Modeling User Leniency and Product Popularity for Sentiment Classification

Wenliang Gao***, Naoki Yoshinaga**†**, Nobuhiro Kaji**† **and Masaru Kitsuregawa**†‡
*Graduate School of Information Science and Technology, The University of Tokyo
†Institute of Industrial Science, The University of Tokyo
‡National Institute of Informatics
{*wl-gao, ynaga, kaji, kitsure*}*@tkl.iis.u-tokyo.ac.jp*

Abstract

Classical approaches to sentiment classification exploit only textual features in a given review and are not aware of the personality of the user or the public sentiment toward the target product. In this paper, we propose a model that can accurately estimate the sentiment polarity by referring to the *user leniency* and *product popularity* computed during testing. For decoding with this model, we adopt an approximate strategy called "two-stage decoding." Preliminary experimental results on two real-world datasets show that our method significantly improves classification accuracy over existing state-of-the-art methods.

1 Introduction

Document-level sentiment classification estimates the sentiment polarity for a given subjective text (hereafter, review). Traditionally, researchers have tried to estimate the sentiment polarity from only the textual content of the review (Pang and Lee, 2004; Li et al., 2011). However, since reviews are written by a user to express his/her emotion toward a particular product, taking the users and products into consideration would play an important role in solving this task.

Recently, the increase of opinionated text within social media, e.g., *Twitter*, has motivated researchers to exploit the user or product information in the sentiment classification task. Some researchers take advantages of the friend relation in a social network because friends are likely to hold common tastes (Tan et al., 2011; Seroussi et al., 2010; Speriosu et al., 2011). Others incorporate user- or product-specific n-gram features (Li et al., 2011; Seroussi et al., 2010). Although these studies have showed that user or product information is useful for sentiment classification, they implicitly assume that the same users or products appear in both training and testing data. Thus, to train such a model, a large amount of the reviews should be labeled for each user and each product. In a real-world scenario, however, this is unrealistic since new users and products are ceaselessly emerging and labeling reviews written by such users (or on such products) is impractical.

In the real world, different users have different rating standards, while different products receive different rating tendencies. For example, a critical person is likely to point out flaws and gives negative ratings, while a popular product receives more praise than negative feedback. We refer to these user- or product-specific polarity biases as user leniency and product popularity, respectively. A sentiment classifier would resort to these biases when textual features are not reliable enough to estimate the sentiment polarity.

In this study, we build a model that automatically computes and uses user leniency and product popularity for sentiment classification. We represent these biases with two types of real-valued global features. Because these features and the labels of the test reviews mutually depend on each other, it is challenging to globally optimize a configuration of polarity labels for a given set of reviews. We here adopt a two-stage decoding strategy (Krishnan and Manning, 2006) for resolving the mutual dependencies in our model.

We evaluated our method on two real-world datasets (Blitzer et al., 2007; Maas et al., 2011). Experimental results demonstrated that the proposed method significantly improved the classification accuracy against the state-of-the-art methods (Dredze et al., 2008; Seroussi et al., 2010).

The remainder of this paper is organized as follows. We first discuss some related work in Section 2. We describe our method in Section 3. We then report experimental results in Section 4. Finally, we conclude our study in Section 5.

International Joint Conference on Natural Language Processing, pages 1107–1111,
Nagoya, Japan, 14-18 October 2013.

2 Related Work

Recently, social media such as *Twitter* has attracted much attention from researchers because it is now apparently the major source of subjective text on the Web. The traditional text-based methods, such as Pang *et al.* (2002), could not easily handle such short and informal text (Jiang et al., 2011).

Tan *et al.* (2010) and Speriosu *et al.* (2011) exploited the user network behind a social media website (*Twitter* in their case) and assumed that friends give similar ratings towards similar products. Seroussi *et al.* (2010) proposed a framework that computes users' similarity on the basis of their usage of text and their rating histories. They then classify a given review by referring to ratings given for the same product by other users who are similar to the user in question. However, such user networks are not always available in the real world.

