

Sentiment Classification of Social Media Text Considering User Attributes

Junjie Li^{♣♣}, Haitong Yang[♦], Chengqing Zong^{♣♣}

[♣] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences

[♦] University of Chinese Academy of Sciences, Beijing, China

[♣] School of Computer, Central China Normal University, Wuhan 430079, China
{junjie.li,cqzong}@nlpr.ia.ac.cn, htyang@mail.ccnu.edu.cn

Abstract. Social media texts pose a great challenge to sentiment classification. Existing classification methods focus on exploiting sophisticated features or incorporating user interactions, such as following and retweeting. Nevertheless, these methods ignore user attributes such as age, gender and location, which is proved to be a very important prior in determining sentiment polarity according to our analysis. In this paper, we propose two algorithms to make full use of user attributes: 1) incorporate them as simple features, 2) design a graph-based method to model relationship between tweets posted by users with similar attributes. The extensive experiments on seven movie datasets in Sina Weibo show the superior performance of our methods in handling these short and informal texts.

1 Introduction

With the rapid development of social media, more and more people express their opinions in the web, such as Twitter, Sina Weibo, etc. To automatically mine public opinions for business marketing or social studies, sentiment classification has attracted much attention [11,7].

Following [12], lots of researches use machine learning algorithms to build sentiment classifier and their approaches work well on formal texts. However, these methods usually perform poorly when handling social media text. Because these texts are often short and contain many informal words (like ‘cooooo!’). To alleviate this problem, researchers focus on two kinds of methods. On one hand, they try to employ sophisticated features, such as emoticons [8] and character ngrams [9]. On the other hand, some studies [4,15,3] explore the effects of user interactions (such as following and retweeting) on sentiment classification.

Despite the success of these approaches, they typically only consider user interactions and ignore demographics information such as age, gender, location, etc(also called user attributes). After considering user attributes, we can not only improve sentiment classification accuracy of these informal texts, but also mine opinions about products by different attribute groups(such as male or age:’19-30’) of consumers. Specifically, we find these attributes can provide lots of information to determine the polarity of social media text, which contains:

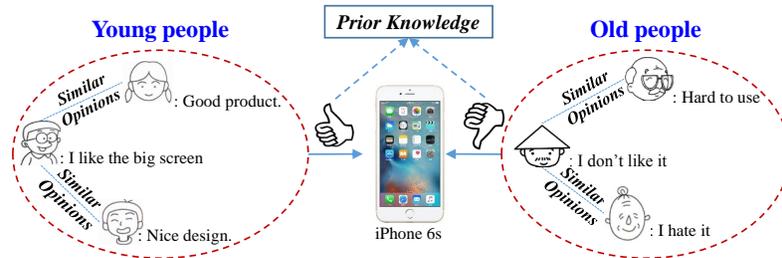


Fig. 1. An illustration to explain *Prior Knowledge* and *Similar Opinions*.

Table 1. Dataset statistics. Movie Name: movie name of the dataset. (N_+, N_-) : number of positive and negative tweets. l : average number of words per tweet. Movie Type: different types of movie.

Dataset	Movie Name	(N_+, N_-)	l	Movie Type
MH	Monster Hunt	(292, 292)	28.94	Comedy, Fantasy
TT4	Tiny Times 4.0	(397, 397)	31.95	Love
SS	Silent Separation	(596, 596)	22.82	Love
FY	Forever Young	(611, 611)	31.13	Love
FOT	Fleet Of Time	(479, 479)	25.48	Love
MCDTM	Monk Comes Down The Mountain	(505, 505)	29.27	Comedy, Love
AHON	A Hero Or Not	(369, 369)	25.70	Comedy, Fantasy

- *Prior Knowledge*: User attributes can provide some prior knowledge about the polarity. For the same product, like iPhone 6s in Figure 1, people with different attributes may hold different opinions. Young people may like the product since it is beautiful and runs smoothly, while old people may give negative comments to it because it is hard to operate.
- *Similar Opinions*: People with similar attributes may have similar backgrounds and possess similar opinions to the same product. Two young people may give similar (positive) comments to iPhone 6s, while two old people may both dislike it.

Therefore, it is feasible to leverage these attributes to build a smarter sentiment classifier and achieve better performance. To take *Prior Knowledge* and *Similar Opinions* into consideration, we propose two strategies: 1) take them as simple features, 2) design a graph-based model to encode relations between tweets posted by users with similar attributes.

