A Neural Group-wise Sentiment Analysis Model with Data Sparsity Awareness

Deyu Zhou[1*] , **Meng Zhang**[1] , **Linhai Zhang**[1] , **Yulan He**[2]

[1] School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China

[2] Department of Computer Science, University of Warwick, UK

{d.zhou, m.zhang, lzhang472}@seu.edu.cn, yulan.he@warwick.ac.uk

Abstract

Sentiment analysis on user-generated content has achieved notable progress by introducing user information to consider each individual's preference and language usage. However, most existing approaches ignore the data sparsity problem, where the content of some users is limited and the model fails to capture discriminative features of users. To address this issue, we hypothesize that users could be grouped together based on their rating biases as well as degree of rating consistency and the knowledge learned from groups could be employed to analyze the users with limited data. Therefore, in this paper, a neural group-wise sentiment analysis model with data sparsity awareness is proposed. The user-centred document representations are generated by incorporating a group-based user encoder. Furthermore, a multi-task learning framework is employed to jointly model users' rating biases and their degree of rating consistency. One task is vanilla population-level sentiment analysis and the other is group-wise sentiment analysis. Experimental results on three real-world datasets show that the proposed approach outperforms some state-of-the-art methods. Moreover, model analysis and case study demonstrate its effectiveness of modeling user rating biases and variances.

Introduction

Sentiment analysis on user-generated content aims to map a given text, such as a post on Twitter or a review on Yelp, to the corresponding sentiment label or score. Traditional methods for sentiment analysis assume that the mapping between a text and a sentiment score is the same for all users. However, such an assumption is rarely true in the real world as people construct preferences based on their own experiences and express their sentiment in many difference ways, exhibiting a diverse range of rating behaviors. Thus, it is crucial to develop a sentiment analysis model considering user information.

Early approaches incorporate the user information into sentiment analysis model in a more crude way. For example, user embeddings were randomly initialized and concatenated with document embeddings before being fed into a neural network to train a sentiment classifier (Tang, Qin, and Liu 2015). Following this way, user information was incorporated into a Long-Short Term Memory (LSTM) network with attention mechanism (Chen et al. 2016). More recently, Wu et al. (2018) utilized a hierarchical neural network with user attention mechanisim to encode user information. Wang et al. (2018) proposed an adversarial cross-lingual learning framework to utilize both English and Chinese corpora for personalized microblog sentiment analysis. Amplayo (2019) investigated the influences of different ways and locations of the attribute incorporation. However, the aforementioned approaches did not consider the data sparsity issue that most users only generate limited content online. According to the study (Wojcik and Hughes 2019), most users rarely tweet, but the top 10% active users create 80% of tweets. Therefore, it is important to tackle the user data sparsity problem in sentiment analysis.

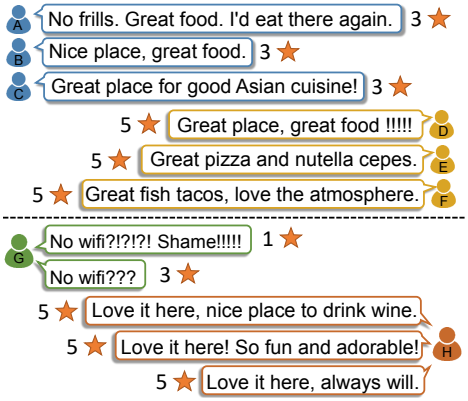

Figure 1: Example snippets of reviews posted by different users in the Yelp 2013 dataset.

Based on the observation on user-generated content, we found that some users exhibit a similar correlation pattern between their chosen words and their ratings. As shown in the upper part of Figure 1, user A, B and C all used '*great*' in their reviews, but they only gave ratings of 3 stars, showing that they seem to be more cautious about giving high rating scores. While user D, E and F also used '*great*' in their reviews, however they gave ratings of 5 stars. These examples illustrate that a sentiment analysis model trained solely on review texts is unable to capture the users' diverse rating

* Corresponding author.

behaviors. We also observed that some users are more consistent in their ratings, while others are not. In the lower part of Figure 1, user G complained about '*no wifi*' with similar words in the two reviews, but gave 1 star and 3 stars respectively, indicating that he/she is less consistent in the ratings. But for user H, the three reviews of similar content all have the same rating of 5 stars. Based on these observations, we hypothesize that: (1) while there is a general correlation pattern between word usage and rating scores, users exhibit different levels of deviation and hence different rating biases; (2) users have different degree of rating consistency, with some being more consistent in their ratings compared to others; (3) users can be categorized into different groups based on the correlation between their language usage and their rating scores as well as the degree of rating consistency. Similar idea was also stated in the theory of social comparison (Suls and Wills 1991), that humans tend to form groups with others with similar minds and abilities.

