviews and the ground truth to all aspects contained in u_t's review. This metric represents the degree to which the recommended reviews cover u_t's

aspect preferences. The aspect coverage is calculated as follows:

$$coverage = \frac{\mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid}{\mid \mathcal{A}_{u_t}^{p_t} \mid} \tag{6}$$

In Eq. (6), $\mathcal{A}_{u_t}^{p_t}$ represents the aspects contained in u_t's review about the target product p_t, and $\mathcal{A}_{U_t^{sim}}^{p_t}$ represents the aspects contained in recommended reviews about p_t. $\mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t}$ represents the same aspects involved in both recommended reviews and u_t's review about product p_t. For example, for the scenarios in Table 4 and Table 5, $\mathcal{A}_{u_t}^{p_t}$= {"case", "color", "material"}, $\mathcal{A}_{U_t^{sim}}^{p_t}$= {"case", "color", "protection"}, $\mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t}$= {"case", "color"}, so the coverage is 0.667.

6.1.2 The Sentiment Precision. The sentiment precision refers to the proportion of aspects with consistent sentiments in the same aspects which are involved in both the recommended reviews and u_t's review. In this metric, the consistent aspect sentiments indicate that the aspects have the same sentiment polarity. This metric tests whether the recommended reviews satisfy u_t's experiences, i.e., whether the recommended reviews have similar aspect sentiments with u_t's review. The sentiment precision is calculated as follows:

$$precision = \frac{\mid \mathcal{A}_{u_t}^{p_t^+} \cap \mathcal{A}_{U_t^{sim}}^{p_t^+} \mid + \mid \mathcal{A}_{u_t}^{p_t^-} \cap \mathcal{A}_{U_t^{sim}}^{p_t^-} \mid}{\mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid} \tag{7}$$

In Eq. (7), $\mathcal{A}_{u_t}^{p_t^+}$ and $\mathcal{A}_{u_t}^{p_t^-}$ represent the positive and negative aspect opinions contained in u_t's review about p_t, respectively. Meanings of $\mathcal{A}_{U_t^{sim}}^{p_t^+}$ and $\mathcal{A}_{U_t^{sim}}^{p_t^-}$ are similar to that of $\mathcal{A}_{u_t}^{p_t^+}$ and $\mathcal{A}_{u_t}^{p_t^-}$, except that they are for similar users of u_t. It is worth noting that, there may be multiple reviews that cover the same aspect. In this case, we will calculate their average sentiment score to represent the sentiment of all recommended reviews on this aspect. We will use the scenario of case study to illustrate how to calculate sentiment precision. For the review of the target user on the target product, in Table 5, $\mathcal{A}_{u_t}^{p_t^+}$ = {"case", "color", "material"}. For the top-3 recommended reviews, in Table 4, the aspect "case" has been mentioned in all the three reviews, so we calculate the average of three aspect sentiment scores of "case". That is, (1.06+0.375+1.0)/3 =0.812, which is lower than 1, indicating negative sentiment. Then we have $\mathcal{A}_{U_t^{sim}}^{p_t^+}$= {"color"}, $\mathcal{A}_{U_t^{sim}}^{p_t^-}$= {"case", "protection"}. Finally, the precision is calculated as (1+0)/2 = 0.5.

6.1.3 Aspect Coverage error. The aspect coverage error tests how many aspects contained in u_t's review are not contained in recommended reviews. We calculate the proportion of aspects that are mentioned in u_t's review but are not mentioned in recommended reviews. The aspect coverage error is calculated as in Eq. (8):

$$error1 = \frac{\mid \mathcal{A}_{u_t}^{p_t} \mid - \mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid}{\mid \mathcal{A}_{u_t}^{p_t} \mid} \tag{8}$$

For example, for the scenarios in Table 4 and Table 5, $\mathcal{A}_{u_t}^{p_t}$= {"case", "color", "material"}, and $\mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t}$= {"case", "color"}. $\mid \mathcal{A}_{u_t}^{p_t} \mid$=3 and $\mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid$=2. Therefore, the error1 is calculated as (3-2)/3 = 0.333.

