Expert Systems With Applications 187 (2022) 115887

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Harshness-aware sentiment mining framework for product review

Xun Wang, Ting Zhou, Xiaoyang Wang, Yili Fang *

School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

ARTICLE INFO

Keywords:
Sentiment mining
Bayesian inference
Probabilistic graphical model

ABSTRACT

Sentiment mining has been a helpful mechanism that targets to understand the market feedback on certain commodities by utilizing user comments. In general, the process of yielding each comment is essentially associated with his/her criteria for rating (i.e., the degree of *harshness*) , which makes users provide biased comments. For instance, for a tolerant user, although the user is extremely dissatisfied with the product, harshness still makes her yield a neutral comment which cannot indicate the product quality. Existing work straightforwardly removes the comments of harsh users and those of tolerant ones, which is not the best strategy. To this end, we propose a harshness-aware sentiment analysis framework for product review. First, we depict the process of providing comments from users as a probabilistic graphical model in which the harshness is incorporated. Second, we employ a Bayesian-based inference for sentiment mining. Extensive experimental evaluations have shown that the results of the proposed method are more consistent with the expert evaluations than those of the state-of-the-art methods, and even outperform the method which infers the final evaluations with the ground truth of comments without considering users' harshness.

1. Introduction

Sentiment mining is to extract patterns from data such as product comments or public opinions (Wiebe, Wilson, & Cardie, 2005). Recently, with the rise of online commentary websites, such as Amazon, sentiment mining has attracted wide attention from the communities of academic and industry (Guan, Li, Gong, Sun, & Zhou, 2018; Pang, Lee, & Vaithyanathan, 2002), which contributes to a business decision based on product evaluation (Giatsoglou et al., 2017).

The product evaluation process can be mainly divided into two steps: *text analysis* (Cambria, 2016; Wang & Lu, 2018) and *viewpoint inference* (Cambria, Chandra, Sharma, & Hussain, 2010; Kauffmann et al., 2019), where text analysis is used to analyze user sentiments and viewpoint inference to obtain product evaluations. (1). Text analysis aims at analyzing and obtaining sentiments in user comments with natural language processing (NLP) (Liu, Xuan, Xuan, & Xuan, 2015; Rao, Huang, Feng, & Cong, 2018), which is viewed as a preprocess of data for viewpoint inference. (2). Then, viewpoint inference mainly uses statistical learning (such as majority voting and its variants Kauffmann et al., 2019) to provide a product evaluation.

The input of this two-step process is users' comments from the Internet. Due to the diversity of users' background, their criteria for rating (i.e., *harshness*) vary, which often makes them provide biased comments and strongly affects the final outcome of viewpoint inference. Although a tolerant user is extremely dissatisfied with the product, harshness still makes her yield a neutral comment. We analyze the comment distribution supplied by users, which support this intuition. As shown in Fig. 1, we present the proportion by ten users in the two datasets (*Amzon*[1] and *IMDB*[2]). We can observe that tolerant users (U_3, U_6, U_7, U_8 in Fig. 1(a) and U_4, U_5, U_6, U_8, U_9, U_{10} in Fig. 1(b)) are more likely to provide positive reviews (the blue part of the bar), while harsh users (U_1, U_2, U_4, U_5 in Fig. 1(a) and U_1, U_2, U_3 in Fig. 1(b)) are more likely give negative reviews (the violet part of the bar). Hence, the harshness makes a difference to product analysis. Nonetheless, most of existing works (Cambria, 2016; Cambria et al., 2010; Gong, Al Boni, & Wang, 2016; Kauffmann et al., 2019; Wang & Lu, 2018; Wang, Lu, & Zhai, 2010) neglected users' harshness which leads to unreliable results of the final product analysis. Consequently, the problem of how to incorporate users' harshness into the viewpoint inference with the aim of improving the accuracy of product analysis rises.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

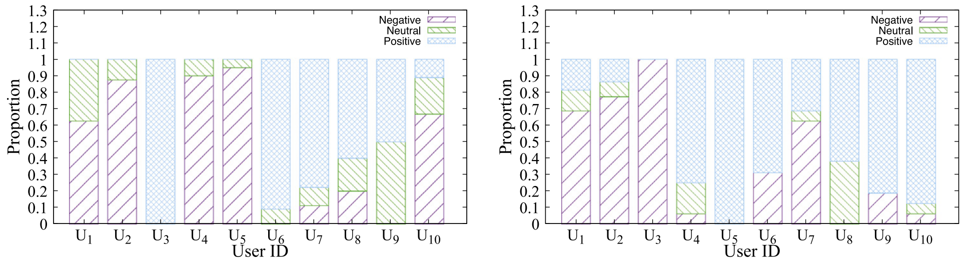
* Corresponding author.

E-mail addresses: xwang@zjgsu.edu.cn (X. Wang), 18020100012@pop.zjgsu.edu.cn (T. Zhou), xiaoyangw@zjgsu.edu.cn (X. Wang), fangyl@zjgsu.edu.cn (Y. Fang).

[1] https://nijianmo.github.io/amazon/index.html

[2] https://datasets.imdbws.com/

https://doi.org/10.1016/j.eswa.2021.115887
Received 10 January 2021; Received in revised form 4 September 2021; Accepted 5 September 2021
Available online 10 September 2021



(a) (b)

Fig. 1. (a) User statistics of the methods with real-world dataset *Amazon*, (b) User statistics with real-world dataset *IMDB*.

In this work, we first present a harshness-based framework (*HBF*) for product analysis that involves text analysis and viewpoint inference. In text analysis, the number of commodity reviews is relatively small and the data is sparse. We employ a Bayesian-based model to depict the correlation among the comments, the product evaluation and the harshness for aspect-level sentiment. Comments are split by extracting product attributes and Support Vector Machine (widely adopted in sentiment mining and text analysis Borg & Boldt, 2020) is employed for analyzing user sentiments in comments. Finally, in viewpoint inference, we introduce a probabilistic graphical model to incorporate users' harshness with text analysis for the inference of user comments. The users' harshness is modeled as a latent variable, and the expectation–maximization (EM) algorithm (Moon, 1996) is used to solve the latent variable and infer true evaluations of products. The main contributions of our work are as follows.

- We propose a framework *HBF* for product analysis based on sentiment mining that well portrays the harshness of users.
- We employ a Bayesian-based model to analyze user sentiments and propose a probabilistic graphical model to obtain product evaluations with the consideration of users' harshness.
- Extensive experiments are conducted in two real-life datasets. The experimental results have shown that the results of proposed method are more consistent with expert evaluations compared with the state-of-the-art methods.

We will provide the code and data upon the publication of this work. The rest of the paper is organized as follows. Section 2 formalizes the problem studied and outlines our harshness-based framework for product analysis. Section 3 introduces the details of our approach. The experimental results are shown in Section 4. Section 5 discusses the related works.and we conclude the paper in Section 6.

2. Problem formulation and framework

We formalize the problems studied in this section. First, we present the related concepts. Then, we define the feature extraction problem (FEP) and product analysis problem (PAP) investigated in this paper. Finally, we outline our harshness-based framework for product analysis. Table 1 shows the notations frequently used in this paper.

2.1. Problem formulation

User comments usually are document-level and contain product evaluations. However, each product has many attributes and sometimes the evaluation of a product attribute is not consistent with the product evaluation. Meanwhile, different product attributes attain different evaluations in the same comments. Therefore, in order to obtain more precise evaluations, we conduct aspect-level sentiment analysis. Specifically, we split a comment into several short sentences and obtain the corresponding attribute commented by each sentence. Subsequently, the sentences are merged according to the attributes to obtain an attribute sentences' vector.

Table 1
Notation description.

Notation	Description
I	The total number of users
J	The total number of products
T	The total number of attributes
L	The total number of type of evaluation
C	The set of comments
c_{ij}	The comment by user i for product j
$\mathcal{A}$	The set of attribute sentences' vectors
A_{ij}	The attribute sentences' vector of user i for product j
a^t_{ij}	The set of sentences commented by user i on attribute t for product j.
j^t	The attribute t of product j
$\mathcal{Z}$	The evaluation set of products
R	The three-by-three confusion matrix
$\mathcal{S}$	The set of attribute sentences' sentiments
s^t_j	The set of sentiment' probabilities of product attribute j^t
s^t_{ij}	The set of sentiment' probabilities of user i for product attribute j^t
$\mathcal{Y}$	The set of true product evaluations
y^t_j	The true evaluation of product attribute j^t
$\mathcal{T}$	The set of evaluations' difficulties of product attribute
$1/\tau^t_j$	The difficulty of the evaluation of product attribute j^t
$\mathcal{U}$	The set of users' harshness
μ_i	The harshness of user i

Given a comment c_{ij} of user i ($i \leq I$) for product j ($j \leq J$), the comment c_{ij} first can be divided into several sentences. Then, we judge the product attribute discussed by each sentence and assign a product attribute to each sentence. Furthermore, we merge sentences of the same attribute and acquire an attribute sentences' vector $A_{ij} = \langle a^1_{ij}, \ldots, a^t_{ij}, \ldots, a^T_{ij} \rangle$ where a^t_{ij} called an attribute sentence denotes the comment about tth attribute in A_{ij}. Formally, we define the feature extraction problem as follows.