In this paper, we propose a novel neural group-wise sentiment analysis model with data-sparsity awareness. More concretely, the predicted sentiment score y is assumed to follow a Gaussian distribution $N(y^b + y^s, \sigma^2)$ where y^b is the population-level rating base score, y^s is the individual-level rating bias and σ^2 is the rating variance. y^b is learned with the vanilla population-level sentiment analysis module while y^s and σ^2 are learned with the group-wise sentiment analysis module where a set of group embeddings are employed to enhance the user-centred document representations with a group-based user encoder. Both modules are learned jointly in a multi-task learning framework. Experiments conducted on three real-world datasets, Yelp 2013, Yelp 2014 and Twitter, show that the proposed approach outperforms a number of competitive baselines.

The main contributions of this work are three-folds:

- A novel neural group-wise sentiment analysis model to address the data sparsity problem is proposed by grouping users with similar rating behaviors and leveraging the information captured in group embeddings to enhance the user-centred document representations. To the best of our knowledge, it is the first neural-network-based approach to model groups for sentiment analysis of the user-generated context.
- Rating score is modeled as a Gaussian distribution $N(y^b + y^s, \sigma^2)$ where y^b is the population-level base score, y^s is the individual-level rating bias and σ^2 is the rating variance. A multi-task learning framework is utilized to model the users' rating behavior.
- Experimental results on three real-world datasets show that the proposed approach outperforms a number of competitive baselines.

Related Work

Sentiment Analysis of User-generated Content

Previous studies have shown the effectiveness of incorporating user information into the sentiment analysis models. Song et al. (2015) utilized the latent factor model to perform sentiment classification on the microblog dataset. Wu and Huang (2016) performed sentiment classification for microblogs in a multi-task learning framework where the user-specific sentiment classifiers are trained. Gong et al. (2017) built a sentiment classification model at group level with a Dirichlet Process prior to automatically form groups. With the development of deep learning methods, deep neural networks have been applied for sentiment analysis of user-generated content and achieved notable progress. Tang et al. (2015) incorporated user embeddings into a neural network for document-level sentiment classification. Chen et al. (2016) and Wu et al. (2018) used hierarchical LSTM network to encode user information via different attention mechanism. Amplayo (2019) investigated the influences of different ways and locations to incorporate attributes into the model. However, all the aforementioned methods ignore the problem of data sparsity, which is common in real-world scenarios.

Sentiment Analysis Addressing Data Sparsity

Recently, the data sparsity problem in neural sentiment analysis has attracted attention in the research community. Akhtar et al. (2018) proposed to leverage bilingual word embeddings to mitigate the data sparsity of word representations in low-resource aspect-based sentiment analysis. Wang et al. (2018) proposed a user-attention-based neural model with an adversarial cross-lingual learning framework to overcome the shortage of personalized microblog data. Amplayo et al. (2018) tried to address the cold-start problem in sentiment classification by representing the review-sparse users with other users similar to them. Yuan et al. (2019) used a hierarchical network with user memory to learn representative users to enrich the semantic representations. Different from the above methods, we address the data sparsity problem by grouping users into different classes and leveraging the similarity among users to learn the rating behavior of different groups and perform sentiment analysis.

Methodology

Problem Setting

Given a document d written by a user u, which consists of M sentences, where each sentence c_i is composed of N_i words, the task of sentiment analysis is to predict the rating score y of the document d. To deal with this task, we propose a Neural Group-wise Sentiment Analysis Model (NGSAM) with data-sparsity awareness. In this model, we make the following assumptions:

- **Rating Bias Assumption** There exists a general correlation pattern between the language usage of sentiment expressions and the rating scores among all users (e.g., the word '*excellent*' is typically correlated with positive rating scores). The rating correlation pattern of each user is a deviation from the general correlation pattern.
- **Rating Consistency Variation Assumption** Different users exhibit different levels of rating consistency. Some users are more consistent in their rating scores (e.g., giving similar scores when using similar words in their reviews), while other show a larger variation in their rating scores.