6.1.4 Global error. The global error tests two points. One is the aspect coverage error (i.e., failing to cover some aspects as calculated in Eq. 8), and the other is the aspect sentiment error (i.e., cover the aspects but with wrong sentiment). The global error is calculated as in Eq. (9):

$$error2 = \frac{\mid \mathcal{A}_{u_t}^{p_t} \mid - \mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid}{\mid \mathcal{A}_{u_t}^{p_t} \mid} + \frac{\mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid - \mid \mathcal{A}_{u_t}^{p_t^+} \cap \mathcal{A}_{U_t^{sim}}^{p_t^+} \mid - \mid \mathcal{A}_{u_t}^{p_t^-} \cap \mathcal{A}_{U_t^{sim}}^{p_t^-} \mid}{\mid \mathcal{A}_{u_t}^{p_t} \cap \mathcal{A}_{U_t^{sim}}^{p_t} \mid} \tag{9}$$

For example, for the scenarios in Table 4 and Table 5, $\mathcal{A}_{u_t}^{p_t^+} \cap \mathcal{A}_{U_t^{sim}}^{p_t^+}$={"case", "color"}, $\mathcal{A}_{u_t}^{p_t^-} \cap \mathcal{A}_{U_t^{sim}}^{p_t^-} = \varnothing$. $\mid \mathcal{A}_{u_t}^{p_t^+} \cap \mathcal{A}_{U_t^{sim}}^{p_t^+} \mid$=2, $\mid \mathcal{A}_{u_t}^{p_t^-} \cap \mathcal{A}_{U_t^{sim}}^{p_t^-} \mid$=0. Therefore, the error2 is calculated as 0.333+(2-2-0)/2= 0.333.

It is worth noting that some indicators can also be defined to check whether the recommended reviews are consistent with the target user u_t's preferences. For instance, we can check which aspects are included in the recommended reviews but are not included in u_t's review. However, we believe that a review that contains more aspects is more helpful for the user to understand the product. Therefore, we do not define metrics to check those scenarios.

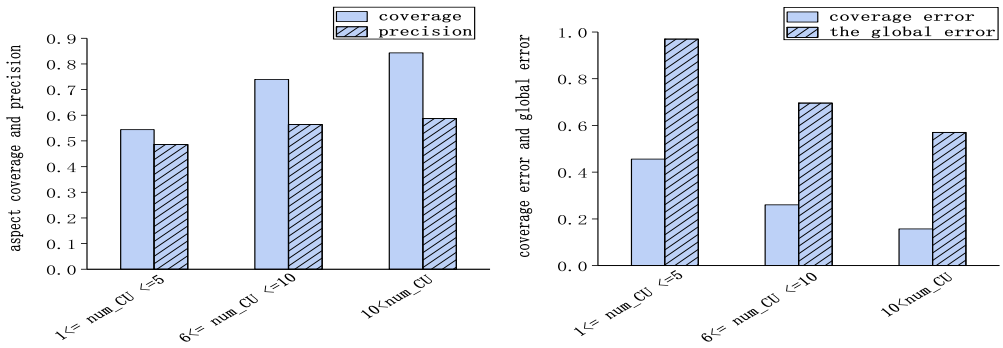
6.2 Baselines

As far as we know, there lacks personalized review recommendation based on users reviews' aspects or the aspect sentiments, so we cannot find appropriate baselines to compare with our work. To check the advantage of the proposed model, we define two sets of baselines. One set consists of non-personalized review recommendation methods; the other is composed of personalized review recommendation methods, where they focus on either reviews' aspects or reviews' sentiment, but not both.

6.2.1 Baselines for non-personalized review recommendation. The popular shopping websites, e.g., *Taobao*, *Amazon*, can display reviews according to their publish time. The newly published reviews may better reflect the dynamic characteristics of product aspects, so we take the newly posted (NR) review recommendation as one of our baselines. Moreover, reviews with more up-votes (UR) or a higher helpfulness (HR, which is the ratio of up-votes to the total votes) generally reflect the reviews' helpfulness for a user's decision-making, so they are also taken as baselines. Since our review data already processes empty text and duplicate data, random recommendations (RR) would allow for a quick review selection, so it would also be a good baseline.

Based on the above analysis, we implement the following four methods as baselines for non-personalized review recommendation, and compare them with our proposed $A2SPR$ model on the four metrics we define above.