Definition 1 (*Feature Extraction Problem—FEP*)**.** Let $C = \{c_{ij} | i \leq I, j \leq J\}$ denote the comments set, where c_{ij} denotes the comment of user i for product j. Feature extraction problem is to find an accurate attribute sentences' vector A_{ij} for every $c_{ij} \in C$.

Definition 1 gives a feature extraction problem about attaining segmented comments classified by attributes. For instance, for the comment "*the plot of this movie is very attractive, but the music is very annoying. By the way, the turning point of the story is what I like*.", it is divided into two parts according to the plot and the soundtrack. Subsequently, we can get the attribute sentences' vector ⟨"*the plot of this movie is very attractive, the turning point of the story is what I like*.","*the music is very annoying*"⟩ where "*music is very annoying*" is an attribute sentence, which evaluates the soundtrack.

Since each user has a different degree of harshness, user comments usually cannot accurately convey user true opinions. Therefore, we take this problem into account and analyze user true opinions based on the comments to acquire true product evaluations. With the users' harshness, we define the product analysis problem as follows.

Definition 2 (*Product Analysis Problem—PAP*). Let $\mathcal{A} = \{A_{ij} | i \leq I, j \leq J\}$ be the set of attribute sentences' vectors where A_{ij} be the attribute sentences' vector of the comment c_{ij}, $\mathcal{U} = \{\mu_i | i \leq I\}$ be the set of users' harshness where μ_i be the harshness of user i, $\mathcal{Z} = \{Negative, Neutral, Positive\}$ be the evaluation set. The product analysis problem is to find the function $g : \mathcal{A} \times \mathcal{U} \rightarrow \mathcal{Z}^J$, which aims to provide a true evaluation of each product.

According to Definition 2, we introduce the users' harshness based method to obtain true product evaluations. Since the user comments are split by attribute, we can get a more fine-grained and accurate evaluation of product attributes. For example, the number of positive, neutral, and negative comments in the movie is 3, 5, and 12 respectively. After segmenting and analyzing the comments of the movie according to the attributes, the number of positive, neutral and negative evaluations about plots is 4, 12, and 4 respectively. Thus, it is concluded that the plots' evaluation of the movie is neutral rather than negative.

In the meantime, there is such a situation in real life that a user is a tolerant person, and he evaluates most of the movies as positive and the rest as neutral, i.e., he never gives a negative evaluation. Hence, when he gives a neutral evaluation to a movie, which means he hates this movie, the true evaluation of the movie should be negative. The evaluations of harsh and tolerant users should be corrected rather than be deleted as in Gong and Wang (2018). This is because if the users' harshness is taken into consideration, product analysis can also benefit from these data. For example, the number of positive, neutral, and negative evaluations about the movie plots is 3, 11, and 6 respectively, so the plots' evaluation is considered to be neutral under normal circumstances. But five tolerant persons who always give positive evaluations give a neutral evaluation to plots. Hence, their evaluations should change to negative after consideration of the users' harshness. Then, the number of positive, neutral, and negative evaluations become 3, 6, and 11 where the plots' evaluation is changed to negative. In view of this, we ought to consider the users' harshness and reassess user opinions to acquire high-quality product evaluations.

2.2. Framework

For the above two problems, we propose a general framework for product analysis as shown in Fig. 2. Different from previous product analysis methods, we consider the users' harshness to get more accurate product evaluations. The workflow can be divided into the following steps.

Step 1. This step acquires user comments on products on the online comment website, such as *IMDB* and *Amazon*.

Step 2. The users, products, and comments are extracted from the user comments separately.

Step 3. Through the Bayesian model, aspect terms are extracted and clustered to acquire product attributes. Then, the comments are split into attribute statement vectors.

Step 4. The attribute sentences' vectors utilize the SVM model to determine sentiments contained in the sentence itself.

Step 5. This step combines the sentiments in step 4 with the extracted users and products in step 2, to obtain users' viewpoints and the final product evaluations via the inference model based on users' harshness that this paper focuses on.

Step 3 and 4 involve our proposed Bayesian model for text analysis, and step 5 relates to the probability graphical model for viewpoint inference. The research in this paper focuses on step 5 to obtain accurate product evaluation results by considering the harshness of users.

3. Our HBF methods

We try to obtain more precise product evaluations based on users' harshness. We first utilize a Bayesian model for text analysis, which enables us to get comments on each product attribute from the overall comment. Then, we propose a probabilistic graphical model to conduct viewpoint inference based on the users' harshness.

3.1. Text analysis with Bayesian model

User comments usually include the evaluations of multiple attributes of products. It is inaccurate to analyze the whole comment directly. Therefore, we aim to acquire the corresponding attribute sentences' vector according to product attributes and carry out aspect-level sentiment analysis. At first, aspect terms are extracted and clustered to acquire product attributes from user comments. On this basis, comments are divided according to product attributes and we obtain attribute sentences' vector.

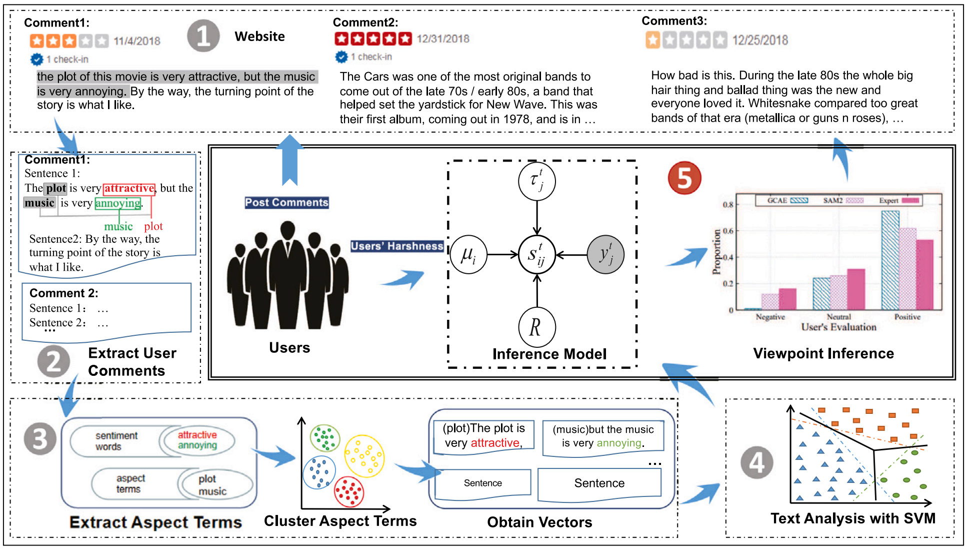
In general, comments consist of two components, i.e., sentiment words and aspect terms. The former is usually adjectives, *e.g.* "good". The latter usually involves nouns, which appear near the sentiment words (Hu & Liu, 2004), such as "performance" in movies. Consequently, we locate all adjectives and single out all the nouns that are in the five words before and after the adjectives. For example, in the sentence "The plot is very attractive, but the music is very annoying", we first locate the adjectives "attractive" and "annoying". Then, we can filter out "plot" and "music", based on the five words before and after "attractive", Similarly, we can filter out "music" based on "annoying". Finally, with "music", we derive "plot" and "music" as the candidate aspect terms. After that, duplicate nouns are merged and used as candidate aspect terms. Next, we sort the candidate aspect terms based on word frequency, and filter the candidate aspect terms that lower than the threshold. Finally, the final aspect terms are determined manually according to domain relevance. Since aspect terms that describe the same product attribute always appear in a similar context, we can cluster aspect terms to acquire product attributes, such as "cast" in movies. Therefore, the mean shift (Comaniciu & Meer, 2002) algorithm is used to cluster the aspect terms. To carry out feature extraction of user comments, we first divide a user comment into several single sentences. For each sentence, we delete product names extracted by user comments as those are related to the whole product rather than a product attribute. Afterward, we remove the stop words and annotate the part-of-speech of words in user comments. Second, we extract the nouns from sentences and then carry out entity detection. The detected entities include personal names, place names, institution names, numbers, dates, etc., except for personal names, others are meaningless in product analysis. Therefore if the noun is not a personal name, it is deleted, otherwise, it is replaced by the corresponding product attribute, such as "cast" in movies. Moreover, the nouns are further filtered by TF–IDF value and term frequency (Salton & Buckley, 1988), because the noun with low TF–IDF value or high term frequency is usually useless. If there is no eligible noun extracted from the sentence, the sentence is deleted. Thirdly, we judge the product attribute discussed by each sentence and assign a product attribute to each sentence. For each extracted noun n_m in the sentence $Sent$, we calculate the distance $d^{(n_m,t)}$ between the noun n_m and each product attribute t where n_m denotes mth noun in extracted noun. Specifically, $d^{(n_m,t)}$ is defined as:

$$d^{(n_m,t)} = \sum_{k \leq N_t} \frac{(n_m - r_{tk})^2}{N_t}, \tag{1}$$

where r_{tk} denotes the extracted kth aspect term in the product attribute t and N_t denotes the total number of aspect terms in the product attribute t. So, we obtain a correspond distance vector $D(n_m) = \langle d^{(n_m,1)}, \ldots, d^{(n_m,t)}, \ldots, d^{(n_m,T)} \rangle$. Then, we add the distance vectors of all nouns to get the distances between the sentence $Sent$ and each product attribute.

$$d^{(Sent,t)} = \sum_{n_m \in Sent} d^{(n_m,t)}. \tag{2}$$

After this, we can get the corresponding distances $D(Sent) = \langle d^{(Sent,1)}, \ldots, d^{(Sent,t)}, \ldots, d^{(Sent,T)} \rangle$, which are between the sentence and each



the plot of this movie is very attractive, but the music is very annoying. By the way, the turning point of the story is what I like.

The Cars was one of the most original bands to come out of the late 70s / early 80s, a band that helped set the yardstick for New Wave. This was their first album, coming out in 1978, and is in …

How bad is this. During the late 80s the whole big hair thing and ballad thing was the new and everyone loved it. Whitesnake compared too great bands of that era (metallica or guns n roses), …

Comment1:
Sentence 1:
The plot is very attractive, but the music is very annoying.
Sentence2: By the way, the turning point of the story is what I like.

Comment 2:
Sentence 1: …
Sentence 2: …
…

Fig. 2. *HBF* framework for product analysis.

product attribute. The closest product attribute is assigned to the sentence $Sent$. Finally, we count the product attribute of all sentences, merge the sentence from the same product attribute, and obtain attribute sentences' vectors. For example, the sentence "The plot is very attractive, but the music is very annoying". is first divided into two phrases, i.e., phrase 1 "The plot is very attractive" and phrase 2 "but the music is very annoying". Then, in Phrase 1, we remove the stop words and annotate the part-of-speech of words to get the noun "plot" and the adjective "attractive". Next, the noun "plot" is extracted for entity detection and TF–IDF, and then the word frequency is calculated. The final product attribute is plot (evaluated by the noun "plot"). Similarly, Phrase 2 is evaluated by the noun "music" to obtain the product attribute of the phrase as sound track.

3.2. Viewpoint inference for user comments

Product evaluations made by users should be adjusted based on user comments and users' harshness, since different people have different degree of harshness.

Attribute sentence analysis. In the previous section, the attribute sentences' vectors are obtained. Combining document embedding pretrained by TF–IDF model and SVM model, we can test the attribute sentences' vectors and acquire the attribute sentences' sentiments. We denote the set of attribute comments' viewpoint $S = \{s_{ij}^t | i \leq I, j \leq J, t \leq T\}$ where $s_{ij}^t \in \{Negative, Neutral, Positive\}$ is the viewpoint of user i for product attribute j^t. In addition, $\mathcal{Y} = \{y_j^t | j \leq J, t \leq T\}$ denotes a set of true product evaluations where y_j^t is the true evaluation of product attribute j^t.

We construct a three-by-three confusion matrix R where $R(z_l, z_h)$ represents the probability of the viewpoint a_{ij}^t is evaluated as z_l when the true evaluation y_j^t of product attribute j^t is z_h. Specifically, $R(z_l, z_h)$ is defined as:

$$R(z_l, z_h) = \begin{cases} s_{ij}^{tz_h} & z_l = z_h \\ \frac{1 - s_{ij}^{tz_h}}{2} & z_l \neq z_h. \end{cases} \tag{3}$$

where $R(z_l, z_h)$ denote the probability that the viewpoint z_h of product attribute is evaluated as z_l by our model without considering users' harshness.

Users' harshness inference. Since users' harshness varies from person to person, the observed evaluations of product attributes are not on behalf of the user's true feeling to a certain extent. At the same time, the difficulty of evaluating products also affects users, as high-quality products are usually easily evaluated positive and vice versa. Therefore, we can determine the harshness of users more accurately by considering the difficulty of evaluating products. The observed user sentiments of each product attribute depend on four causal factors: (1) the true evaluation of each product attribute; (2) the user comments; (3) the harshness of each user; and (4) the difficulty of each product attribute's evaluation. In view of it, all of the factors mentioned above should be considered together to obtain high-quality and realistic product evaluations.

We model the difficulty of the evaluation of product attribute j^t using the parameter $1/\tau_j^t \in [0, \infty)$ where τ_j^t is constrained to be positive. Here $1/\tau_j^t = \infty$ means that evaluating product attribute j^t is extremely difficult where the product attribute j^t is similar to most product attributes or many background knowledge and relevant competencies is needed to evaluate it. Even relatively professional experts have only a 33.33% chance of being able to evaluate correctly. $1/\tau_j^t = 0$ means that it is easy to evaluate product attributes. The larger $1/\tau_j^t$ is, the more difficult that the user to evaluate product attributes. Moreover. the harshness of each user i is modeled by the parameter $\mu_i \in (-\infty, +\infty)$. Here $\mu_i = +\infty$ means the user always gives the same evaluation as most people; $-\infty$ means the user always gives the different evaluation from most people and his/her standard is especially different from ordinary people. Finally, $\mu_i = 0$ means that the user has no idea about this product, i.e., his/her evaluation carries no information about the true evaluation and give the evaluation randomly.

We represent the probability that s_{ij}^t is equal to z given y_j^t, μ_i, τ_j^t, R as follows.

$$p(s_{ij}^t = z | y_j^t, \mu_i, \tau_j^t, R) \propto \begin{cases} \frac{1}{1 + e^{-\mu_i \tau_j^t}} R(z, y_j^t) & z = y_j^t \\ \frac{1 - \frac{1}{1+e^{-\mu_i \tau_j^t}} R(z, y_j^t)}{L - 1} & z \neq y_j^t \end{cases} \tag{4}$$

where y_j^t is the true evaluation of the product attribute j and μ_i is the harshness parameter of user i. When harshness μ_i of user i tends to positive infinity, user i is more right and fair, user i is more likely to give an unbiased evaluation, that is $p(s_{ij}^t = z | y_j^t, \mu_i, \tau_j^t, R) = R(z, y_j^t)$.

We have s_{ij}^t being the observed sentence sentiment falling in evaluation domain $\mathcal{Z}$ and a confusion matrix R. The unobserved variables are the true evaluation y_j^t, the harshness parameter μ_i, the difficulty parameter τ_j^t. In this model, our goal is to find the posterior distribution of y_j^t and select the evaluation y_j^t with the maximum a posterior estimation as the final evaluation of product attribute j^t.

The prior distribution of y_j^t is a uniform discrete distribution over evaluation domain $\mathcal{Z}$. For simplicity, we ignore the prior of μ_i and τ_j^t and we use EM algorithm to obtain maximum likelihood estimates of the parameters of μ_i and τ_j^t similar to Whitehill, Wu, Bergsma, Movellan, and Ruvolo (2009).