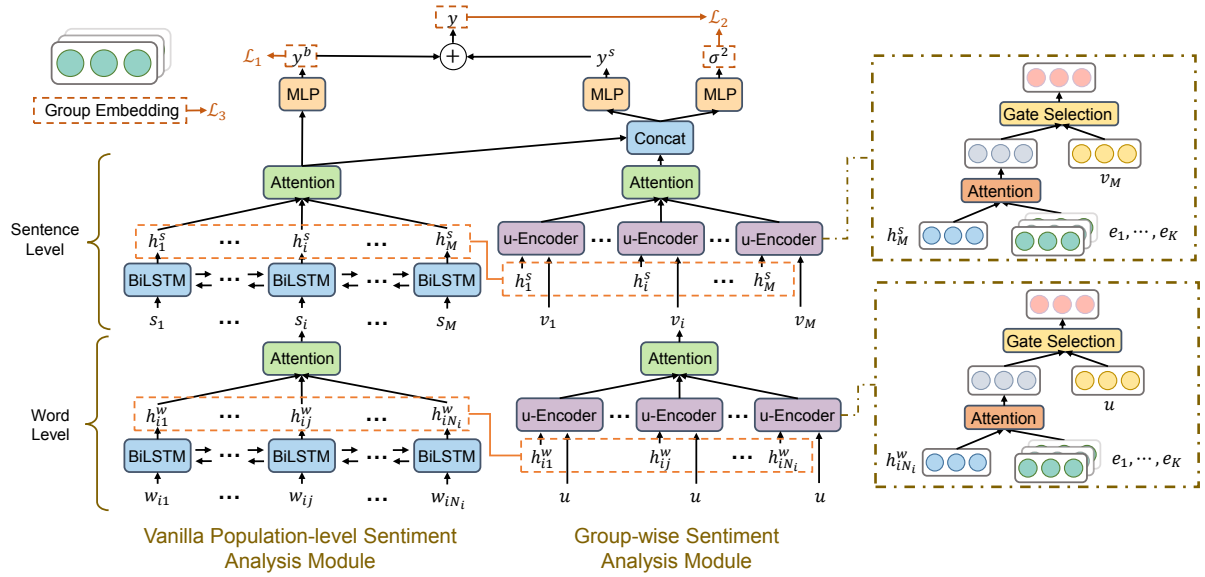

Figure 2: The architecture of NGSAM.

Notation	Description
c_i	The i-th sentence
w_{ij}	The j-th word in c_i
N_i	Number of words in c_i
K	Number of user groups
u	User embedding of the user writing d
E	Group embedding matrix, $E = \{e_1, e_2, \cdots, e_K\}$
y	Predicted rating score
$\hat{y}$	Ground-truth rating score
y^b	Predicted rating base score
y^s	Predicted rating bias
σ^2	Predicted rating variance
h^w_{ij}	Hidden state of w_{ij}
h^s_i	Hidden state of c_i
s_i	Semantic representation of c_i
g_{ij}	Group representation of w_{ij}
p_{ij}	Enhanced user-centred representation of w_{ij}
v_i	Enhanced user-centred embedding of c_i
r^b	Semantic representation of document d
r^u	Enhanced user-centred representation of d
r	Concatenated representation of d, $r = [r^b, r^u]$

Table 1: Notations used in the model.

Based on these assumptions, users are assumed to share a general rating correlation pattern, which results in a population-level base score y^b, and the individual-level rating bias y^s is a deviation from y^b. The degree of rating consistency of a user is measured by the rating variance σ^2. Thus, the predicted rating score y follows a Gaussian distribution: $y \sim \mathcal{N}(y^b + y^s, \sigma^2)$. The task of the sentiment analysis of user-generated content can be framed as the simultaneous prediction of y^b, y^s and σ^2.

The overall architecture of the proposed method is presented in Figure 2. It consists of two parts: (1) a vanilla population-level sentiment analysis module to obtain the rating base score y^b; (2) a group-wise sentiment analysis module to compute the rating bias y^s and variance σ^2. Finally, y^b and y^s are added to obtain the final output y and σ^2 is used to estimate the document-level rating scores of review documents during the optimization of the model. The notations used in the approach are shown in Table 1.

Vanilla Population-Level Sentiment Analysis Module

As a document contains a list of sentences, and each sentence consists of a list of words, a hierarchical attention network (HAN) (Yang et al. 2016) is used to obtain the semantic meaning of the document, which can capture long-distance dependencies in texts. Firstly, a word-level network is used to generate the semantic representation of each sentence based on the word embeddings. Then, these representations are fed into a sentence-level network to obtain document representations.

As for the word-level network, given a sentence $c_i = \{w_{i1}, w_{i2}, \cdots, w_{iN_i}\}$ in the document d, we first map each word to its pre-trained word embedding. Then a standard Bi-directional LSTM (BiLSTM) is employed to sequentially process each word as shown below:

$$\begin{aligned} \overrightarrow{h^w_{ij}} &= LSTM\left(\overrightarrow{h^w_{i(j-1)}}, w_{ij}\right) \\ \overleftarrow{h^w_{ij}} &= LSTM\left(\overleftarrow{h^w_{i(j+1)}}, w_{ij}\right) \end{aligned} \tag{1}$$

Then, for each word w_{ij}, the hidden state h^w_{ij} is generated by concatenating $\overrightarrow{h^w_{ij}}$ and $\overleftarrow{h^w_{ij}}$.