- `Recommending reviews newly posted` *(NR)*: sorting the review list by its posted time and recommending the most recently posted reviews to users.
- `Recommending reviews with high helpfulness` *(HR)*: review helpfulness represents the ratio of the up-votes to total votes. This method recommends users the reviews with higher helpfulness.
- `Recommending reviews with more up-votes` *(UR)*: this method only takes into account the number of up-votes. It recommends users the reviews with more up-votes.
- `Recommending reviews randomly` *(RR)*: it randomly selects reviews and recommends them to users.



(a) coverage and precision (b) coverage error and global error

Fig. 9. Comparisons for the different numbers of common users (num_CU) between the target product and its related-products, when top three reviews are recommended.

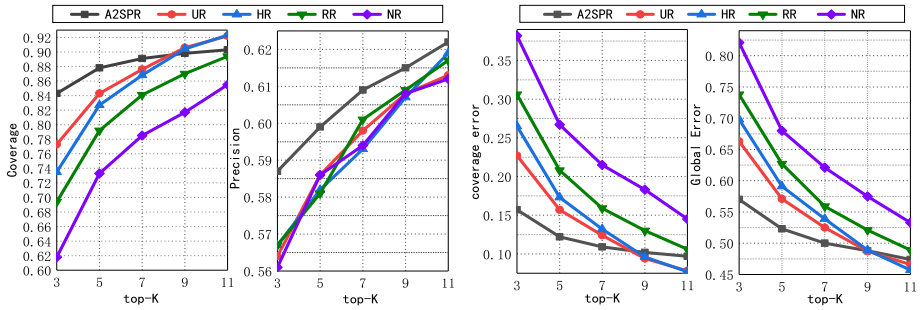
6.2.2 Baselines for personalized review recommendation. To compare the performance with personalized review recommendations, we implement four personalized methods as our baselines. They are all based on user similarity, and they may consider product relevance, reviews' aspects, or reviews' sentiment, as follows.

- `Similar to our method, but ignoring the item's relevance` (*NPRR)*: this method is similar to our model, but it does not consider the user-product interactions when calculating the user similarity.
- `Just consider the common aspect mentioned in reviews` (*APR)*: this method ignores users' aspects sentiment. It assumes that the more common aspects users comment on, the higher similarity they share.
- `Just consider users' sentiment, not involving aspects` (*SPR)*: it judges the similarity between users by considering the closeness of users' rating. For an item, if the two users rate closer, they would have more similarity.
- `Consider neither the aspects nor the sentiment` (*PR)*: in this method, the users who purchased the related-products are in their own similar user groups. The more related-products users have purchased, the higher the user similarity they have.

6.3 Experimental results

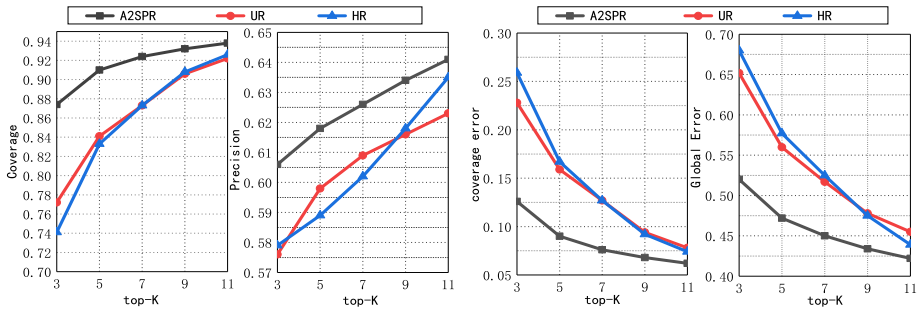
6.3.1 Effects of parameter (num_CU). In $A2SPR$, we make personalized review recommendations based on user similarity. In order to find similar users more accurately, we look for u_t's similar users in the user groups who have purchased target product p_t and its related products. If very few products are related to the target product p_t, or few common users have purchased the target product p_t and its related products, the number of similar users will be very small. The accuracy of user similarity may be impacted. The more common reviewers between the target product p_t and its related products $\mathcal{P}_t^{rel}$, the larger the number of similar users is. We remove the cold start data for testing, e.g., the target product p_t has no related products, or p_t and its related products have no common reviewers.