We evaluate our methods on seven movie datasets from Sina Weibo¹, which is the largest Chinese microblogging service. Compared with existing content-based methods, the two strategies we proposed can improve average classification accuracy by 1.9 percent and 1.0 percent respectively. When we combine them together, we can get the best results which outperform the baseline by 2.2 percent on average.

¹ <http://weibo.com/>

In sum, our contributions in this paper are twofold. First, we propose two strategies to effectively capture *Prior Knowledge* and *Similar Opinions* and integrate them into a graph-based model (§3). Second, in order to stimulate further research on this direction, we make our datasets (§2) consisting of 6,498 movie reviews with reviewers’ attribute information publicly available.

2 Datasets

2.1 Data

Our datasets are made up of tweets from the movie special column² of Sina Weibo, in which users can post comments about movies. The statistics about our datasets are given in Table 1. Not only we crawl tweets from Sina Weibo, we have also crawled all available information about users who post the tweets, including their following relationships and public profiles like age and gender. Each tweet is rated by the users from Star-1 to Star-5. The tweets with Star-1 and Star-2 are labeled as *Negative*, and those with Star-4 and Star-5 are labeled as *Positive*. From Table 1, we can find the average length of tweets is very short (around 28 words), which accords with the characteristics of social media text. *All the data used in our experiments will be made available.*

2.2 User Attributes

Table 2. List of user attributes and the overall percentage of each attribute-value in all datasets.

Attribute	Values(Percentage)
Gender	male(29.75%), female(70.25%)
Age	1-18(19.26%), 19-30(37.34%), 31-45(3.09%), 45+ (0.27%), NULL (40.04%)
Location	abroad(3.28%), first-tier city ³ (13.75%), second-tier city(24.25%), third-tier city(27.73%), fourth-tier(17.82%), NULL(13.17%)
Fan	true(40.58%), false(59.42%)

We collected four kinds of user information in Sina Weibo: *gender*, *age*, *location* and *fan*, in which *fan* indicates whether the user is fan of the main actors or actresses and can be obtained easily from users’ follow list. To quantitatively measure these attributes, we have further discretized them into different bins and the details of user attributes are shown in Table 2. Since we have four dimensions in user attributes, we utilize a quadruple to represent attribute information of a

² <http://movie.weibo.com>

³ We divide all cities in China into different grades according to their economic level. For example, the first-tier city contains Beijing, Shanghai, etc

user and call it attribute quadruple. If we don't collect any value in a dimension, we use 'NULL' to represent it. For example, we can use a quadruple (male, 1-18, abroad, true) to represent a 16 years old boy, who is the fan of the main actress and lives abroad.

From Table 2, we can find that users always keep their age privacy and don't fill in age (the filling rate in age is about only 60%), while 100% users write gender and about 87% users provide location. Among all users, female and young account for a significant proportion (70.25% and 56.6%).

3 The Proposed Method

In this section, we propose two methods to model *Prior Knowledge* and *Similar Opinions* and a combination strategy to merge them together. In the following, we introduce these methods respectively.

3.1 Some Notations

For clear illustration, some notations are given. Suppose our dataset D has n tweets. For each tweet t_i , we collect its content d_i , attribute quadruple of its owner u_i and its sentiment label y_i . So the dataset D can be formalized: $D = \{(t_i, d_i, u_i, y_i)\}_{i=1}^n$. $c \in \{pos, neg\}$ denotes the sentiment label that is to be predicted by classification methods.

3.2 Content-based Method

The content-based method only uses tweet content and it computes the probability of a label c being assigned to a tweet t_i as follows:

$$p(c|t_i) = p(c|d_i) \quad (1)$$

in which $p(c|d_i)$ can be computed by any generative or discriminative model with content features.

3.3 Feature-based Method

In this subsection, we propose a feature-based method to consider *Prior Knowledge*.

When computing the probability of c to t_i , we not only consider tweet content d_i , but also incorporate attribute quadruple of its owner u_i . Its formulation is as follows:

$$p(c|t_i) = p(c|d_i, u_i) \quad (2)$$

in which $p(c|d_i, u_i)$ can also be computed by any generative or discriminative model with the combination of content features and user attributes features. We take all attribute-values in Table 2 as binary features and treat these features as User Attributes Features (UAF).