Since words do not contribute equally to the semantic representation of the sentence, the widely used attention mechanism (Bahdanau, Cho, and Bengio 2014) is adopted to place more weights to the words with higher importance in sentence representation learning. More concretely,

$$\begin{aligned} \varphi_{ij} &= tanh(W_b h_{ij}^w + b_b) \\ \alpha_{ij} &= softmax({\varphi_{ij}}^\top q_b) \end{aligned} \tag{2}$$

where α_{ij} is the attention weight of word w_{ij}. And the corresponding sentence representation s_i is computed as:

$$s_i = \sum_j \alpha_{ij} h_{ij}^w \tag{3}$$

As for the sentence-level network, the sentence representations $\{s_1, s_2, \cdots, s_M\}$ are fed into BiLSTM to obtain the hidden state h_i^s of each sentence s_i and the attention mechanism is further applied to obtain the semantic representation r^b in a similar way.

Finally, we use a multi-layer perceptron (MLP) with one hidden layer to predict the sentiment base score y^b using the document representation r^b as the input.

$$y^b = MLP(r^b) \tag{4}$$

Group-Wise Sentiment Analysis Module

The group-wise sentiment analysis module is proposed to predict the individual-level rating bias and variance based on user data and group information. Similar to the vanilla population-level sentiment analysis module, a word-level neural network is firstly used to generate the user-centred representation of each sentence. Then these representations are fed into a sentence-level neural network to obtain the user-centred document representation. At each layer, a user encoder is designed to enhance the user-centred embeddings based on the contextual representations as well as the corpus-level group embeddings, where the attention mechanism is employed to aggregate the user-centred representations.

Assuming that there are K user groups in the corpus, each group has a corresponding group embedding e_k, which is initialized randomly and updated during the model training process as described in (Pergola, Gui, and He 2019).

As for the word-level neural network, given the hidden state h_{ij}^w of each word w_{ij} and the user-centred embedding u for the document d, a user encoder is designed to enhance user-centred embeddings. It contains two steps: (1) calculating the group representation g_{ij} of w_{ij}; (2) generating the enhanced user-centred representation p_{ij}.

Firstly, to capture the interactions between groups and words, the attention mechanism is utilized to calculate the word-level group representation g_{ij} of w_{ij}. If the word is similar to other words used in the user group k, then its group embedding e_k will receive a high attention weight. Thus, the attention weight γ_{ijk} is calculated as the softmax output of the similarity between the hidden state h_{ij}^w and the group embedding e_k. Then, the attention weights are used to generate g_{ij}.

$$\begin{aligned} \gamma_{ijk} &= softmax(h_{ij}^\top e_k) \\ g_{ij} &= \sum_k \gamma_{ijk} e_k \end{aligned} \tag{5}$$

Secondly, since the learned user-centred embeddings can be unreliable for users with limited data, the gating mechanism is used to enhance the user-centred embeddings based on their corresponding group representations. The gate value f_{ij} is a sigmoid transformation of the concatenation of g_{ij} and u, as illustrated in the equation below,

$$\begin{aligned} f_{ij} &= \sigma(W_f[g_{ij}, u] + b_f) \\ p_{ij} &= f_{ij} \times g_{ij} + (1 - f_{ij}) \times u \end{aligned} \tag{6}$$

where p_{ij} is the enhanced user-centred representation, f_{ij} is a gated value that represents the contribution of g_{ij} to p_{ij}, and W_f and b_f are the learnable parameters. It should be pointed out that different from recurrent neural networks which sequentially calculate the hidden states of a word sequence, the proposed user encoder can compute enhanced user-centred embeddings in parallel, which greatly improve the computational efficiency.

Not all user-centred representations p_{ij} of w_{ij} contribute equally to the user-centred representation of the sentence. Therefore, the attention mechanism is also used to compute the sentence-level enhanced user-centred embedding v_i. More concretely,

$$\begin{aligned} \psi_{ij} &= tanh(W_u p_{ij} + b_u) \\ \beta_{ij} &= softmax({\psi_{ij}}^\top q_u) \\ v_i &= \sum_j \beta_{ij} p_{ij} \end{aligned} \tag{7}$$

where the weight β_{ij} is the attention weight of the user-centred representation p_{ij} and W_u, b_u and q_u are the learnable parameters.

As for the sentence-level neural network, the sentence-level enhanced user-centred embeddings $\{v_1, v_2, \cdots, v_M\}$ and the corresponding contextual hidden states $\{h_1^s, h_2^s, \cdots, h_M^s\}$ are fed into the user encoder and the attention mechanism is also used for the further generation of document-level enhanced user-centred representation r^u.