We conduct experiments to test the effects of different number of common reviewers (num_CU) between p_t and $\mathcal{P}_t^{rel}$. Fig. 9 shows the results, when top 3 reviews (i.e., reviews with top 3 helpfulness scores from similar users) are recommended. We gain the following main findings: (1) When there are more common reviewers between the target product p_t and $\mathcal{P}_t^{rel}$, our model will achieve a better performance, i.e., a higher aspect coverage and sentiment precision, and a lower coverage error and global error. (2) When num_CU >10,



(a) coverage and precision (b) coverage error and global error

Fig. 10. The comparisons among the four non-personalized recommendation methods and A2SPR when using data of $num\_CU \geq 10$.



(a) coverage and precision (b) coverage error and global error

Fig. 11. The comparisons among A2SPR, UR and HR when using data of $num\_CU \geq 20$.

our model has the highest coverage and precision; its coverage and precision are about 30% and 10% higher than that of $num\_CU \leq 5$, respectively; its coverage error and global error are about 27% and 40% lower than that of $num\_CU \leq 5$, respectively.

Therefore, for ensuring sufficient test data and a better review recommendation performance, we use the purchase data where each pair of u_t and p_t has no less than 10 common reviewers as our test data. It is worth noting that, the setting does not indicate the application limitation of our model. Because in real life, it is not quite necessary to make review recommendation unless there are large number of reviews.

6.3.2 Model comparison for non-personalized review recommendation. We take the *NR*, *HR*, *UR* and *RR* as baselines for non-personalized recommendations, which have been introduced in Section 6.2.1. They are non-personalized recommendation methods and all users are recommended the same review list. We compare our *A2SPR* model with those baselines in terms of aspect coverage, precision, coverage error and global error, when recommending top k reviews. The results are shown as in Fig. 10. Our main findings are as follows:

(1) When recommending the top 3-7 reviews, our *A2SPR* method achieves the highest coverage and precision. That is, the reviews recommended by *A2SPR* satisfy u_t's aspect preferences better and have more similar aspect sentiment with u_t. The reason is that the non-personalized recommendations do not take into account the user's fine-grained preferences. In terms of coverage, *UR* is slightly better than *HR*, and they are higher than *NR* and *RR*. *NR* has the lowest coverage. *RR* may randomly select reviews from similar users or reviews with more up-votes, which leads to a higher coverage than *NR*. The precision

achieved by those methods is lower and unstable. This may be because they do not consider users' sentiment contained in their reviews.

(2) When recommending the top 9-11 reviews, the coverage in our model is a bit lower than that of *UR* and *HR*, so it makes the errors bigger in our model. We believe that the num_CU is an important factor and we conduct experiments to verify it later. From Fig. 10(a), we can see that $A2SPR$ always achieves the best precision when recommending the top k reviews. It is because that the aspects sentiment contained in the reviews recommended by $A2SPR$ are the most similar to that of the target users.

As precision can demonstrate the consistency of aspect sentiments contained in recommended reviews and the target user's review, it is the most important indicator for evaluating the quality of recommendation. All the performance of the four non-personalized recommendation methods on Precision is lower than ours. Therefore, our model achieves a better performance.

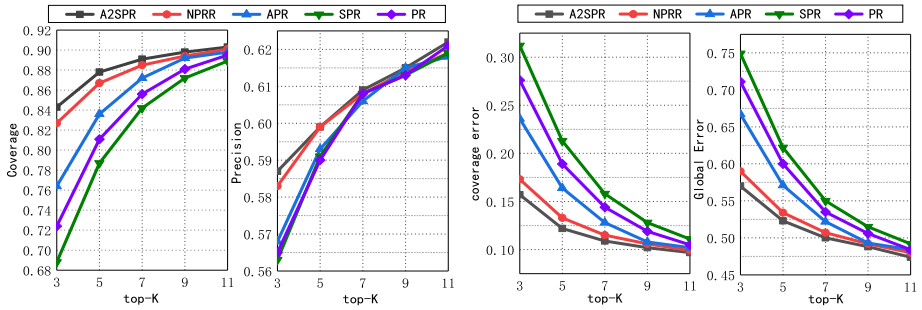
Next, we explain in detail why our model has a slightly lower coverage than *HR* and *UR*, when top 9 or 11 reviews (i.e., reviews with top 9 or 11 helpfulness scores from similar users) are recommended. The reasons may lie in the number of common reviewers between target product p_t and $\mathcal{P}_t^{rel}$, which affects the real number of recommended reviews.