Algorithm 1 Pruning

Input: old UAG(oUAG), pruning parameter λ , train dataset D ;**Output:** new UAG(nUAG)

```

1: allAGList = ConstructAllAttrGroups()
2: hcAGList  $\leftarrow$   $\emptyset$ 
3: for each group  $g \in$  allAGList do
4:   posPer  $\leftarrow$  ComputePosPerForGroup( $D, g$ )
5:   negPer  $\leftarrow$  1 - posPer
6:   if  $|\text{posPer} - \text{negPer}| \geq \lambda$  then
7:     hcAGList  $\leftarrow$  hcAGList  $\cup$   $g$ 
8:   end if
9: end for
10: nUAG  $\leftarrow$  oUAG
11: for each edge  $e \in$  nUAG do
12:   d1, d2 = getNodeAttributeInfo( $e$ )
13:   if  $d1 \notin$  hcAGList or  $d2 \notin$  hcAGList then
14:     delete  $e$  from nUAG
15:   end if
16: end for

```

3.4 Graph-based Method

We design a graph-based method to incorporate *Similar Opinions*. First, a graph called User Attributes Graph(UAG) is constructed according to user similarity, in which we connected tweets posted by similar users. Then, we use an iterative method to infer the graph.

Now, we present the details on constructing UAG:

1. **Connecting tweets posted by similar users:** The idea of constructing UAG is to connect tweets posted by similar users, because similar users may have similar opinions. We use a similarity score to measure the similarity of any two users, which can be obtained by computing the number of same value (except for 'NULL') in their attribute quadruple. If the similarity score of two users is higher than half of the number of all dimensions in user attributes (the value is 2 since we collect 4 dimensions in Table 2), we call they are similar users and connect tweets posted by them. For example, the similarity score of two users, whose attribute quadruples are (male, 1-18, abroad, true) and (female, 1-18, abroad, true), is 3. Because their *age*, *location* and *fan* are same and are not 'NULL'. Thus, we connect their tweets.
2. **Pruning:** In the construction of UAG, we connect all tweets posted by similar users. However, this may bring some noises into our graph because similar users don't have to hold the same sentiment exactly. Therefore, to improve sentiment consistency of all edges in UAG, we propose a simple pruning strategy. The detailed pruning process is shown in Algorithm 1. Firstly we build all attribute groups by traversing any combinations of attribute-value in Table 2 (line:1). Attribute groups can contain one dimension such as (Gender:male) or the combination of different dimensions such as (Fan:true \cap Gender:female \cap

Age:1-18). Secondly, we need to mine some high consistency attribute groups through pruning parameter λ , where users in these groups tend to express the same opinions with high probability (line: 2-9). Thirdly, we remove the edge in UAG, whose nodes' owner (users) are not in these high consistency attribute groups (line: 10-16). We also test the influences of different λ for our method in §4.3, and we set λ to 0.7.

After UAG being constructed, we design a graph-based method for sentiment classification of tweets. In our model, G means UAG, $N(t_i)$ represents the neighborhood of t_i in G and $l(t_i)$ denotes the label of tweet t_i . When computing the probability of c to t_i in G , we make the Markov assumption that the determination of sentiment polarity can only be influenced by either the content of the tweet d_i or sentiment assignments of neighbor tweets $t_k \in N(t_i)$. Thus we get Equation 3.

$$p(c|t_i, G) = p(c|d_i, N(t_i)) \quad (3)$$

After applying the additional independence assumption that there is no direct coupling between the content of a document and the labels of its neighbors and using $l(N(t_i))$ to represent a specific assignment of sentiment labels to all immediate neighbors of the review t_i , we get Equation 4.

$$p(c|d_i, N(t_i)) = p(c|d_i) \times \sum_{l(N(t_i))} p(c|l(N(t_i)))p(l(N(t_i))) \quad (4)$$

We can convert the output scores of a review by the content-based method into probabilistic form and use them to approximate $p(c|d_i)$, which is a base classifier to the graph-based method. Then a relaxation labeling algorithm described in [2] can be used on the graph to iteratively estimate $p(c|t_i, G)$ for all reviews. After the iteration ends, for any review in the graph, the sentiment label that has the maximum $p(c|t_i, G)$ is considered the final label.