Then, the document semantic representation r^b and the enhanced user-centred representation r^u are concatenated to obtain the final document representation $r = [r^b, r^u]$. Two MLPs with one hidden layer are used to generate the rating bias score y^s and the variance σ^2 based on r.

$$\begin{aligned} y^s &= MLP(r) \\ \sigma^2 &= MLP(r) \end{aligned} \tag{8}$$

Finally, the rating base score y^b, the rating bias y^s and the rating variance σ^2 are used to obtain the final rating score y.

$$y \sim \mathcal{N}(y^b + y^s, \sigma^2) \tag{9}$$

Since $(y^b + y^s)$ is the unbiased estimation of y, it is used as the final output score y. While the rating variance σ^2 is used as a regularization term during the training to modulate the weights of samples.

Training Objective

The predictions of population-level rating scores and user-specific rating biases and variances can be considered as a multi-task problem. Therefore, a joint loss function is used as the training objective.

For the vanilla population-level sentiment analysis module, we use the mean square error (MSE) loss,

$$\mathcal{L}_1 = \frac{1}{T}\sum_t \parallel \hat{y}_t - y_t^b \parallel^2 \tag{10}$$

where T is the number of training samples, $\hat{y}_t$ is the ground-truth score and y_t^b is the predicted rating base score.

For the group-wise sentiment analysis module, we use the Gaussian penalty function, which could be decomposed into two components: (1) a residual regression that utilizes the rating variances, which helps learn the rating variances implicitly from the loss function and reduces the impact of outlier data points with extreme rating variances; (2) a variance regularization term that prevents the module from predicting too large rating variances. The group-wise loss function is defined as follows,

$$\mathcal{L}_2 = \frac{1}{T}\sum_t \frac{1}{2\sigma_t^2}\|\hat{y}_t - y_t\|^2 + \frac{1}{2}\log\sigma_t^2 \tag{11}$$

where y_t is the predicted rating score and σ_t^2 is the predicted rating variance.

Moreover, to make group embeddings more discriminative and different from each other, a penalization term is introduced,

$$\mathcal{L}_3 = ||EE^\top - I||_F^2 \tag{12}$$

where I is an identity matrix and $||\cdot||_F$ stands for the Frobenius norm of a matrix.

Finally, the combined loss function is defined as follows,

$$\mathcal{L} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 + \lambda_3\mathcal{L}_3 + \lambda_4||\Theta||^2 \tag{13}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyperparameter and $||\Theta||^2$ is a L_2 regularization item.

Experiments

In this section, we describe the datasets, evaluation metrics and implementation details, before discussing the experimental results.

Datasets: We evaluate the effectiveness of the proposed method on three real-world datasets: Yelp 2013 (Tang, Qin, and Liu 2015), Yelp 2014 (Tang, Qin, and Liu 2015) and Twitter (Go, Bhayani, and Huang 2009). Since the original datasets were curated to investigate the effect of incorporating the user information, there are enough documents for each user with the average number of documents per user around 50. To investigate the effectiveness of our model in dealing with data sparsity, we randomly draw 10 documents from each user. The statistics of three datasets are shown in Table 2.

Evaluation Metrics: Since we formalize the task as a regression problem, the root mean squared error (RMSE) and the mean absolute error (MAE) are adopted as the evaluation measures.

Dataset	Documents	Users	Labels	Avg d/u	Avg s/d
Yelp 2013	16310	1631	5	10	10.30
Yelp 2014	36130	3613	5	10	10.64
Twitter	28220	2822	2	10	3.64

Table 2: Detailed information of datasets. 'Avg d/u' means the average number of documents for each user and 'Avg s/d' means the average number of sentences in each document.

Implementation Details: We implement the models in Tensorflow2.0. Tokens which appear less than twice are filtered. The word embeddings are initialized with the pre-trained GloVe (Pennington, Socher, and Manning 2014). The dimensions of the embeddings and the hidden states are set to 300 and the batch size is 32. The number of epochs is 10. λ_1, λ_2 and λ_3 are all set to 1 and λ_4 is 0.001. The loss function is minimized using Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.001 and a dropout rate of 0.5. The numbers of user groups on Yelp 2013, Yelp 2014 and Twitter are set empirically as 10, 4 and 6 respectively. All the parameters are chosen based on the validation sets which are 10% of the training sets.