Algorithm 1 EM Algorithm for Viewpoint Inference

```
Input: Observed sentence sentiments $\mathcal{S}$ and Confusion matrices $R$
Output: Product evaluations $\mathcal{Y}$
Initialization :
harshness parameters $\mathcal{U}$
difficulty parameters $\mathcal{T}$
for $n = 1$ to maxiter do
    if sum of harshness and difficulty errors < tolerance then
        break
    end if
    E step : //calculate by Eq. (5)
    compute $p(y_j^t|\mathcal{S}, \mathcal{U}, \mathcal{T}, R)$
    M step : //calculate by Eq. (6)
    update $\mathcal{U}$,$\mathcal{T}$ by $\max_{\mu,\tau} E[\ln p(\mathcal{S}, \mathcal{Y}|\mathcal{U}, \mathcal{T}, R)]$
end for
$y_j^t = \max_{y_j^t} p(y_j^t|\mathcal{S}, \mathcal{U}, \mathcal{T}, R)$
```

E-step. Let $s_j^t = \{s_{ij}^t | i \le I\}$ be the sentiment set of product attribute j^t. The user does not evaluate each product or each product attribute. In this case, the index variable i in s_{ij}^t refers only to those users who evaluated the product attribute j^t. We can compute the posterior probabilities of all $y_j^t \in \mathcal{Z}$ given the $\mathcal{U}$, $\mathcal{T}$ values from the last maximization step and the observed $\mathcal{S}$, R:

$$\begin{aligned} p(y_j^t|\mathcal{S}, \mathcal{U}, \mathcal{T}, R) &= p(y_j^t|s_j^t, \mathcal{U}, \tau_j^t, R) \\ &\propto p(y_j^t|\mathcal{U}, \tau_j^t) p(s_j^t|y_j^t, \mathcal{U}, \tau_j^t, R) \\ &\propto p(y_j^t) \prod_i p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R), \end{aligned} \tag{5}$$

where we noted that $p(y_j^t|\mathcal{U}, \tau_j^t) = p(y_j^t)$ using the conditional independence assumptions in the probabilistic graphical model, that is, the event y_j^t is conditionally independent for given events $\mathcal{U}$ and τ_j^t. And $p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R)$ can be obtained with Eq. (4).

M-step. We compute the posterior probabilities of the $\mathcal{Y}$ values during the last expectation step and then maximize the standard auxiliary function Q. Function Q is defined as the expectation of the joint log-likelihood of the observed and hidden variables $(\mathcal{S}, \mathcal{Y})$ given the parameters $(\mathcal{U}, \mathcal{T})$. We compute the function Q as follows.

$$\begin{aligned} &Q(\mathcal{U}, \mathcal{T}, R) \\ &\quad = E[\ln p(\mathcal{S}, \mathcal{Y}|\mathcal{U}, \mathcal{T}, R)] \\ &\quad = E\left[\ln \prod_{jt} \left(p(y_j^t) \prod_i p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R)\right)\right] \\ &\quad = \sum_{jt} E\left[\ln p(y_j^t) + \sum_i \ln p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R)\right] \\ &\quad = \sum_{jt} E[\ln p(y_j^t)] + \sum_{ijt} E[\ln p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R)] \\ &\quad = Const + \sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p(y_j^t|\mathcal{S}, \mu_i^{old}, \tau_j^{old}, R) \ln p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R), \end{aligned} \tag{6}$$

where $p(y_j^t|\mathcal{S}, \mu_i^{old}, \tau_j^{old}, R)$ and $\mathcal{S}$ can be obtained with $\mu_i^{old}, \tau_j^{old}$ that are already estimated in E-step.

To simplify the presentation, Based on Eq. (4), we redefine the probability of user evaluation as follows.

$$p(s_{ij}^t = k|y_j^t = k, \mu_i, \tau_j^t, R) = \varrho R = \sigma. \tag{7}$$

where $\varrho = \frac{1}{1+e^{-\mu_i \tau_j^t}}$,For the multi-class case, we further assume uniform probability over all incorrect responses, i.e., for all $k' \neq k$,

$$p(s_{ij}^t = k'|y_j^t = k, \mu_i, \tau_j^t, R) = \frac{1}{L-1}(1-\sigma), \tag{8}$$

where L denotes the total number of type of evaluation. Therefore, we modify slightly the equations for the probability of user evaluation and the auxiliary function:

$$p(s_{ij}^t|y_j^t, \mu_i, \tau_j^t, R) = \sigma^{\delta(s_{ij}^t, y_j^t)} \left(\frac{1}{L-1}(1-\sigma)\right)^{1-\delta(s_{ij}^t, y_j^t)}, \tag{9}$$

where $\delta(s_{ij}^t, y_j^t)$ is the Kronecker delta function (Robb & Jiang, 1999). For brevity we write $\delta(s_{ij}^t, y_j^t)$ simply as δ, $p^k = p(y_j^t = k|\mathcal{S}, \mu^{old}, \tau^{old}, R)$ and ignore constant values. Then we can define Q as:

$$\begin{aligned} Q(\mathcal{U}, \mathcal{T}, R) &= \sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p^k \ln [\sigma^\delta (\frac{1}{L-1}(1-\sigma))^{1-\delta}] \\ &= \sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p^k [\delta \ln \sigma + (1-\delta) \ln(\frac{1}{L-1}(1-\sigma))]. \end{aligned} \tag{10}$$

To maximize Q, we can differentiate function Q to get the gradients:

$$\begin{aligned} \frac{\partial Q}{\partial \mu_i} &= \sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p^k \delta \frac{1}{\sigma} R \varrho (1-\varrho) \tau + \\ &\sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p^k (1-\delta) \frac{1}{1-\sigma} (-R) \varrho (1-\varrho) \tau \\ &= \sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p^k [\delta (1-\varrho) \tau - (1-\delta) \frac{1}{1-\sigma} \sigma (1-\varrho) \tau]. \end{aligned} \tag{11}$$

Similarly, we can get:

$$\frac{\partial Q}{\partial \tau_i} = \sum_{ijt} \sum_{y_j^t \in \mathcal{Z}} p^k [\delta (1-\varrho) \mu - (1-\delta) \frac{1}{1-\sigma} \sigma (1-\varrho) \mu]. \tag{12}$$

where symbols δ, σ and ϱ are discussed earlier in Section 3.2. With Eq. (7), we can use gradient ascent (Baxter, Weaver, & Bartlett, 1999) to obtain τ_i and μ_i that maximizes Q in the current step. We then iteratively apply E-Step and M-Step to obtain τ_i and μ_i, and to maximize the likelihood.

The EM algorithm is summarized in Algorithm 1. Through the EM algorithm (Moon, 1996), we solve two latent variables, which are the harshness parameter $\mathcal{U}$ and the difficulty parameter $\mathcal{T}$, and obtain true evaluations $\mathcal{Y}$ of products.

4. Experimental evaluation

We conducted extensive experiments on two datasets to evaluate our model and report empirical results. We first prepared the data needed for the experiment. Next, we introduce the experimental settings. Then, some evaluation metrics are introduced to validate our proposed model. Finally, we present the evaluation results of the model.

4.1. Datasets

We conducted experiments on two sentiment classification datasets with user and product information, which are from *IMDB* (Tang, Bing, & Liu, 2015) and music subset of *Amazon* (He & Mcauley, 2016). The details of the datasets are shown in Table 2. The label ranges from 1 to 10 on *IMDB* and ranges from 1 to 5 on *Amazon*. Because we aim at dividing evaluations into *positive*, *neutral* and *negative*, we quantified the label. To balance the dataset, we set 1 to 4 as *negative*, 5 to 6 as *neutral* and 7 to 10 as *positive* on *IMDB* while we set 1 to 3 as *negative*, 4 as *neutral* and 5 as *positive* on *Amazon*.

Due to the lack of the standard classification of attributes commented by each sentence, we invited experts to mark attributes based on the clustering result of attributes, of 1000 sentences on *IMDB* and 500 sentences on *Amazon*, to test the accuracy of feature extraction. Besides, in order to train the SVM model that used to analyze the sentiment of attribute sentences, we also invited experts to evaluate attribute sentences manually. 2000 sentences were annotated as training sets and 200 sentences were annotated as testing sets on *IMDB*. Similarly, 1200 sentences were annotated as training sets and 600 sentences were annotated as testing sets on *Amazon*. Since we aim to verify the consistency between the evaluation of our model and true product evaluations, we collected ground truth on the corresponding famous review website. We considered professional reviewers on the website as domain experts, because they have sufficient professional qualities and have been strictly checked by the website. *All the datasets will be released if accepted.*

4.2. Experimental settings

In our experiments, we employ the NLTK natural language processing toolkit (Bird, Klein, & Loper, 2010) to remove stop words. The Stanford CoreNLP natural language processing toolkit (Manning et al., 2014) is used to separate sentences as tokens, annotate the parts of speech and entity detection. Next, due to the clustering effect less than satisfactory on high-dimensional data by mean shift algorithm, the *t-SNE* algorithm (Maaten & Hinton, 2008) is utilized to reduce the dimensionality of data. In the meantime, we employ the silhouette coefficient (Rousseeuw, 1987) to select the best parameters of mean shift algorithm. Then, on each dataset, we leverage the SkipGram (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) to pre-train word embeddings with 300 dimensions and train the TF–IDF model (Řehůřek & Sojka, 2010) that we used to get 200-dimensional document embedding. These two models all implemented in Gensim (Řehůřek & Sojka, 2010) and the SVM model implemented in Scikit-Learn (Pedregosa, Varoquaux, Gramfort, Michel, & Louppe, 2013). Finally, the EM algorithm is optimized by using SLSQP algorithm (Kraft, 1988) implemented in SciPy (Jones, Oliphant, & Peterson, 2014). When using these tools, all parameters are set to their default values.