Li *et al.* (2011) incorporate user- or product-dependent n-gram features into a classifier. They argue that users use a personalized language to express their sentiment, while the sentiment toward a product is described by product-specific language. This approach, however, requires the training data to contain reviews written by test users and written for test products. This is infeasible since labeling reviews requires too much manual work.

3 Method

Given a set of reviews, $\mathcal{R}$, our task is to estimate label $y_r \in \{+1, -1\}$ for each review, $r \in \mathcal{R}$, with estimation function $g(\boldsymbol{x}_r)$:

$$g(\boldsymbol{x}_r) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_r, \tag{1}$$

$$y_r = \begin{cases} +1 & if\ g(\boldsymbol{x}_r) > 0 \\ -1 & otherwise \end{cases},$$

where $\boldsymbol{x}_r$ is r's feature vector and $\boldsymbol{w}$ is the weight vector.

3.1 Idea

Our interest is to exploit user leniency and product popularity to improve sentiment classification. We encode each of them into two real-valued global features, which are detailed in Section 3.2. Since these global features depend on the labels of the input reviews, we cannot independently estimate the labels of reviews. We then discuss a decoding strategy in Section 3.3.

Note that we assume to know which reviews are written by the same user and which are written on the same product. This assumption is realistic nowadays since user information is available in many real-world datasets (Blitzer et al., 2007; Pang and Lee, 2004), while product information can be extracted from text if not available (Qiu et al., 2011). We should emphasize here that our method does not require user profiles, product descriptions, or any sort of extrinsic knowledge on the users and products.

3.2 Features

The review r's feature vector, $\boldsymbol{x}_r$, is composed of local features ($\boldsymbol{x}_r^l$) and global features ($\boldsymbol{x}_r^g$), such that $\boldsymbol{x}_r = (\boldsymbol{x}_r^l, \boldsymbol{x}_r^g)$. In this study, we use word n-grams ($n = 1, 2$) in the textual content of the review as local features, while we encode the user leniency and product popularity into global features. We introduce four global features to capture the user leniency and product polarity:

$$\boldsymbol{x}_r^g = \{f\_u^+, f\_u^-, f\_p^+, f\_p^-\},$$

where the first two features, f_u^+ and f_u^-, represent the user leniency as the ratio of positive and negative reviews written by the same user of r, while the other two features, f_p^+ and f_p^-, represent the product popularity as the ratio of positive and negative reviews on the same product of r. The global features are thereby computed as:

$$f\_u^+(r) = \frac{|\{r_j \,|\, y_j = +1, r_j \in N_u(r)\}|}{|N_u(r)|},$$
$$f\_u^-(r) = \frac{|\{r_j \,|\, y_j = -1, r_j \in N_u(r)\}|}{|N_u(r)|},$$
$$f\_p^+(r) = \frac{|\{r_j \,|\, y_j = +1, r_j \in N_p(r)\}|}{|N_p(r)|},$$
$$f\_p^-(r) = \frac{|\{r_j \,|\, y_j = -1, r_j \in N_p(r)\}|}{|N_p(r)|},$$

where $N_u(r)$ represents a set of reviews written by the same user as r and $N_p(r)$ represents a set of reviews written for the same product as r, respectively:

$$N_u(r) = \{r' \,|\, u_r = u_{r'} \wedge r \neq r'\},$$
$$N_p(r) = \{r' \,|\, p_r = p_{r'} \wedge r \neq r'\}.$$

3.3 Decoding

Because global features are computed for each user or product, we want to process as many test reviews at once so that they include many reviews for each user or on each product to compute reliable global features. However, because the possible ways of assigning labels to a given set of reviews, $\mathcal{R}$, is $2^{|R|}$ and the two types of global features introduce complex label dependencies to be resolved, exact decoding is computationally expensive even with dynamic programming. In this study, we thus resort to an approximate decoding strategy called "two-stage decoding" (Krishnan and Manning, 2006). It splits the decoding process into a local decoding stage and a global decoding stage. Each stage takes linear time with respect to the number of reviews processed. This strategy is thereby scalable to a larger number of test reviews.