Overall Rating Prediction Results

In order to evaluate the effectiveness of the proposed method, we compare the method with the following baselines:

- **BiLSTM** employs a BiLSTM for rating predictions.
- **Bi+A** uses BiLSTM and attention mechanism for rating prediction.
- **Bi+A+u** concatenates the user embeddings with the output representations of Bi+A for prediction.
- **HAN** (Yang et al. 2016) is a hierarchical attention network which learns the document-level representation in a hierarchical manner using attention.
- **UNN** (Tang, Qin, and Liu 2015) incorporates user information into HAN.
- **HCSC** (Amplayo et al. 2018) solved the cold-start problem with shared user vectors constructed from other users.
- **RRP-UM** (Yuan et al. 2019) is the state-of-the-art approach dealing with data sparsity problem in user generated content via a hierarchical architecture with user memory to learn representative users.

We adapted HCSC[1] using only user data since the product information is not available in our datasets and re-implemented other baselines with mean square error loss. To our best knowledge, the method proposed in (Gong, Haines, and Wang 2017) is the only method that utilize group information in sentiment analysis. But we do not chose it as the baseline because: (1) it deals with binary classification (positive/negative), which is not suitable for the three datasets; (2) the implementation details are unavailable.

[1] https://github.com/rktamplayo/HCSC

Model	Yelp 2013		Yelp 2014		Twitter	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
BiLSTM	0.8029	0.6086	0.7847	0.6186	1.6564	1.3493
Bi+A	0.7837	0.6112	0.7687	0.6142	1.6385	1.3960
Bi+A+u	0.7857	0.6130	0.7655	0.6105	1.5726	1.1630
HAN	0.7370	0.5739	0.7353	0.5774	1.5984	1.2271
UNN	0.7705	0.6064	0.7312	0.5753	1.5272	**1.0636**
HCSC	0.7447	0.5844	0.7241	0.5615	1.5839	1.1914
RRP-UM	0.7404	0.5798	0.7397	0.5837	1.5538	1.1818
NGSAM	**0.7163**	**0.5466**	**0.7031**	**0.5503**	**1.5011**	1.1569

Table 3: Overall rating prediction results.

Model	Yelp 2013		Yelp 2014		Twitter	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
NGSAM	**0.7163**	**0.5466**	**0.7031**	**0.5503**	**1.5011**	**1.1569**
w/o bias	0.7268	0.5631	0.7099	0.5670	1.5890	1.2379
w/o var	0.7396	0.5800	0.7211	0.5670	1.6011	1.2436
w/o $\mathcal{L}_3$	0.7368	0.5717	0.7177	0.5727	1.5917	1.2231

Table 4: Ablation study results of NGSAM. 'w/o' means 'without.

The experimental results on three datasets are presented in Table 3. It can be observed from Table 3 that: (1) among the models without the user information (BiLSTM, BiLSTM-Att and HAN), HAN gives the best results, showing the effectiveness of using the hierarchical structure to capture the semantic information in documents; (2) models with concatenated user embeddings (BiLSTM+ATT+u, UNN) perform worse than their corresponding counterparts without the user information (BiLSTM+Att, HAN) on Yelp 2013 and Yelp 2014, though better on Twitter, indicating that simply concatenating user embeddings does not consistently perform well when facing with the data sparsity problem; (3) NGSAM performs remarkably better than all the baselines on three datasets, demonstrating the effectiveness of modeling users' rating behaviours with the enhanced user representations based on the shared group embeddings to deal with data sparsity problem.

Ablation Study

In order to analyze the contributions of various components of NGPAM, we perform the following ablation study: **NGSAM(w/o bias)** without modeling the rating bias, **NGSAM(w/o var)** without modeling the rating variance, and **NGSAM(w/o $\mathcal{L}_3$)** without the loss function of group embeddings.

Experimental results on three datasets are shown in Table 4. It can be observed that the performance drops more significantly without modeling rating variance followed by the removal of the group embedding loss function. The removal of the component for modeling rating bias has the least impact on the model performance. Overall, NGSAM with all three components gives the best results.

We also investigate the performance of the proposed NGSAM with different group numbers. Figure 3 shows the performance of NGSAM on Yelp 2013. It can be observed that, in general, as the number of the groups grows, RMSE and MAE continue to decline, showing that the performance of NGSAM is influenced by the number of groups. When the number of groups is beyond 10, both RMSE and MAE converge, indicating that the model has captured the characteristics of different groups.

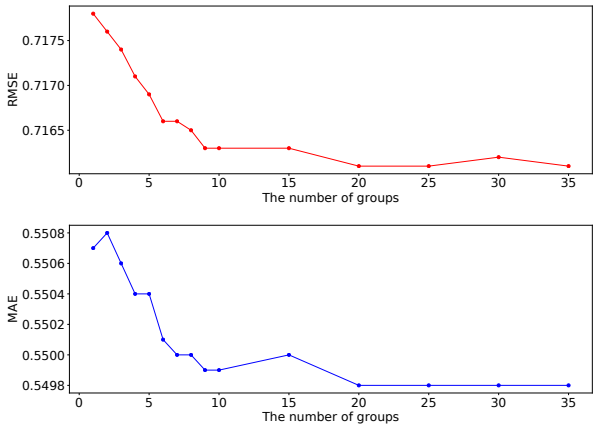

Figure 3: Performance with different group numbers. Top: RMSE. Bottom: MAE.