In order to verify the correctness and effectiveness of our method, we implemented a comparison of eight methods. They are shown in Table 3.

- **ComTL** uses statistical methods for product analysis according to the ground truth of origin dataset, which represents the upper limit of sentiment mining of user comments, i.e., the accuracy rate of sentiment mining is 100%.
- **ATAE** is an attention-based LSTM that used to conduct aspect-level sentiment analysis. It appends the given aspect embedding with each word embedding as the input of LSTM and let aspect embedding play a role in computing the attention weight (Wang, Huang, Zhao, et al., 2016).
- **IAN** is an interactive attention network for aspect-level sentiment analysis, which is also based on LSTM and attention mechanisms. The model learns the attentions in the contexts and the aspects interactively and generates representations of contexts and aspects, respectively (Ma, Li, Zhang, & Wang, 2017).
- **GCAE** is a CNN-based model to solve the problem of aspect-level sentiment analysis. It uses the gated Tanh-ReLU units to selectively output the sentiment features according to the given aspect embedding (Xue & Li, 2018).
- **HUB** is a probabilistic generative model developed to integrate two companion learning tasks of opinionated content modeling and social network structure modeling for users. User preference is considered by social networks. And the model deletes many comments of harsh and tolerant users in data pre-processing (Gong & Wang, 2018).
- **HUAPA** takes a framework to encode user and product information and consider user preference of word-level. Two individual hierarchical neural networks are applied to generate two representations, with user attention or with product attention (Wu, Dai, Yin, Huang, & Chen, 2018).
- **AAL-LEX** is a framework that combines aspect-aware learning. Through a two-way neural network, it can simultaneously supervise AAL components and sentiment classification. The AAL component is based on a dictionary (Zhu, Chen, Zheng, & Qian, 2019).

Table 2
Statistics of *IMDB* dataset and *Amazon* dataset. #sens/com represent average number of sentences contained in per document.

Datasets	#comments	#users	#products	#sens/com
IMDB	67,426	1310	1635	16.08
Amazon	9288	575	614	10.17

Table 3
Comparative method statistics.

Methods	Label	Preference	Aspect	Harshness
ComTL (Idealized)	✓	–	–	–
ATAE (2016)	–	–	✓	–
IAN (2017)	–	–	✓	–
GCAE (2018)	–	–	✓	–
HUB (2018)	–	✓	–	–
HUAPA (2018)	–	✓	–	–
AAL-LEX (2019)	–	–	✓	–
AAL-SS (2019)	–	–	✓	–
SVM-MV	–	–	✓	–
SVM-WMV	–	✓	✓	–
HBF (our method)	–	–	✓	✓

Notes: Label means ground truth is used for analysis. Preference means user preference is considered. Aspect means an aspect-level sentiment analysis. Harshness means users' harshness is considered. ✓ denotes the factor is considered in model, while — denotes not.

- **AAL-SS** is similar to AAL-LEX. But the AAL component uses sentence-level supervision (Zhu et al., 2019).
- **SVM-MV** splits each comment into several attribute sentences according to product attributes and then using SVM as the sentiment classifier for text analysis. Then, it uses majority voting (MV) to infer user comments in the viewpoint inference. It is the vanilla versions of our method.
- **SVM-WMV** is based on the SAM method given in this paper. It obtains the weight in the viewpoint inference according to the user preferences previously obtained and uses weight majority voting to analyze the product. It is the second simplified versions of our method
- **HBF** uses the text analysis method of this work. In the viewpoint inference, we carry on the inference according to the proposed Bayesian method in which the users' harshness is considered.

We use the hand-labeled sentences with aspects to training all the models considered of aspect. Furthermore, we evaluate each attribute of each product by accumulating evaluations of users and picking the most numerous one as the final result. If the number of two evaluations is equal, then choose randomly.

4.3. Evaluation metric

To evaluate the result, different evaluation metrics are employed. We first adopt evaluation metrics of sentiment analysis, which are widely used for evaluating the experimental results. Evaluation metrics are accuracy, recall, and f-measure (Singh & Kaur, 2015). The formula is as follows.

Accuracy: It is measured as the proportion of correctly classified instances to the total number of instances being evaluated.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{13}$$

where true positive(TP) is truly classified as positive, false positive (FP) is not labeled by the classifier as positive but should be, true negative (TN) is truly classified as negative, and false negative (FN) is not labeled by the classifier as negative but should be.

Precision and Recall: Precision is also referred to measure the exactness, and recall is used to measure the completeness of the model, which are defined as follows.

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}. \tag{14}$$

Table 4
Product attributes and aspect terms for different datasets.

Datasets	Attribute	Aspect terms
IMDB	Scene	Sequence stunt scene moment chase fight battle circumstances antic
	Plot	Adventure epic plot drama thriller premise conceptsuspense storyline story theme
	Cast	Performance acting portrayal cast role actor actress filmmaker director face
	Soundtrack	Score soundtrack music song photography cinematography editing costume setting
Amazon	Musician	Singer vocalist guitarist musician artist performer songwriter drummer lyricist
	Instrument	Horn percussion drum keyboard guitar orchestrationinstrument guitar piano
	Melody	Hook melody rhythm lyric beat content groove riff tune flow tempo soundtrack
	Style	Metal punk rock rap pop disco hiphop wave jazz blue dance type genre style coast

Table 5
Experimental results on *IMDB* dataset.

Methods	Accuracy	Recall			F-score		
		Negative	Neutral	Positive	Negative	Neutral	Positive
ComTL (Idealized)	61.00	25.00	22.60	94.30	40.00	29.79	75.19
ATAE (2016)	62.00	18.60	19.40	**100.0**	31.58	32.43	73.61
IAN (2017)	63.00	12.50	51.61	84.91	22.22	50.79	75.63
GCAE (2018)	62.00	6.30	38.70	92.50	11.76	43.64	76.56
HUB (2018)	62.00	43.75	9.68	98.11	50.00	17.14	75.91
HUAPA (2018)	62.00	25.00	22.58	96.23	40.00	29.79	76.69
AAL-LEX (2019)	63.00	25.00	64.52	73.58	40.00	54.79	72.90
AAL-SS (2019)	64.00	6.25	74.19	75.47	11.76	59.74	75.47
SVM-MV	69.00	12.50	**77.40**	81.10	22.22	67.61	77.48
SVM-WMV	70.00	12.50	74.19	84.91	22.22	**67.65**	78.95
HBF (our method)	**73.00**	**50.00**	61.30	86.80	**57.14**	66.67	**80.00**

Note: Best scores are in bold.

F-measure: It is referred as the harmonic mean of precision and recall.

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}. \quad (15)$$

Besides, we calculate the kappa coefficient Cohen (1960) and the kendall coefficient (Kendall, 1938). These two coefficients are used for consistency testing and can also be used to measure classification accuracy. Therefore, we verify the consistency between the evaluation of the model and expert evaluation by using them. The bigger the coefficient is, the more consistent the results are. A coefficient of 1 means that there is a perfect agreement between the prediction and the actual values; whereas a coefficient of 0 means that the prediction is completely unrelated to the actual values.

4.4. Empirical study I with dataset IMDB

Evaluation of feature extraction. In order to conduct aspect-level sentiment analysis, we performed feature extraction of user comments to obtain sentences that comment on different product attributes. The accuracy of the feature extraction is 71%. Through the research of feature extraction results, we find that the product attributes are scene, plot, cast, and soundtrack in this movie dataset. The aspect terms of the dataset are shown in Table 4. This is consistent with reality and demonstrates the effectiveness of our approach.

