Leveraging Personalized Sentiment Lexicons for Sentiment Analysis

Dominic Seyler, Jiaming Shen, Jinfeng Xiao, Yiren Wang and ChengXiang Zhai
{dseyler2,js2,jxiao13,yiren,czhai}@illinois.edu
University of Illinois at Urbana-Champaign

ABSTRACT

We propose a novel personalized approach for the sentiment analysis task. The approach is based on the intuition that the same sentiment words can carry different sentiment weights for different users. For each user, we learn a language model over a sentiment lexicon to capture her writing style. We further correlate this user-specific language model with the user's historical ratings of reviews. Additionally, we discuss how two standard CNN and CNN+LSTM models can be improved by adding these user-based features. Our evaluation on the Yelp dataset shows that the proposed new personalized sentiment analysis features are effective.

KEYWORDS

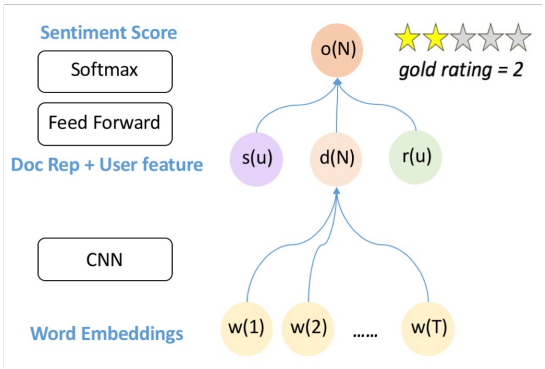
sentiment analysis, sentiment lexicons, personalization

ACM Reference Format:
Dominic Seyler, Jiaming Shen, Jinfeng Xiao, Yiren Wang and ChengXiang Zhai. 2020. Leveraging Personalized Sentiment Lexicons for Sentiment Analysis. In *Proceedings of the 2020 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '20), September 14–17, 2020, Virtual Event, Norway*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3409256.3409850

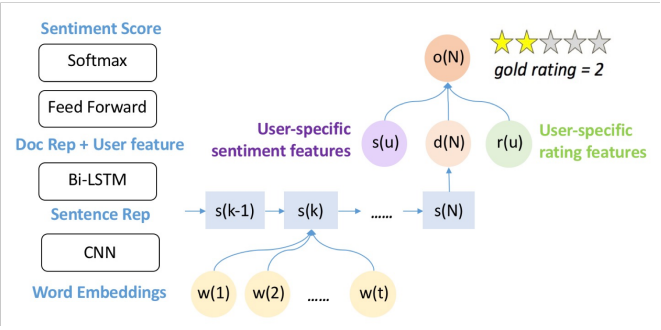
1 INTRODUCTION

Sentiment analysis has seen lots of attention in research due to its large benefit to downstream applications, such as recommender systems [15, 16], social media analysis [4] and e-commerce websites [24]. In sentiment analysis, the system has to predict the sentiment of an input text. This is usually done by either predicting a coarse sentiment label, or a probability score for each sentiment dimension. User reviews, such as Yelp reviews, are an indirect statement of sentiment and can therefore be used as a proxy for the sentiment analysis paradigm. More concretely, in the case of rated user reviews the model receives the textual review and has to predict the user's review score.

Since user ratings are highly subjective, it is natural to incorporate personalization into the model, which enables us to learn a user-specific predictive function. Along with text, the model takes user profiles as input and returns multi-level sentiment scores. The user profile includes documents previously written by the same author and can be utilized to infer user traits related to her writing style. As a result, a personalized sentiment model can help improve recommendations or search results for a user.



(a) Personalized CNN model.



(b) Personalized CNN-LSTM model.

Figure 1: Model architectures for personalized sentiment analysis.

Based on the intuition that sentiment words carry different sentiment weight for different users, we leverage an external sentiment lexicon for personalization. Here, the model is personalized by learning a user-specific unigram language model over the most prominent words in a sentiment lexicon. Similar to Gao et al. [11], we try to estimate the general rating behavior of a user, which has been shown to be a strong signal for predicting future ratings. However, we capture a different set of statistical rating-based features and use them in combination with a personalized language model, which is not present in [11]. For example, a certain person might use the word "good" very generously, therefore it does not carry much value in predicting the sentiment. However, a more conservative person might use "good" only sporadically, thus, it has great predictive value for high-scoring reviews. We find that

ICTIR '20, September 14–17, 2020, Virtual Event, Norway

ACM ISBN 978-1-4503-8067-6/20/09…$15.00
https://doi.org/10.1145/3409256.3409850

the information of sentiment words can be enriched with signals derived from each user's rating habit. We therefore exploit rating history in addition to sentiment lexicons.