Model	30 d/u		20 d/u		10 d/u	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Bi+A+u	0.7181	0.5515	0.7415	0.5773	0.7857	0.6130
UNN	0.6764	0.5321	0.6802	0.5342	0.7705	0.6064
HCSC	0.6703	0.5212	0.6746	0.5270	0.7447	0.5844
RRP-UM	0.6976	0.5499	0.7022	0.5400	0.7404	0.5798
NGSAM	**0.6690**	**0.5162**	**0.6692**	**0.5207**	**0.7163**	**0.5466**

Table 5: Performance on different sparse datasets of Yelp 2013. 'd/u' means the number of documents per user.

Comparisons on Datasets with Different Sparsity Levels

Table 5 shows the performance of BiLSTM+Att+u, UNN, HCSC, RRP-UM and NGSAM on the Yelp 2013 datasets with different levels of data sparsity. As shown in the table, in general, as more data become available, the performance of all the models improves. NGSAM performs the best among the competing models on all datasets across all the sparsity settings. We also notice that the performance gap between NGSAM and the baselines increases as the number of documents per user decreases, showing the superior capability of NGSAM on dealing with data sparsity.

Visualization of User-centred Document Representations

To provide an insight into the characteristics of the learned user-centred representations of the review documents, we visualize the enhanced user-centred representations of documents in the Yelp 2013 test set with the global rating base scores, rating biases and variances respectively in Figure 4. Since these representations are high-dimensional vectors, we used t-SNE (Maaten and Hinton 2008) to map them into a 2D space. Each user-centred document representation

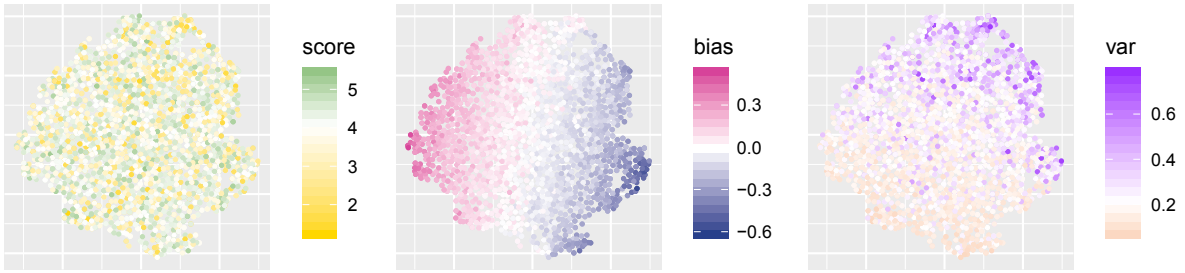

Figure 4: The visualization result of user-centred document representations. Left: global rating base score. Middle: rating bias. Right: rating variance.

Text	User	$\hat{y}$	y^b	y	σ^2
[D1] Everyone looks for that little hole in the wall that turns out to be a gem in the rough. This isn't it. It's closer to rough than gem. Food is iffy at best. Service is borderline. They have good hours and the chips are relatively fresh so all is not a loss.	A	2	2.68	**2.16**	**0.30**
[D2] Excellent crust, quality ingredients, nice beer selection, and the pizza is always cooked perfectly.	A	4	4.57	**3.99**	**0.12**
[D3] This is a good, easy-going burger place, very friendly, good food... A good alternative to just go and relax, and eat yummy burgers without the high price.	B	3	4.02	**3.46**	**0.31**
[D4] Consistent food, nice family atmosphere and spicy pickles. What more could you want?	C	3	4.01	**3.45**	**0.31**
[D5] Now we know how it feels to be scalped. Anyone who thought this was a great event is a member of the tribe.	D	1	3.87	**3.76**	**0.84**

Figure 5: Case study of rating bias on Yelp 2013. '$\hat{y}$' is the ground-truth. 'y^b' is the predicted rating score of the vanilla sentiment analysis module. 'σ^2' is the predicted rating variance. 'y' is the predicted rating score of the group-wise sentiment analysis module.

is represented as a point in the same position of three subfigures. It can be seen from the figures that: (1) the global rating base scores are uniformly distributed in the space, showing that the rating base scores are not user-specific and instead capture the overall rating behaviors across all users; (2) documents with similar rating biases or variances tend to be closer, which verifies the effectiveness of modeling users into groups with different rating biases and variances; (3) in the point cloud of rating biases, the bias values show a left-to-right decreasing trend while in the point cloud of rating variances, the values show a top-to-bottom decreasing trend. These results show that rating biases and variances are unrelated to each other and can be modeled separately.