3.5 Combination Strategy

To merge *Prior Knowledge* and *Similar Opinions* together, we improve the graph-based method by adding u_i when computing c to t_i :

$$p(c|d_i, N(t_i)) = p(c|d_i, u_i) \times \sum_{l(N(t_i))} p(c|l(N(t_i)))p(l(N(t_i))) \quad (5)$$

The only difference between Equation 4 and Equation 5 is the base classifier. In Equation 4, the base classifier is computed by $p(c|d_i)$ and only takes content information to decide the label. However in Equation 5, we can add user demographics u_i into the content-based method and utilize $p(c|d_i, u_i)$ to build the base classifier.

4 Experiments

4.1 Experimental Settings

We evaluate the proposed methods on seven datasets introduced in Section 2. In our experiments, tweets in each dataset are randomly split up into five folds (with four folds serving as training data and the remaining one fold serving as test data). All of the following results are reported in terms of an averaged accuracy of five-fold cross validation. We compare our model with content-based sentiment classification methods:

- 1) NB: We implement the Naïve Bayes Classifier based on a multinomial event model.
- 2) ME: Maxent Entropy⁴ is a classic discriminative model and widely used in sentiment classification.
- 3) SVM: Support Vector Machine is a also widely used baseline method to build sentiment classifier. LibSVM⁵ toolkit is chosen as the SVM classifier. The penalty parameter is set as 0.1.

Following the standard experimental settings in sentiment classification, we use term presence as the weight of feature, and evaluate two kinds of features, 1) *ui*: unigrams, 2) *bi*: both unigrams and bigrams. The paired *t*-test [20] is performed for significant testing with a default significant level of 0.05.

4.2 Performance Comparison

Table 3. Classification accuracy of baseline systems. The best results are in **bold**.

Methods	MH	TT4	SS	FY	FOT	MCDTM	AHON	Average
NB- <i>ui</i>	0.8750	0.8387	0.7961	0.8740	0.8507	0.7941	0.8495	0.8397
NB- <i>bi</i>	0.8904	0.8236	0.8071	0.8765	0.8508	0.7733	0.8617	0.8405
SVM- <i>ui</i>	0.8444	0.7682	0.7777	0.8265	0.8319	0.7495	0.8102	0.8012
SVM- <i>bi</i>	0.8410	0.7657	0.7819	0.8331	0.8246	0.7366	0.8129	0.7994
ME- <i>ui</i>	0.8358	0.7796	0.7819	0.8314	0.8319	0.7554	0.7899	0.8008
ME- <i>bi</i>	0.8409	0.7745	0.7802	0.8405	0.8226	0.7386	0.8130	0.8015

Table 3 reports the classification accuracy of baseline systems. On one hand, we can find that NB gets the best results in our datasets. Some researchers [10,17] showed that NB is better than SVM when the training set is small or texts are short. Our datasets satisfy the two conditions, thus, it is not surprising that NB obtains better performance. On the other hand, adding bigram features always improve the performance. Thus, we choose NB-*bi* as our baseline system.

⁴ http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

⁵ <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Table 4. Classification accuracy of our methods. *Base*: the baseline system (NB-*bi* in Table 3). *Base+UAF*: Adding user attribute features into the baseline. *Base+UAG*: Using *Base* as the base classifier to construct graph-based model. *Base+UAF+UAG*: Using *Baseline+UAF* as the base classifier to construct graph-based model. The best results are in **bold**.

Methods	MH	TT4	SS	FY	FOT	MCDTM	AHON	Average
Base	0.8904	0.8236	0.8071	0.8765	0.8508	0.7733	0.8617	0.8405
Base+UAG	0.8990	0.8488	0.8138	0.8822	0.8601	0.7792	0.8699	0.8504
Base+UAF	0.9008	0.8690	0.8246	0.8854	0.8633	0.7921	0.8820	0.8596
Base+UAF+UAG	0.9059	0.8753	0.8255	0.8887	0.8664	0.7931	0.8848	0.8628

The results of our methods are shown in Table 4. From the results, we can get the following observations. Firstly, after user attribute as feature added into the baseline system, we get 1.9 percent improvements on average, which indicates that user attributes are very useful for sentiment classification and taking user attribute as features can be a good supplement to content features. Secondly, after encoding relations between tweets in our graph-based method, we can outperform the baseline system by 1.0 percent on average, which shows the effectiveness of our graph-based method. Lastly, after the two strategies is integrated, we achieve the best performance, which surpasses the baseline system by 2.2 percent on average and is significant according to the paired *t*-test.