In the experiments, we take the data that the number of common reviewers (num_CU) ≥ 10 for testing. We select u_t's similar users from the common reviewers of p_t and $\mathcal{P}_t^{rel}$, and recommend the top k helpful reviews from similar users to u_t. Generally, each reviewer usually post one review on a single product. Therefore, the number of reviewers is the same with that of reviews. However, there are some cases where the number of similar users is inadequate.

It can be seen from Fig. 10(a) that, when our model recommends top 9 and top 11 helpful reviews respectively, their coverage is very close to each other. Moreover, when top 9 and top 11 reviews are being recommended, our coverage is slightly lower than that of *HR* and *UR*. Based on those observations, we have the following hypotheses: (1) some target users have fewer than 9 or 11 similar users when using the test data of $num\_CU \geq 10$, because some common reviewers may have zero similarity with the target user and they will not be selected into the similar user group; (2) since each similar user has one review, it's possible that we cannot actually recommend 9 or 11 reviews due to the lack of similar users. Therefore, when our model recommends 9 and 11 reviews, the coverage cannot be improved.

To further validate our hypotheses, we take the data of $num\_CU \geq 20$ for more testing. We compare $A2SPR$ with UR and HR. The results are shown in Fig. 11. It can be observed that our $A2SPR$ model achieves significant performance on all the four metrics. As we can see from Fig. 11(a), the coverage of *HR* and *UR* has little change, while the coverage of *A2SPR* is improved significantly and reaches the highest among the three methods. It validates the importance of common reviewers. It also verifies that our model works well for review recommendation, especially for those hot products where there are a large number of reviews and common reviewers.

6.3.3 Model comparison for personalized recommendation. To compare our model performance in personalized recommendations, we take the *NPRR*, *APR*, *SPR* and *PR* as baselines, which have been introduced in Section 6.2.2. We compare our aspect sentiment similarity-based personalized recommendation model ($A2SPR$) with baselines in terms of aspect coverage, precision, coverage error and global error, when top k reviews are recommended. The results are shown in Fig. 12. Our main findings are as follows:



(a) coverage and precision (b) coverage error and global error

Fig. 12. The comparisons among the four personalized recommendation methods and A2SPR.

(1) When recommending top k reviews, *A2SPR* achieves the best performance in coverage. It demonstrates that our model is better at mining the user's aspect-level preferences. In terms of coverage, compared to our method, *NPRR* is slightly lower, which may be caused by neglecting the user-product interactions when calculating the user similarity. *APR* which only considers product aspects has a higher coverage than *SPR* and *PR*. When there are more co-purchases between the target user and others, *PR* finds similar users more accurately than *SPR*. Therefore, the coverage of *PR* is higher than *SPR*.

(2) The precision of $A2SPR$ is also the highest, because $A2SPR$ concentrates on aspect-level sentiments. It demonstrates that the reviews recommended by our model have the similar aspects sentiment with that of u_t. *NPRR* also concentrates on aspect-level sentiment, but it neglects the user-product interactions, which leads to a slightly lower precision than ours. *APR* and *PR* do not consider sentiment. *SPR* focuses on user ratings, which is a coarse-grained sentiment. Therefore, the precision in those three methods are close to each other, and they are all lower than our $A2SPR$. The improvements of $A2SPR$ over other baselines, are shown in Table 6.

In summary, when recommending top k reviews, $A2SPR$ which considers aspects, aspect sentiments, and user-product interactions, achieves the best performance. It recommends reviews that can better meet the u_t's aspect preferences and experiences.

6.4 Summary of Experiments

Our $A2SPR$ model considers users' fine-grained sentiments and preferences, and recommends reviews that are customized for each individual. The extensive experiments and analysis verify that our model performs better than all the eight baseline methods. It is worth noting that, our model works better when there are many reviews and plenty common reviewers.

7 CONCLUSION

In this paper, we present the aspect sentiment similarity-based personalized review recommendation method ($A2SPR$), to recommend the most helpful reviews to users from large number of reviews. We analyze the product relevance and users' aspect sentiment similarity, and propose a new method to measure the user similarity by considering both of them as user-product interactions. Then we redefine review helpfulness score at the aspect-level, which indicates the review's helpfulness for the target users' purchase decision-making. In recommending the top k reviews, the reviews with high helpfulness scores will be recommended. The empirical experiments validate that the proposed $A2SPR$ model is able to recommend users with reviews that meet their aspect preferences and experiences accurately.