Case Study

To further investigate the meaningfulness of modeling rating biases and variances, we present five examples in Figure 5. As the documents are long, we only present their summaries which are constructed manually. It can be observed from the figure that: (1) [D1] contains both positive and negative comments and the overall actual rating is 2 stars. The vanilla sentiment analysis module gives a predicted score of 2.68, but the group-wise module is able to further reduce the score to be closer to the ground-truth; (2) In [D2], user A expressed a high appreciation with the words '*excellent*', '*quality*', '*nice*' and '*perfectly*', but only gave 4 stars. It shows that user A tends to be more cautious about giving high rating scores, which is also captured by the group-wise module; (3) The review texts in [D3] and [D4] are mostly positive containing the words '*nice*', '*good*' and '*friendly*', but with 3 stars, which are much lower than population-level sentiment scores. The vanilla sentiment analysis module outputs the scores of 4.02 and 4.01 respectively. However, the group-wise module captures the personal characteristics, and modifies them to 3.46 and 3.45, which are closer to the ground-truth; (4) [D5] is difficult for sentiment analysis since it is short and contains a positive word '*great*', though it expressed a negative feeling, so our model predict a higher variance for this sample. These results show that users indeed exhibit different rating behaviors and modeling rating biases and variances is an effective way for sentiment analysis of user-generated content.

Conclusions

In this paper, we have proposed a novel approach to capture different rating behaviors of users for sentiment analysis of user-generated content. Specifically, a set of group embeddings is introduced and a group-based user encoder is designed to enhance the user-centred document representations for users with limited review data. Moreover, users rating biases and the degree of rating consistency are explicitly modeled with the group-enhanced representation in a multi-task framework in order to differentiate user groups with different rating patterns. Experimental results demonstrate the effectiveness of our method. In the future, we will extend the model to utilize more information to form groups.

Acknowledgments

This work was funded by the National Key Research and Development Program of China (2016YFC1306704), the National Natural Science Foundation of China (61772132), Innovate UK (grant no. 103652), the EPSRC (grant no. EP/T017112/1, EP/V048597/1) and a Turing AI Fellowship funded by the UK Research and Innovation (UKRI) (grant no. EP/V020579/1). The authors would like to thank the anonymous reviewers for the insightful comments.

References

Akhtar, M. S.; Sawant, P.; Sen, S.; Ekbal, A.; and Bhattacharyya, P. 2018. Solving data sparsity for aspect based sentiment analysis using cross-linguality and multilinguality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 572–582.

Amplayo, R. K. 2019. Rethinking Attribute Representation and Injection for Sentiment Classification. In *EMNLP*.

Amplayo, R. K.; Kim, J.; Sung, S.; and Hwang, S.-w. 2018. Cold-Start Aware User and Product Attention for Sentiment Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2535–2544. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1236. URL https://www.aclweb.org/anthology/P18-1236.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Chen, H.; Sun, M.; Tu, C.; Lin, Y.; and Liu, Z. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1650–1659.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1(12): 2009.

Gong, L.; Haines, B.; and Wang, H. 2017. Clustered model adaption for personalized sentiment analysis. In *Proceedings of the 26th International Conference on World Wide Web*, 937–946. International World Wide Web Conferences Steering Committee.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Pergola, G.; Gui, L.; and He, Y. 2019. TDAM: A topic-dependent attention model for sentiment analysis. *Information Processing & Management* 56(6): 102084.

Song, K.; Feng, S.; Gao, W.; Wang, D.; Yu, G.; and Wong, K.-F. 2015. Personalized sentiment classification based on latent individuality of microblog users. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Suls, J. E.; and Wills, T. A. E. 1991. *Social comparison: Contemporary theory and research.* Lawrence Erlbaum Associates, Inc.

Tang, D.; Qin, B.; and Liu, T. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1014–1023.

Wang, W.; Feng, S.; Gao, W.; Wang, D.; and Zhang, Y. 2018. Personalized microblog sentiment classification via adversarial cross-lingual multi-task learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 338–348.

Wojcik, S.; and Hughes, A. 2019. Sizing up Twitter users. *Washington, DC: Pew Internet & American Life Project. Retrieved May* 1: 2019.

Wu, F.; and Huang, Y. 2016. Personalized microblog sentiment classification via multi-task learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1480–1489.

Yuan, Z.; Wu, F.; Liu, J.; Wu, C.; Huang, Y.; and Xie, X. 2019. Neural Review Rating Prediction with User and Product Memory. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2341–2344.