We propose and design general personalized sentiment lexicon features and study how to effectively incorporate them into a learning model. Our model learns personalized sentiment lexicons and rating patterns as complementing features to common text-based features. Since neural network models have proven effective for sentiment analysis [7, 10, 12, 22, 25, 27], we show that our features boost performance in two neural models. However, how to integrate these features into a neural network is non-trivial. We accomplish this by leveraging the intuition from wide and deep learning in Cheng et al. [8], and concatenate review representations with our personalization features. When these personalization components are added to our models, they outperform other personalized baselines that use user history. Thus, our contributions are as follows:

(1) We propose novel personalized sentiment features derived from a user-specific unigram language model over an external sentiment lexicon.
(2) We show that the personalized sentiment features improve a deep learning framework, when integrated in addition to a user's rating history.
(3) We compare our model to other (deep) personalized baselines and find that our models perform up to 1.3% better.
(4) We create a dataset with user history, based on the Yelp 2018 dataset.

2 RELATED WORK

The incorporation of sentiment lexicons for the sentiment analysis task is common [6, 13, 17, 23]. For example, Teng et al. [25] uses the context of sentiment words in a neural model to make sentiment predictions. However, none of the current methods explicitly models a user's use of sentiment words. Personalization is usually done implicitly by grouping user reviews and trying to learn some hidden user representation [10, 12, 27]. For example, Tang et al. [22] represents users and products as separate feature vectors, where similar users and products are closer in the embedding space. Chen et al. [7] combines user and product information as attentions over these different levels for personalization. In contrast, our models capture user specificity by explicitly modeling the user's writing style of expressing sentiment using a personalized sentiment lexicon.

3 OUR APPROACH

As in previous work [7, 22], we map the problem of sentiment analysis to predicting the score of user reviews on Yelp. The prediction task is as follows: given an input text (i.e., the review) and a user profile (i.e., a list of previous reviews by the same user), the model has to predict the review score. In what follows, we discuss our proposed user-specific features and our methods of representing the textual input. Figure 1 illustrates our model architectures.

3.1 User-Specific Features

Sentiment Language Model Features (lm): Sentiment words are important triggers for review rating classification. We use a unigram language model on two constructed sentiment lexicons (positive and negative) to capture user-specific language features. For sentiment lexicon construction, we build a sentiment word candidate list derived from the SentiWordNet dictionary [5]. The candidate list is filtered with a sentiment score threshold: $score^{pos}(word) > \epsilon$ or $score^{neg}(word) > \epsilon$ (we empirically choose $\epsilon = 0.3$), and a threshold for absolute difference between the positive and negative score: $|score^{pos}(word) - score^{neg}(word)| > \epsilon^{diff}$ (we empirically choose $\epsilon^{diff} = 0.5$). We learn a user-specific language model with the vocabulary based on the sentiment lexicon using maximum-likelihood estimation. Notice that the vocabulary size is usually large ($\sim$ 4,000 in our case). We therefore use truncated-SVD to compress the user-specific language model into a low-dimensional space. Finally, each user u is represented by a dense vector $s(u) \in \mathbf{R}^k$ where k is the number of largest singular values selected in the truncated SVD ($k = 100$ in our model).

Rating Features (rating): The user's rating history is another useful signal for sentiment analysis [11]. Therefore, we obtain additional user-specific features based on her rating statistics, including maximum, minimum, mean score, standard deviation and the histogram of the user's rating history. We denote the user-specific feature vector as $r(u)$.

3.2 Text Representations

In this paper, we focus on studying the impact of adding personalized features to neural networks, though those features can also be incorporated into other machine learning models. Our model uses a hierarchical neural network model to encode the review text into a low dimensional vector space. At the word level, we use word embeddings for word representation. We use two different architectures for the review representations:

CNN: Similar to Kim [18], our CNN model obtains the review document representation directly from the word level (w_i) representation:

$$d(N) = CNN(w_1, w_2, ..., w_T).$$

CNN-LSTM: A hierarchical model, where at sentence level we adopt a CNN to embed each sentence in the review into a dense vector. The sentence representations are then passed into a LSTM layer to generate document level representations:

$$s(k) = CNN(w_{k1}, w_{k2}, ..., w_{kt}), d(N) = LSTM(s(1), ..., s(N)).$$

For each sentence vector $s(i)$, the LSTM model will output a vector $o(i)$. We use max pooling[1] over these N output vectors for constructing the final document representation:

$$d(N) = max\{o(1), ..., o(N)\}.$$

Inspired by Cheng et al. [8], the text representation, user sentiment language features and user rating features are concatenated and fed into a fully connected layer, which is passed into a softmax layer for classification. The personalized models with both review representations and user-specific features are denoted as **P-CNN** and **P-CNN-LSTM**.

[1]We test mean pooling, direct sum of output vectors $s(i)$, and the attention mechanism from Yang et al. [28]. Max pooling gives the best performance.