Evaluation of viewpoint inference. We collected expert evaluations of 25 movies on the Rotten Tomatoes.[3] we use accuracy, recall, and F-measure to evaluate our approach. The results are shown in Table 5. We have the following observations. To begin with, our methods using feature extraction outperform the methods using aspect embedding in terms of accuracy. It means that feature extraction is more able to detect the product attributes expressed in each sentence. Based on account of consideration of user preference, the *SVM-WMV* method has subtle accuracy improvements than the *SVM-MV* method. Then, since positive comments are the majority in the dataset, most of the results obtained through the *ATAE*, and *GCAE* methods are also positive. Therefore, these methods have a higher value in the positive case but the lower value in other cases. Compared with these two models, the *IAN* method used two attention layers to interactively guide the representation learning of context and aspects and achieved improvement in neutral comments. Next, the *AAL-LEX* method and the *AAL-SS* method were guided by clear aspect information, thereby enhancing the interaction between aspect learning and sentiment classification tasks, so the model achieved better performance. Besides, the *HUB* method and the *HUAPA* method that consider user preference achieve a more accurate analysis of the negative in recall and F-measure. It means that considering user preference can better understand the true opinions of many tolerant people. Finally, our *HBF* method achieves the highest value on the negative and neutral side. The comparisons indicate that after considering the users' harshness, positive evaluations given by some tolerant users begin to turn into neutral or negative evaluations while negative evaluations given by some harshness users begin to turn into neutral or positive evaluations.

In Table 5, since the *AAL-SS* and *AAL-LEX* show the best accuracy among the existing methods, *SVM-MV* and *SVM-WMV* use *SVM* as text analysis methods, which are two simplified versions of our method *HBF*. As shown in Fig. 3, *AAL-SS* method tends to obtain a positive result, which is more in line with the users' labels. However, in *SVM-MV*, *SVM-WMV* and *HBF*, and expert evaluations, positive evaluations are decreased with increasing negative evaluations and neutral evaluations. Our method *HBF* with users' hardness is the most consistent method with expert evaluations among the 5 methods. As discussed, the tolerant user tends to give a *neutral* evaluation, but the actual evaluation may be neutral or even negative. Therefore, when there are a large number of tolerant users in the dataset, we should take into account the users' harshness, and the result changes and becomes more consistent with expert evaluation.

In addition, the results of the kappa coefficient and the kendall coefficient are shown in Table 6. According to the kappa coefficient, our method has a moderate agreement with expert evaluation. And these two coefficients all show that the result of our method is closer to expert evaluation than that of the other eight methods.

4.5. Empirical study II with dataset Amazon

Evaluation of feature extraction. In this dataset, the accuracy of the corresponding feature extraction is 78.6%. The corresponding product

[3] https://www.rottentomatoes.com/

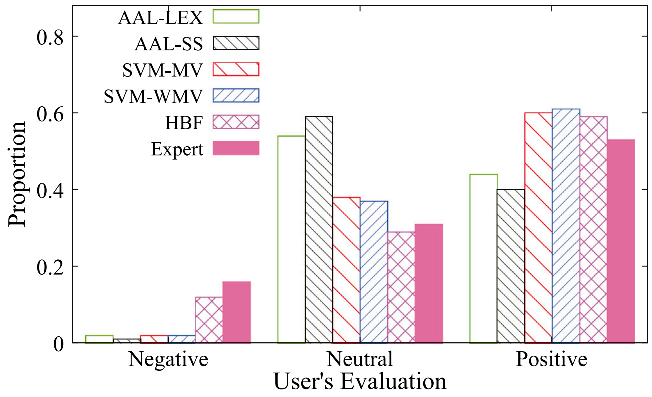


Fig. 3. Result distribution of the methods with *IMDB*.

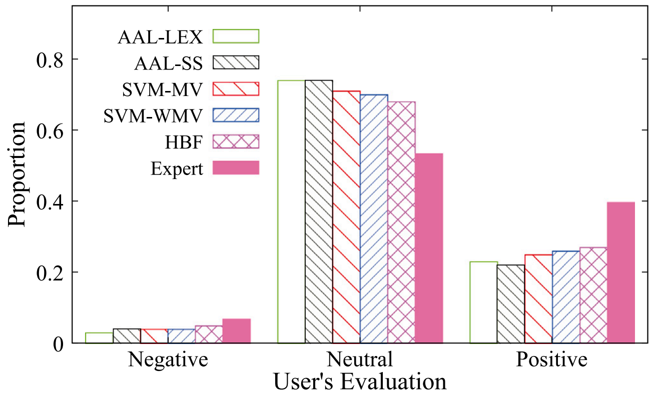


Fig. 4. Result distribution of the methods with *Amazon*.

Table 6
Kappa & Kendall coefficient with *IMDB*.

Methods	Kappa	Kendall
ComTL	25.00%	44.76%
ATAE	23.12%	31.12%
IAN	32.46%	42.05%
GCAE	27.83%	43.16%
HUB	27.37%	43.99%
HUAPA	26.92%	50.46%
AAL-LEX	35.90%	43.56%
AAL-SS	37.38%	43.59%
SVM-MV	45.17%	42.24%
SVM-WMV	46.31%	44.66%
HBF (our method)	**52.76%**	**50.66%**

Note: Best scores are in bold.

attributes are musician, instrument, melody, and style, which is consistent with the actual situation of the Amzon dataset. The aspect terms of the dataset are shown in Table 4.

Evaluation of viewpoint inference. We gathered expert evaluations of 58 music album on the AllMusic.[4] The evaluation results are shown in Table 7. First, in terms of accuracy, the *HUB* method achieves a relatively low accuracy. Because the *HUB* method deletes many comments of harsh and tolerant users and tends to obtain a negative or positive result, which is not suitable to three classifications including neutral classification. Besides, the *ATAE* method, *IAN* method and *GCAE* method are almost equal and all higher than the *ComTL* method in accuracy aspect. It conforms to the view mentioned before that the evaluation of the product is not equivalent to the evaluations of product attributes. Then, the *HUAPA* method achieves higher recall and F-measure in terms of negative and neutral, thus showing the effectiveness of user preference. The *AAL-LEX* method and the *AAL-SS* method had learned more aspects of knowledge through aspect aware learning components, resulting in more improvements. And the *AAL* component based on sentence-level supervision was superior to the dictionary-based AAL component. Next, in the music dataset, the scores are generally relatively high and the comments are more biased towards praise. Although we try to balance the dataset, the negative comments are still less in the annotated comments. Therefore most of the methods have a comparatively low F-measure in the negative aspect. Finally, our method obviously superior to other methods in terms of neutral, due to we consider users' harshness. Because many the extreme user comments in the dataset are corrected, such as positive comments of tolerant users and negative comments of harsh users.

As shown in Fig. 4, we further analyze the distribution of the result besides dataset *IMDB*. We observe that with the consideration of the users' harshness, a great number of evaluations change from positive to neutral where it is more consistent with the trend of expert evaluation than other four methods without considering harshness.

Furthermore, we verify the consistency between the results of methods and expert evaluation by using the kappa coefficient and the kendall coefficient. The results are shown in Table 8. It shows that compare with *ComTL* method, our method is more effective while the kappa coefficient grows 16.61% and the kendall coefficient grows 10.45%. However, we can find that the results are all relatively low, mainly because music comments are not as straightforward as movie comments, and machine learning methods sometimes have difficulty understanding semantics. If deep learning methods are used to obtain sentence sentiments, the semantics of sentences may be better understood.

5. Related work

Effectively using online comments for product analysis to help consumers and manufacturers make better decisions has attracted widespread attention. The product analysis method based on sentiment mining involves two phases, i.e., text analysis (Cambria, 2016; Wang & Lu, 2018) and viewpoint inference (Cambria et al., 2010; Kauffmann et al., 2019). Text analysis is mainly based on natural language processing methods to perform sentiment mining on user comments and extract sentiments expressed by users. Viewpoint inference aims at employing statistical learning methods to reasoning users' viewpoints about products and acquires product evaluations.

5.1. Text analysis

As for text analysis, it targets to analyze user sentiments from comments. Most previous research aims to analyze comments from a holistic perspective and conduct document-level sentiment analysis. However, the analysis granularity of these methods is too coarse. Therefore, researchers try to investigate aspect-level sentiment analysis. Generally, there are three kinds of user preferences in text analysis. This preference is reflected in terms of words, aspects and polarity, respectively.