Table 6. Improvement of $A2SPR$ over each baseline in terms of four metrics.

top-k	Non-personalized (Improvement %)					personalized (Improvement %)				
	model	coverage	precision	coverage error	global error	model	coverage	precision	coverage error	global error
k=3	NR	36.41%	4.63%	58.90%	30.57%	PR	16.44%	3.89%	43.12%	19.83%
	RR	21.47%	3.53%	48.69%	22.76%	SPR	22.53%	4.26%	49.68%	23.90%
	HR	14.69%	3.53%	40.75%	18.34%	APR	10.34%	3.35%	33.47%	14.54%
	UR	9.06%	4.08%	30.84%	14.03%	NPRR	1.93%	0.69%	9.25%	3.39%
k=5	NR	19.78%	2.22%	54.31%	23.09%	PR	8.26%	1.53%	35.45%	12.83%
	RR	10.86%	3.10%	41.35%	16.59%	SPR	11.56%	1.35%	42.72%	15.92%
	HR	6.17%	2.92%	29.48%	11.51%	APR	5.02%	1.01%	25.61%	8.41%
	UR	4.15%	2.22%	22.29%	8.41%	NPRR	1.27%	0.00%	8.27%	2.06%
k=7	NR	13.50%	2.53%	49.30%	19.48%	PR	4.09%	0.16%	24.31%	6.54%
	RR	5.95%	1.33%	31.45%	10.55%	SPR	5.82%	0.16%	31.01%	9.09%
	HR	2.65%	2.70%	17.42%	7.24%	APR	2.18%	0.50%	14.84%	4.21%
	UR	1.71%	1.84%	12.10%	4.76%	NPRR	0.68%	0.16%	5.22%	1.38%
k=9	NR	9.91%	1.15%	44.26%	15.13%	PR	1.93%	0.33%	14.29%	3.56%
	RR	3.22%	0.99%	21.54%	6.33%	SPR	2.98%	0.33%	20.31%	5.24%
	HR	-0.67%	1.32%	-6.25%	0.20%	APR	0.67%	0.00%	5.56%	1.01%
	UR	-0.89%	1.15%	-8.51%	-0.20%	NPRR	0.45%	0.16%	3.77%	0.81%
k=11	NR	5.61%	1.63%	33.10%	11.07%	PR	0.89%	0.16%	7.62%	2.07%
	RR	1.01%	0.81%	8.49%	3.07%	SPR	1.57%	0.48%	12.61%	3.66%
	HR	-2.21%	0.48%	-20.62%	-3.59%	APR	0.56%	0.65%	4.90%	2.07%
	UR	-2.10%	1.47%	-19.59%	-1.69%	NPRR	0.22%	0.48%	2.02%	1.25%

In the future, we are interested in improving our work in the following directions: (1) We could attempt to explore the word clustering to improve aspect words extraction. (2) The aspects in product reviews bear the sparsity, leading to the difficulty of capturing users' aspect preferences. We would like to exploit other methods (e.g., [10]) to improve our model. (3) The popular shopping sites are filled with many fake reviews, which will affect the user's judgment for the products. We can build a user trust framework [17, 18] and look for similar and trustworthy users. We are also interested in applying our model to more fields, e.g., course recommendation in e-learning.

8 ACKNOWLEDGMENTS

This research was supported by NSFC grant 61632009, Guangdong Provincial NSF Grant 2017A030308006, Open project of Zhejiang Lab 2019KE0AB02, and in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1618398, CNS 1651947, and CNS 1564128.

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, and Bo Pang. 2011. Personalized Recommendation of User Comments via Factor Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Association for Computational Linguistics, 571–582.

[2] Muhmmad Al-Khiza'ay, Noora Alallaq, Qusay Alanoz, Adil Al-Azzawi, and N. Maheswari. 2018. PeRView: A Framework for Personalized Review Selection Using Micro-Reviews. *CoRR* abs/1804.08234 (2018). arXiv:1804.08234

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. *Proceedings of LREC* 10.

[4] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2014. Neighbor Selection and Weighting in User-Based Collaborative Filtering: A Performance Prediction Approach. *ACM Trans. Web* 8, 2, Article 12 (March 2014), 30 pages.

[5] Lidong Bing, Tak-Lam Wong, and Wai Lam. 2016. Unsupervised Extraction of Popular Product Attributes from E-Commerce Web Sites by Considering Customer Reviews. *ACM Trans. Internet Technol.* 16, 2, Article 12 (2016), 17 pages.