Dataset	#Reviews	#Users
Total	89,150	1,950
Train. (2011-2015)	50,018	1,950
Dev. (2016)	21,120	1,950
Test (2017)	18,012	1,950

Table 1: Statistics of the Yelp18-U dataset.

4 EVALUATION

4.1 Dataset

Currently available datasets on sentiment analysis (e.g., [9, 22]) contain a random sample of users and are split into training, development and testing. There are no guarantees that users in the training dataset will also be included in development and testing. In order to study the effect of personalization, the underlying dataset has to have user history available[2]. We therefore create our own dataset that abides by the previously mentioned characteristics.

We use the 2018 version of the Yelp dataset [1], which includes reviews about businesses, as well as meta-data about users and businesses. We find that in the dataset, the level of user activity varies greatly. While most users are moderately active, there are some users with over a thousand reviews. To make sure our training and evaluation are not dominated by these very active users, we restrict our analysis to users with 20-200 reviews.

Table 1 shows the characteristics our dataset (*Yelp18-U*), where the following two conditions are satisfied: (1) Ensure that all users in the development and test datasets have been observed in the training dataset. (2) The total number of reviews in the training set should be close to 50k. Using this process we sample 1,950 users with 50,018 reviews during 2011-15. Those reviews were used as training data and a development and test dataset are formed by taking reviews written by those users in 2016 and 2017, respectively.

4.2 Experimental Setup

In our experiments we answer the following research questions: (1) we measure the effectiveness of our personalization features by performing an ablation study. (2) we compare our model's performance with other personalized methods:

Semantic Representations for Users and Products (UPNN) [22]: We use the source code provided by the authors [2] to run their models on our dataset and use the provided settings for hyper-parameters and GloVe's [21] 200-dimensional word embeddings.

Neural Sentiment Classification (NSC) [7]: We use the source code provided by the authors [3]. We use the standard settings for hyper-parameters and the pre-trained word embeddings that are provided with the source. There are three versions: NSC is the basic implementation, NSC+LA uses local semantic attention and NSC+UPA leverages user product attention.

We evaluate all models using classification accuracy, which is a hard mapping from the predicted label to the sentiment class and reflects the practical performance of a system. For our models, we apply grid search to find the optimal parameter settings that maximize accuracy on the development set, which are: dropout=0.5, kernel-number=200, learning-rate=1e-3 and batch-size=32. We train word2vec [20] word embeddings on the Yelp dataset. User-specific features are estimated on the training set only. Thus, as long as the user is seen in the training data, the method can leverage the personalized features directly. If a user is previously unseen, their personalized feature values would be zero.

[2]This setup avoids the cold-start problem, however the focus of our work is on improving personalization, which requires historical user information.

Model	Accuracy
CNN	66.37%
P-CNN	67.14% (+1.2%)
– rating feature (lm-only)	66.77% (+0.6%)
– lm feature (rating-only)	67.09% (+1.1%)
CNN-LSTM	66.15%
P-CNN-LSTM	67.26% (+1.7%)
– rating feature (lm-only)	66.04% (-0.2%)
– lm feature (rating-only)	67.04% (+1.3%)

Table 2: Feature ablation with improvement using personalization in parenthesis.

4.3 Feature Ablation Analysis

We investigate the effectiveness of our proposed personalized features by studying the performance in isolation and combined. Table 2 shows the results for our models on the Yelp18-U dataset. The models CNN and CNN-LSTM have no personalization component and perform worse than the personalized models. Surprisingly, CNN can outperform CNN-LSTM, even though it does not explicitly obtain a sentence representation, but the personalized CNN-LSTM model outperforms the personalized CNN model. The improvement of the lm features alone is smaller than the rating features (for P-CNN-LSTM it actually hurts performance). However, when lm is added to the rating features it is still able to improve overall, as they boost performance for both P-CNN and P-CNN-LSTM models. To this end, we conclude that both personalized features are effective and the highest additive benefit can be achieved when combined.

4.4 Comparison to Baselines

In Table 3, we compare our models to the baselines. We find that our CNN and CNN-LSTM models achieve comparable results with the NSC model, even though the NSC model has a more complex architecture (i.e., hierarchical LSTM). Adding user features improves performance for the baseline NSC+LA/UPA models and our P-CNN(-LSTM) models. Both of our personalized models outperform the best personalized baseline model. In addition, the P-CNN-LSTM model can slightly better leverage the user features (1.3% improvement), compared to the P-CNN model (1.1% improvement). Even though our models make use of a user's history, it does not entail that these models cannot handle unseen users. In a real-word setting, the model's performance would fall within the boundaries of the personalized and non-personalized models, depending on how many users with or without history are observed.