Document-level sentiment analysis. Document-level sentiment mining refers to get a polarity label for each user comment. Turney (2002) presented a simple unsupervised learning algorithm that automatically constructs sentiment dictionaries for sentiment analysis. Pang et al. (2002) first adopted a supervised machine learning algorithm to predict the polarity of a comment. They trained support vector machines (SVMs), Naive Bayes and maximum entropy classifiers by using a diverse set of features. In addition, sentences are often inherently related in comments. Rao et al. (2018) proposed a neural network model with two hidden layers to exploit the semantic relations between sentences.

[4] https://www.allmusic.com/

Table 7
Experimental results on *Amazon* dataset.

Methods	Accuracy	Recall			F-score		
		Negative	Neutral	Positive	Negative	Neutral	Positive
ComTL (Idealized)	41.38	**68.80**	21.80	63.00	34.38	30.00	52.73
ATAE (2016)	47.41	6.25	20.97	90.22	9.52	32.50	58.66
IAN (2017)	47.84	20.00	69.81	47.48	28.57	41.81	57.14
GCAE (2018)	48.28	6.25	21.77	91.3	9.09	33.13	60.22
HUB (2018)	43.53	68.75	0.00	**97.83**	**45.83**	0.00	61.64
HUAPA (2018)	54.31	25.00	63.71	46.74	28.57	59.85	50.00
AAL-LEX (2019)	54.74	30.77	60.75	51.79	27.59	56.28	56.86
AAL-SS (2019)	55.17	45.45	80.39	48.24	37.04	46.86	**62.60**
SVM-MV	56.47	6.30	60.00	60.90	9.52	60.66	56.28
SVM-WMV	57.33	6.25	54.84	69.57	9.09	59.65	59.81
HBF (our method)	**61.64**	12.5	**81.5**	43.50	14.29	**70.38**	53.69

Note: Best scores are in bold.

Table 8
Kappa & Kendall coefficient with *Amazon*.

Methods	Kappa	Kendall
ComTL	8.11%	22.89%
ATAE	10.74%	14.81%
IAN	17.01%	20.09%
GCAE	12.14%	21.58%
HUB	12.95%	**36.29%**
HUAPA	14.94%	28.00%
AAL-LEX	18.92%	27.52%
AAL-SS	23.85%	28.92%
SVM-MV	19.26%	24.13%
SVM-WMV	22.43%	26.56%
HBF (our method)	**26.72%**	33.34%

Note: Best scores are in bold.

The goal of aspect-level sentiment mining is to conduct sentiment mining on each aspect category or specific aspect item, which use the topic model or machine learning method based on attention mechanisms. Jo and Oh (2011) used a probabilistic topic model to automatically discover aspects in comments with an unsupervised way. They incorporated aspect and sentiment together to model sentiments toward different aspects and discovered pairs of {aspect, sentiment}, named senti-aspects. Kim, Zhang, Chen, Oh, and Liu (2013) proposed a hierarchical topic model whose whole structure is a tree. Each node itself is a two-level tree, where the root represents an aspect and the children represent the sentiment polarities associated with it. Wang et al. (2016) adopted an attention-based long short-term memory network (ATAE) for the aspect-level sentiment classification. The model considered aspect information in two manners: (1) one was to join the aspect vector into the sentence hidden representations for computing attention weights, and (2) another one was to add the aspect vector into the input word vectors. Ma et al. (2017) believed that both aspects and sentiments should be treated specially. Thus, they employed an interactive attention network(IAN) to learn the attention in sentiments and aspects interactively. Zhu et al. (2019) employed a novel aspect-aware learning (AAL) framework to exploit the interaction between the aspect category and the contents. They designed a two-way memory network for integrating AAL into the framework of sentiment classification. Ma, Peng, Khan, Cambria, and Hussain (2018) specifically focus on leveraging commonsense knowledge in the deep neural sequential model and augment the LSTM using a stacked attentional mechanism. Akhtar, Ekbal, and Cambria (2020) proposed a stacked ensemble approach to predict the degree of intensity for emotion and sentiment by using a multi-layer perceptron network.

Furthermore, the aspect-level sentiment mining task is often accompanied by aspect extraction, whose purpose is to automatically extract aspects from comments. Aspect extraction is commonly done by using a supervised or an unsupervised approach. The unsupervised approach includes methods, such as bootstrapping-based extraction (Wang et al., 2010), syntactic rules-based extraction (Poria, Cambria, Ku, Gui, & Gelbukh, 2014), frequent pattern mining (Jeyapriya & Selvi, 2015), topic modeling (Shams & Baraani-Dastjerdi, 2017), etc. Wang et al. (2010) proposed a probabilistic rating regression model. Given a few seed words describing the aspects, a bootstrapping-based algorithm was employed to identify the major aspects and split comments. Poria et al. (2014) employed a novel rule-based method, which used commonsense knowledge and sentence dependency trees to detect explicit and implicit aspects. Jeyapriya and Selvi (2015) used frequent itemsets to mine aspects, where frequent itemset mining is used to find all frequent itemsets using minimum support count. Shams and Baraani-Dastjerdi (2017) proposed a method based on topic model. It discovered more precise aspects by incorporating co-occurrence relations as prior domain knowledge into the latent dirichlet allocation (LDA) topic model. Ray and Chakrabarti (2019) employed a seven-layer deep convolutional neural network for tagging aspects. They used a combination of deep learning methods and a set of rule-based methods to improve aspect extraction.

User preferences in sentiment analysis. Most of the above work applied a global model to all users, so it failed to recognize the preferences of different users. User preferences can be divided into word-level, aspect-level and polarity-level. User preference at the word-level means the word-using habit varies from person to person. Such as the word “good”, the majority of people use it in positive comments while a small group of people conveys the disgust by using it in a sarcasm style. Wu et al. (2018) observed comments and found that some words could display strong user preference, and some others tend to show the product features. Thus, they proposed a model (HUAPA) that encodes user and product information, respectively, into two hierarchical networks to generate two individual text representations with user attention or product attention. Gong et al. (2016) proposed a multi-task learning solution. A global sentiment model was constantly updated to capture the homogeneity of users’ opinions, while personalized models were simultaneously adapted from the global model to identify the heterogeneity of personal opinions. Yang and Eisenstein (2017) utilized social networks to follow people with the same language habits. They compressed the social network into vector representations of each author node and then incorporated user information into an attention-based neural network model, called Social Attention.

User preference at the aspect-level implies that each user cares about different attributes of the same product. For example, some people mind the plot of movies, but others pay more attention to actors or special efficacy. Wang et al. (2010) presented a model that distinguished user preference of different aspect. They computed the weights of aspects of each user, thus personal sort can be carried out. Li, Li, Kang, Yang, and Zong (2019) proposed a model called hierarchical user attention network, which integrated different kinds of user preferences. The method modeled the user preference based on user embedding and attention mechanisms.

User preference at the polarity-level refers to the preference that exists in users’s ratings. Harsh people tend to give negative comments, while tolerant people tend to provide positive ones. Gao, Yoshinaga, Kaji, and Kitsuregawa (2013) presented a model that combines user leniency and product popularity to study user preference. Gong and Wang (2018) developed a probabilistic generative model (HUB) to integrate the user comment model and social network model. Individual users were modeled as a hybrid over the instances of paired learning tasks to achieve their behavior heterogeneity, and the tasks were clustered by sharing a global prior distribution to capture the homogeneity among users.

In practice, the number of commodity reviews is relatively small and the data is sparse. In this case, the efficiency of end-to-end supervised machine learning is low, which requires a large number of training sets and parameters in order to get better results. So it is difficult to analyze commodity reviews online in the later stage, while rule-based methods is simple and practical, they has been successfully applied in emotional mining and text analysis. Therefore, in this work, we employ a rule-based method to analyze the text of the data.

5.2. Viewpoint inference

With user sentiment obtained with text analysis, product evaluations are obtained by viewpoint inference. Kauffmann et al. (2019) proposed a general product analysis framework that used natural language processing (NLP) techniques, including sentiment analysis, text data mining and clustering techniques, to obtain user sentiments for different product features. The domain ontology was used to detect the main product features. Besides, they defined a new global score for the product, which included prices, labels, two sentiment scores. However, the model did not consider user preference.