At the first stage, all the global features are set to 0, and only local features are used to classify the reviews. In the second stage, labels estimated in the first stage are used to compute the values of the global features. The labels are then revised by using both local and global features. In our case, the two-stage decoding at first uses only word n-gram features to estimate the labels of reviews. The estimated labels are used to compute user leniency features and product popularity features. Then, the decoding revises the labels considering both the word n-gram features and the user leniency and product popularity features.

3.4 Training

We train a binary classifier as the score estimation function in Eq. 1, considering word n-gram features, user leniency features, and product popularity features. The values of global features are computed by using the gold labels. We assume that a value of the user leniency feature or product popularity feature for a review whose user has no other reviews or whose product has no other reviews is set to 0.

4 Experiments

We evaluated our method in terms of accuracy on two real-world datasets (Blitzer et al., 2007; Maas et al., 2011) for a document-level sentiment classification task.

For each review, we at first use OpenNLP[1] to detect sentence boundaries and tokenize each sentence in order to obtain word n-gram features. Following Pang *et al.* (2002)'s settings, we take negation (such as *n't* and *cannot*) into consideration. Because features with low frequency are unreliable, any n-gram that appears less than six times in the training data are ignored.

Dataset	Blitzer	Maas
No. of reviews	188,350	50,000
No. of users	123,584	n/a
No. of products	101,021	7,036
No. of reviews/user	1.5	n/a
No. of reviews/products	1.9	7.1

Table 1: Dataset statistics.

We adopted a confidence weighted linear classifier (Dredze et al., 2008) as our binary classifier. This is because it has been reported to perform best on the sentiment classification task (Dredze et al., 2008).

4.1 Datasets

We used two datasets that were developed by Blitzer *et al.* (2007) and Maas *et al.* (2011). The datasets contain user/product and only product information. The statistics of the two datasets are summarized in Table 1.

The original Blitzer dataset contains more than 780,000 reviews (88% positive, 12% negative), which were collected from amazon.com across several domains, such as books, movies and games. We automatically delete reviews written by the same user on the same product, which results in about 740,000 reviews. Then, the reviews are balanced for positive and negative labels (94,175 reviews for each) to maintain consistency with the setting in other existing works.

The Maas dataset has 25,000 positive and 25,000 negative reviews on movies. The dataset provides a URL for each review, which represents the sentiment target, a movie. We thus use the URL as a unique identifier for the movie. The user information cannot be fully recovered, so we only model the product dependency on this dataset.

Our method performs best when the reviews written by/on the same user/product are in the same set (training or testing) since we can compute more reliable global features when we have more reviews written by/on the same user/product. In the two datasets, reviews were originally ordered by user or product. To prevent a seemingly unfair accuracy gain under this particular splitting, we randomly shuffled the reviews and performed

[1] `http://opennlp.apache.org/`

Method	**Accuracy (%)**	
	Blitzer	**Maas**
Seroussi *et al.* (2010)	89.37	n/a
Maas *et al.* (2011)	n/a	88.89^3
baseline	90.11	91.35
proposed	$91.01^{>}$	$92.68^{\gg}$

Table 2: Accuracy on review datasets. Accuracy marked with "$\gg$" or "$>$" was significantly better than baseline ($p < 0.01$ or $0.01 \leq p < 0.05$ assessed by McNemar's test).

a 2-fold cross-validation.

4.2 Results

In this section, we report the accuracy of our sentiment classifier. Accuracy is measured as the number of correctly classified reviews divided by the number of all the reviews. We prepared two baseline classifiers to see the advantage of our classifier. As one baseline, we used a confidence-weighted linear classifier (Dredze et al., 2008) that takes only textual features into account. As another baseline, we implemented a user similarity-based method proposed by Seroussi *et al.* (2010).[2] The similarity of users is computed by using a word n-gram Jaccard distance (called "AIT" in Seroussi *et al.* (2010)). When the user of an input review is unseen in the training data, a default classifier, which is trained with all the training reviews, is used to classify the review.