[6] Rihab Bouchlaghem, Aymen Elkhelifi, and Rim Faiz. 2016. A Machine Learning Approach For Classifying Sentiments in Arabic tweets. In *Proc. WIMS*. 1–6.
[7] Marta Caro-Martinez and Guillermo Jimenez-Diaz. 2017. Similar Users or Similar Items? Comparing Similarity-Based Approaches for Recommender Systems in Online Judges. In *International Conference on Case-based Reasoning*.
[8] X. Ding, W. Jiang, and J. He. 2018. Generating Expert's Review from the Crowds': Integrating a Multi-Attention Mechanism with Encoder-Decoder Framework. In *Proc.UIC*. 954–961.
[9] Abdeljalil Elouardighi, Mohcine Maghfour, Hafdalla Hammia, and Fatima Zahra Aazi. 2017. A Machine Learning Approach for Sentiment Analysis in the Standard or Dialectal Arabic Facebook Comments. In *Proc. CloudTech*.
[10] Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. 2019. Attentive Aspect Modeling for Review-Aware Recommendation. *ACM Trans. Inf. Syst.* 37, 3, Article 28 (March 2019), 27 pages.
[11] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2016. A Novel Evidence-Based Bayesian Similarity Measure for Recommender Systems. *ACM Trans. Web* 10, 2, Article 8 (May 2016), 30 pages.
[12] Ido Guy, Avihai Mejer, Alexander Nus, and Fiana Raiber. 2017. Extracting and Ranking Travel Tips from User-Generated Reviews. In *Proc. WWW*. 987–996.
[13] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proc. WWW*. 507–517.
[14] Wenjun Jiang, Jing Chen, Xiaofei Ding, Jie Wu, Jiawei He, and Guojun Wang. 2020. Review Summary Generation in Online Systems: an Integrated Framework for Supervised and Unsupervised Scenarios. *ACM Transactions on the Web* (2020).
[15] Wenjun Jiang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Jie Wu. 2016. Understanding Graph-Based Trust Evaluation in Online Social Networks: Methodologies and Challenges. *ACM Comput. Surv.* 49, 1, Article 10 (May 2016), 35 pages.
[16] W. Jiang, J. Wu, F. Li, G. Wang, and H. Zheng. 2016. Trust Evaluation in Online Social Networks Using Generalized Flow. *IEEE Transactions on Computers (TC)* 65(3) (2016), 952–963.
[17] Wenjun Jiang, Jie Wu, and Guojun Wang. 2015. On Selecting Recommenders for Trust Evaluation in Online Social Networks. *ACM Trans. Internet Technol.* 15, 4, Article 14 (Nov. 2015), 21 pages.
[18] W. Jiang, J. Wu, G. Wang, and H. Zheng. 2016. Forming Opinions via Trusted Friends: Time-evolving Rating Prediction Using Fluid Dynamics. *IEEE Transactions on Computers (TC)* 65(4) (2016), 1211–1224.
[19] Guillermo Jimenez-Diaz, Pedro Pablo Gmez Martn, Marco Antonio Gmez Martn, and Antonio A. Snchez-Ruiz. 2016. Similarity Metrics from Social Network Analysis for Content Recommender Systems. (2016), 203–217.
[20] Zhipeng Jin, Qiudan Li, Daniel D. Zeng, YongCheng Zhan, Ruoran Liu, Lei Wang, and Hongyuan Ma. 2016. Jointly Modeling Review Content and Aspect Ratings for Review Rating Prediction. In *Proc. SIGIR*. 4.
[21] Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. In *Proc. WSDM*. ACM, 219–230.
[22] Ralf Krestel and Nima Dokoohaki. 2011. Diversifying Product Review Rankings: Getting the Full Picture. In *Proc. WI-IAT*. IEEE Computer Society, 138–145.
[23] Theodoros Lappas, Mark Crovella, and Evimaria Terzi. 2012. Selecting a Characteristic Set of Reviews. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, 832–840.
[24] Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee, and Jidong Shao. 2017. Bimodal Distribution and Co-Bursting in Review Spam Detection. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, 1063–1072.
[25] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting Product Review Spammers Using Rating Behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)*. ACM, 939–948.
[26] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. 2010. Exploiting Social Context for Review Quality Prediction. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, 691–700.
[27] Luciana B. Maroun, Mirella M. Moro, Jussara M. Almeida, and Ana Paula C. Silva. 2016. Assessing Review Recommendation Techniques Under a Ranking Perspective. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media (HT '16)*. ACM, 113–123.