Among the researches of user comments, one of the studies is to detect malicious comments in the network by detecting the semantics in comments and characteristics of malicious users. Malicious comments refer to inflammatory, extraneous or off-topic messages. Cambria et al. (2010) first proposed the technology of automatically detecting malicious comments on the network based on sentiment analysis. They employed two sentiment mining tools to extract semantic and sentiment from web posts and used the final results to detect and filter malicious posts. Mihaylov and Nakov (2016) presented a model to examine two types of opinion manipulation trolls. They trained two logistic regression classifiers through various features such as the bag of words, metadata, sentiment, named entities, etc. Unlike malicious comments, comments given by harshness users that we consider in this work are strict about products, which do not contain malicious semantics. Thus, it cannot be detected semantically. Besides, in the existing work of product analysis, researchers regard all the comments of datasets as normal user comments.

Different from the user preferences, which make user give lenient (rigorous) evaluation for some categories of products, harshness make user given the lenient (rigorous) evaluation for all categories products. We improve sentiment analysis via incorporating harshness into viewpoint inference in this paper.

6. Conclusions

In this work, we give a product analysis method based on sentiment analysis based on the analysis of users' harshness. Specifically, considering users' harshness, we first present a general product analysis framework based on sentiment analysis. The framework includes text analysis and viewpoint inference. Second, we present a Bayesian-based text analysis method. Third, considering users' harshness, we develop a viewpoint inference method based on a probabilistic graphical model. Extensive experiments are conducted in two real-life datasets. Our framework achieves superior results when compared to the state-of-the-art methods, and superior to methods, which use the ground truth of comments to infer final evaluations without considering users' harshness.

We focus on the evaluation analysis of static information, which is beneficial to commodity research. With the growth of product evaluation information, there is a compelling need for an online real-time method of commodity evaluation. To this end, we will design an adaptive method which evaluates the products in an online form. Furthermore, we will enhance the accuracy of text analysis via improving the corpus or using neural network. More specifically, we use corpus to train TF–IDF model for distinguishing the words that have frequencies in various fields, so as to further improve the accuracy of text analysis.

CRediT authorship contribution statement

Xun Wang: Management and coordination responsibility for the research activity planning and execution. **Ting Zhou:** Writing – original draft. **Xiaoyang Wang:** Writing. **Yili Fang:** Ideas, Formulation or evolution of overarching research goals and aims.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported partly by the National Natural Science Foundation of China (No.61976187, U1609215). We would like to thank Pengpeng Chen from Beihang University for his valuable advice to improve this work. The authors would thank the anonymous reviewers for the helpful comments and suggestions to improve this paper.

References

Akhtar, M. S., Ekbal, A., & Cambria, E. (2020). How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble. *IEEE Computational Intelligence Magazine*, *15*(1), 64–75.

Baxter, J., Weaver, L., & Bartlett, P. L. Direct gradient-based reinforcement learning: II. Gradient ascent algorithms and experiments (in preparation).

Bird, S., Klein, E., & Loper, E. (2010). *Natural language processing with python*.

Borg, A., & Boldt, M. (2020). Using vader sentiment and SVM for predicting customer response sentiment. *Expert Systems with Applications*, [ISSN: 0957-4174] *162*, Article 113746.

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, *31*(2), 102–107.

Cambria, E., Chandra, P., Sharma, A., & Hussain, A. (2010). *Do not feel the trolls*. Shanghai: ISWC.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Gao, W., Yoshinaga, N., Kaji, N., & Kitsuregawa, M. (2013). Modeling user leniency and product popularity for sentiment classification. In *Proceedings of the sixth international joint conference on natural language processing* (pp. 1107–1111).

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, *69*, 214–224.

Gong, L., Al Boni, M., & Wang, H. (2016). Modeling social norms evolution for personalized sentiment classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*: *1*, (pp. 855–865).

Gong, L., & Wang, H. (2018). When sentiment analysis meets social network: A holistic user behavior modeling in opinionated data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1455–1464). ACM.

Guan, X., Li, Y., Gong, H., Sun, H., & Zhou, C. (2018). An improved SVM for book review sentiment polarity analysis. In *2018 international conference on transportation & logistics, information & communication, smart city*. Atlantis Press.

He, R., & Mcauley, J. (2016). Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In *International conference on world wide web*.

Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. In *National conference on artificial intelligence*.

Jeyapriya, A., & Selvi, C. S. K. Extracting aspects and mining opinions in product reviews using supervised learning algorithm. In *2nd international conference on electronics and communication systems* (pp. 548–552).

Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 815–824). ACM.

Jones, E., Oliphant, T., & Peterson, P. (2014). Scipy: Open source scientific tools for Python.

Kauffmann, E., Peral, J., Gil, D., Ferrández, A., Sellers, R., & Mora, H. (2019). Managing marketing decision-making with sentiment analysis: An evaluation of the main product features using text data mining. *Sustainability*, *11*(15), 4235.

Kendall, M. (1938). A new measure of rank correlation. *Biometrika*, *30*(1–2), 81–93.

Kim, S., Zhang, J., Chen, Z., Oh, A., & Liu, S. (2013). A hierarchical aspect-sentiment model for online reviews. In *Twenty-seventh AAAI conference on artificial intelligence*.

Kraft, D. (1988). *Deutsche forschungs- und versuchsanstalt für luft- und raumfahrt köln: forschungsbericht*, *A software package for sequential quadratic programming*. Wiss. Berichtswesen d. DFVLR.

Li, J., Li, H., Kang, X., Yang, H., & Zong, C. (2019). Incorporating multi-level user preference into document-level sentiment classification. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *18*(1), 7.

Liu, B., Xuan, Z., Xuan, Z., & Xuan, Z. (2015). Incorporating domain and sentiment supervision in representation learning for domain adaptation. In *International conference on artificial intelligence*.

Ma, D., Li, S., Zhang, X., & Wang, H. (2017).

Ma, Y., Peng, H., Khan, T., Cambria, E., & Hussain, A. (2018). Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis. *Cognitive Computation*, *10*(4), 639–650.

Maaten, L. vander, & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The stanford CoreNLP natural language processing toolkit. In *Association for computational linguistics, acl system demonstrations* (pp. 55–60).

Mihaylov, T., & Nakov, P. (2016). Hunting for troll comments in news community forums. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: short papers)* (pp. 399–405). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P16-2065.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, *13*(6), 47–60.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. *Vol. 10*, In *Proceedings of the ACL-02 conference on empirical methods in natural language processing* (pp. 79–86). Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Louppe, G. (2013). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(10), 2825–2830.

Poria, S., Cambria, E., Ku, L.-W., Gui, C., & Gelbukh, A. (2014). A rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media* (pp. 28–37). Association for Computational Linguistics and Dublin City University.

Rao, G., Huang, W., Feng, Z., & Cong, Q. (2018). LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*, *308*, 49–57.

Ray, P., & Chakrabarti, A. (2019). A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis. *Applied Computing and Informatics*.

Řehůřek, Radim, & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50). Valletta, Malta: ELRA.

Robb, R. A., & Jiang, H. J. (1999). Biomedical imaging, visualization, and analysis. In *International conference of the ieee engineering in medicine & biology society*.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Shams, M., & Baraani-Dastjerdi, A. (2017). Enriched LDA (ELDA): Combination of latent Dirichlet allocation with word co-occurrence analysis for aspect extraction. *Expert Systems with Applications*, *80*, 136–146.

Singh, R., & Kaur, R. (2015). Sentiment analysis on social media and online review. *International Journal of Computer Applications*, *121*(20), 44–48.

Tang, D., Bing, Q., & Liu, T. (2015). Learning semantic representations of users and products for document level sentiment classification. In *Meeting of the association for computational linguistics & the international joint conference on natural language processing*.

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417–424). Association for Computational Linguistics.

Wang, Y., Huang, M., Zhao, L., et al. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615).

Wang, B., & Lu, W. (2018). Learning latent opinions for aspect-level sentiment classification. In *Thirty-second AAAI conference on artificial intelligence*.

Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 783–792). ACm.

Whitehill, J., Wu, T.-fan, Bergsma, J., Movellan, J. R., & Ruvolo, P. L. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems* (pp. 2035–2043).

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, *39*(2–3), 165–210.

Wu, Z., Dai, X.-Y., Yin, C., Huang, S., & Chen, J. (2018). Improving review representations with user attention and product attention for sentiment classification. arXiv preprint arXiv:1801.07861.

Xue, W., & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. arXiv preprint arXiv:1805.07043.

Yang, Y., & Eisenstein, J. (2017). Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*, *5*, 295–307.

Zhu, P., Chen, Z., Zheng, H., & Qian, T. (2019). Aspect aware learning for aspect category sentiment analysis. *ACM Transactions on Knowledge Discovery from Data*, [ISSN: 1556-4681] *13*(6), http://dx.doi.org/10.1145/3350487.