Table 2 shows the experimental results. The proposed method significantly improved the classification accuracies across the two datasets. A larger improvement was acquired on the Maas dataset because the average number of reviews for each product in the dataset was larger than that in the Blitzer dataset.

Impact of size on test reviews In our method, since global features play a key role, acquiring more reliable global features is our major concern to make the improvement more significant.

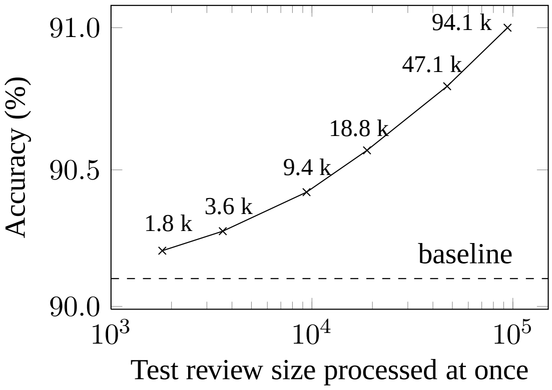

Figure 1: Accuracy when we changed the size of test reviews processed at once by our classifier.

We thus performed 2-fold cross-validation with the same splitting for the Blitzer dataset, while changing the size of test reviews processed at once to investigate the impact of test review size on classification accuracy. In this experiment, we split the test reviews into equal-sized smaller subsets and applied our classifier independently to each of the subsets.

As shown in Figure 1, when we processed a larger number of test reviews at once, the accuracy increased. This result confirms our expectations.

5 Conclusion

We presented a model that captures and uses user leniency and product popularity for sentiment classification. Different from the previous studies that are aware of the user and product of the review, our model does not require the training data to contain reviews written by the test users or written on the test products. To infer labels under our proposed model, we investigated a two-stage decoding strategy.

We conducted experiments on two real-world datasets to demonstrate the effectiveness of our proposed method. The method performed more accurately than did the baseline method, which only uses n-gram features, and an existing user-aware approach. We also showed that processing more test reviews at once lead to better accuracy.

We plan to publish our code and datasets.[4] A detailed exploration of this work will be reported in Gao *et al.* (2013).

[2] We built user-specific classifiers for users who wrote reviews with positive polarity and negative polarity more than a pre-specified threshold. After several trials, the threshold was set to be 5 to gain the best performance.

[3] This result was computed under a different splitting from ours. Under Maas *et al.* (2011)'s splitting, the accuracy for the baseline and proposed method was 90.83% and 92.29%. The main difference between our baseline and their method is the features. They use only unigram features, while we use unigram and bigram as features. Using only unigram features under their splitting, the accuracy of the baseline method was 87.8%.

[4] `http://www.tkl.iis.u-tokyo.ac.jp/~wl-gao/`

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, pages 440–447, Prague, Czech Republic.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of ICML*, pages 264–271, New York, NY, USA.

Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Collective sentiment classification based on user leniency and product popularity. In *Proceedings of PACLIC*, Taipei, Taiwan. to appear.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings ACL-HLT*, pages 151–160, Portland, Oregon, USA.

Vijay Krishnan and Christopher D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of COLING-ACL*, pages 1121–1128, Sydney, Australia.

Fangtao Li, Nathan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. 2011. Incorporating reviewer and product information for review rating prediction. In *Proceedings of IJCAI*, pages 1820–1825, Barcelona, Catalonia, Spain.

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 ACL-HLT*, pages 142–150, Portland, Oregon, USA.

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, pages 271–278, Stroudsburg, PA, USA.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2010. Collaborative inference of sentiments from texts. In *Proceedings of UMAP*, pages 195–206, Berlin, Heidelberg.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of EMNLP, workshop on Unsupervised Learning in NLP*, pages 53–63, Edinburgh, UK.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of KDD*, pages 1397–1405, New York, USA.