[28] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. ACM, 43–52.

[29] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proc Conference on Empirical Methods in Natural Language Processing*.

[30] Amanda J. Minnich, Nikan Chavoshi, Abdullah Mueen, Shuang Luan, and Michalis Faloutsos. 2015. TrueView: Harnessing the Power of Multiple Review Sites. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. International World Wide Web Conferences Steering Committee, 787–797.

[31] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. 2011. Review Recommendation: Personalized Prediction of the Quality of Online Reviews. In *Proc. CIKM*. 2249–2252.

[32] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. 2012. ETF: Extended Tensor Factorization Model for Personalizing Prediction of Review Helpfulness. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)*. ACM, 163–172.

[33] Elaheh Momeni, Claire Cardie, and Nicholas Diakopoulos. 2015. A Survey on Assessment and Ranking Methodologies for User-Generated Content on the Web. *ACM Comput. Surv.* 48, 3, Article 41 (Dec. 2015), 49 pages.

[34] Thanh-Son Nguyen, Hady W. Lauw, and Panayiotis Tsaparas. 2015. Review Synthesis for Micro-Review Summarization. In *Proc. WSDM*. ACM, 169–178.

[35] Gerardo Ocampo Diaz and Vincent Ng. 2018. Modeling and Prediction of Online Product Review Helpfulness: A Survey. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 698–708.

[36] Debanjan Paul, Sudeshna Sarkar, Muthusamy Chelliah, Chetan Kalyan, and Prajit Prashant Sinai Nadkarni. 2017. Recommendation of High Quality Representative Reviews in e-Commerce. In *Proc. RecSys*.

[37] Thiago R.P. Prado and Mirella M. Moro. 2017. Review Recommendation for Points of Interest's Owners. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT '17)*. ACM, 295–304.

[38] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction Through Double Propagation. *Comput. Linguist.* 37, 1 (March 2011), 9–27.

[39] Gerard Salton and Christopher Buckley. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management* 24 (01 1988).

[40] Sunil Saumya, Jyoti Singh, and Yogesh Dwivedi. 2019. Predicting the helpfulness score of online reviews using convolutional neural network. *Soft Computing* (02 2019).

[41] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi. 2017. NetSpam: A Network-Based Spam Detection Framework for Reviews in Online Social Media. *IEEE Transactions on Information Forensics and Security* 12, 7 (July 2017), 1585–1595.

[42] Vaishak Suresh, Syeda Roohi, and Magdalini Eirinaki. 2014. Using Social Data for Personalizing Review Rankings. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM.

[43] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly D. Voll, and Manfred Stede. 2011. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics* 37, 2 (2011), 267–307.

[44] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. 2013. Context-aware Review Helpfulness Rating Prediction. In *Proc. RecSys*. ACM, 1–8.

[45] Son Trinh, Luu Nguyen, Minh Vo, and Phuc Do. 2016. *Lexicon-Based Sentiment Analysis of Facebook Comments in Vietnamese Language*. Springer International Publishing.

[46] Panayiotis Tsaparas, Alexandros Ntoulas, and Evimaria Terzi. 2011. Selecting a Comprehensive Set of Reviews. In *Proc. KDD*. ACM, 168–176.

[47] Cong Wang, Guoqing Chen, and Qiang Wei. 2018. A temporal consistency method for online review ranking. *Knowledge-Based Systems* 143 (2018), 259 – 270.

[48] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In *Proc. KDD*. ACM, 783–792.

[49] Tak-Lam Wong, Wai Lam, and Tik-Shun Wong. 2008. An Unsupervised Framework for Extracting and Normalizing Product Attributes from Multiple Web Sites *(SIGIR '08)*. Association for Computing Machinery, 35C42.

[50] Y. Yang, C. Chen, and F. S. Bao. 2016. Aspect-Based Helpfulness Prediction for Online Product Reviews. In *Proc. ICTAI*. 836–843.

[51] Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. 2015. Semantic Analysis and Helpfulness Prediction of Text for Online Product Reviews. In *Proc. ACL*. Association for Computational Linguistics, 38–44.