4.5 Case Study

We perform a case study and investigate individual examples in our data that have a large difference between the predicted and the actual review score. We find that most examples fall within one of the following categories:

Model	Accuracy
UPNN (Tang et al. [22])	59.11%
NSC (Chen et al. [7])	66.06%
NSC+LA (Chen et al. [7])	66.17%
NSC+UPA (Chen et al. [7])	66.42%
CNN	66.37% (+0.5%)
P-CNN	67.14% (+1.1%)
CNN-LSTM	66.15% (+0.1%)
P-CNN-LSTM	**67.26%** (+1.3%)

Table 3: Improvement over best non-personalized baseline (NSC) for CNN and CNN-LSTM and best personalized baselines (NSC+UPA) for P-CNN and P-CNN-LSTM.

Misinterpreted sarcasm. Reviews that use positive sentiment words in an ironic way can mislead classification, e.g., "*wow! the carne asada taco tastes like cheap steak.*".

Times are changing. Customers of a business state that their service used to be great, but now it's bad, e.g., "*Everything started out awesome with UNK UNK then things changed [...]*".

Disaster avoided. Reviewers spend most of the review complaining about something that went wrong but give a good score: "*15 minutes for a cold sandwich when no one is in front of you is too long! [...] Thank you Kim for making things right!*".

In these extreme cases, we find that there is often a part of the review that sounds extremely negative or positive while the review as a whole has the opposite rating.

5 CONCLUSION AND FUTURE WORK

We presented a novel personalized approach for sentiment analysis that captures a user's specific use of sentiment words in a language model and correlates them with her rating behavior. We showed that incorporating personalized sentiment lexicons can improve overall performance and beat other personalized baselines.

For future work, we will explore more rigorous ways to derive sentiment words to close the gap between manually curated sentiment dictionaries and the informal language used in social media. One promising direction is to look at the nearest neighbors of sentiment seed words in an embedding space, trained on informal text. Our results show great promise for using personalized features (*lm* and *ratings*) to improve a sentiment analyzer even if added in a straightforward way. It's reasonable to believe that with more sophisticated incorporation of such personalized features, we could achieve an even larger improvement, which would be an interesting future direction. Furthermore, we plan to test our models on datasets of other domains, e.g, [14, 19, 26].

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1801652.

REFERENCES

[1] [n.d.]. https://www.yelp.com/dataset/
[2] [n.d.]. http://ir.hit.edu.cn/~dytang/paper/acl2015/UserTextNN.zip
[3] [n.d.]. https://github.com/thunlp/NSC
[4] Cuneyt Gurcan Akcora, Murat Ali Bayir, Murat Demirbas, and Hakan Ferhatosmanoglu. 2010. Identifying breakpoints in public opinion. In *Proceedings of the first workshop on social media analytics*. 62–66.
[5] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the International Conference on Language Resources and Evaluation*.
[6] Erik Cambria. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems* 31, 2 (2016), 102–107.
[7] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1650–1659.
[8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.
[9] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 193–202.
[10] Zi-Yi Dou. 2017. Capturing User and Product Information for Document Level Sentiment Analysis with Deep Memory Network. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 521–526.
[11] Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Modeling user leniency and product popularity for sentiment classification. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 1107–1111.
[12] Alberto Garcia-Duran, Roberto Gonzalez, Daniel Onoro-Rubio, Mathias Niepert, and Hui Li. 2020. Transrev: Modeling reviews as translations from users to items. In *European Conference on Information Retrieval*. 234–248.
[13] Marco Guerini, Lorenzo Gatti, and Marco Turchi. 2013. Sentiment Analysis: How to Derive Prior Polarities from SentiWordNet. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1259–1269.
[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.
[15] Minqing Hu and Bing Liu. 2004. Mining Opinion Features in Customer Reviews. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*. 755–760.
[16] Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. 2009. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. 57–64.
[17] Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*. 1367–1373.
[18] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1746–1751.
[19] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the Association for Computational Linguistics*. 142–150.
[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[21] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1532–1543.
[22] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the Association for Computational Linguistics*. 1014–1023.
[23] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the Association for Computational Linguistics*. 1555–1565.
[24] Huifeng Tang, Songbo Tan, and Xueqi Cheng. 2009. A survey on sentiment detection of reviews. *Expert Systems with Applications* 36, 7 (2009), 10760–10773.
[25] Zhiyang Teng, Duy Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing*. 1629–1638.
[26] Yiren Wang, Dominic Seyler, Shubhra Kanti Karmaker Santu, and ChengXiang Zhai. 2017. A Study of Feature Construction for Text-based Forecasting of Time Series Variables. In *Proceedings of the ACM on Conference on Information and Knowledge Management*. 2347–2350.
[27] Zhen Wu, Xin-Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. Improving Review Representations With User Attention and Product Attention for Sentiment Classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 5989–5996.
[28] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. 1480–1489.