4.3 Effects of Pruning

In the process of building UAG, we propose a pruning strategy and set pruning parameter λ to 0.7. To further investigate the need of the pruning strategy and the sensitivity of graph-based method to the pruning parameter λ , we give the experiment results in Table 5 and plot the sentiment classification accuracy with pruning parameter λ from 0.0 to 1.0 on our datasets in Figure 2.

Before Pruning V.S After Pruning

Table 5. *edgeNum*, *conProb* and classification accuracy in UAG before and after pruning.

UAG		MH	TT4	SS	FY	FOT	MCDTM	AHON	Average
Before Pruning	edgeNum	31,636	65,926	136,141	171,950	102,513	85,506	45,151	91,260
	conProb	0.8281	0.8728	0.7772	0.8173	0.7825	0.7623	0.7818	0.8031
	accuracy	0.8871	0.8438	0.8121	0.8707	0.8528	0.7644	0.8780	0.8441
After Pruning	edgeNum	13,037	31,340	4,477	7,408	11,427	1,327	5,309	10,618
	conProb	0.9558	0.9588	0.8635	0.9856	0.9985	0.9726	0.9849	0.9600
	accuracy	0.8990	0.8488	0.8138	0.8822	0.8601	0.7792	0.8699	0.8504

In theory, our graph-based method is influenced by two important factors: edge number of the graph (*edgeNum*) and the probability of sentiment consistency

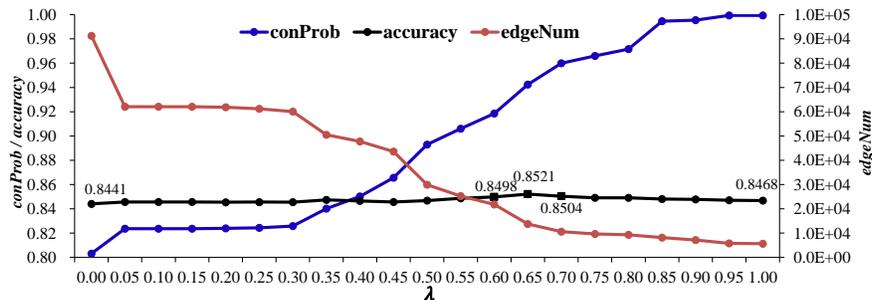


Fig. 2. average *conProb*, average *edgeNum* and average accuracy in our datasets when varying λ .

(*conProb*) of all edges in the graph. More *edgeNum*, and higher *conProb* will result in better performance. Thus, we give statistics about the two factors of graph before pruning and graph after pruning in Table 5.

From Table 5, we can find before pruning, the UAG graph contains 91,260 edges average and the average *conProb* is only 0.8031. After pruning, although *edgeNum* drops to 10,618, *conProb* rises greatly to 0.96. Just as stated before, the two factors (*edgeNum* and *conProb*) have great effects on the graph-based model. But we think compared with *edgeNum*, *conProb* is more important because a lot of inconsistent edges may cause many noises. Finally, after the pruning strategy, our model improve the average accuracy by 0.6 percent.

Sensitivity to different pruning parameter

From Figure 2, we can find when λ equals to 0.0 (it means there is no pruning in constructing UAG), *edgeNum* reaches the maximum, *conProb* gets the minimum and the accuracy is worst, which means many inconsistent edges in UAG hurt the performance. As λ increases, we add pruning in building UAG and delete many noisy edges in UAG, get higher *conProb* and better performance. The curves of accuracy always reach the peak when λ is around 0.65. When we continue to increase λ , the performance begin to decrease. In this case, *conProb* is at a high value (average *conProb* is higher than 0.94) and can insure edges in UAG are mostly consistent. As λ increases, we can get higher *conProb*. Meanwhile we also lose too many consistent edges which results in the bad performance. Especially, when λ equals to 1.0, *edgeNum* drops to less than 10,000 and the accuracy drops to 0.8468.

4.4 Attribute Group Preference Analysis

Through considering user attributes, our model can not only boost sentiment classification accuracy, but also learn which attribute group is more or less likely to like a given movie, which is called attribute group preference analysis. Figure 3 shows the normalized weight of user attribute feature in *Base+UAF*.

Attribute		MH	TT4	SS	FY	FOT	MCDTM	AHON	Average
Gender	female	0.64	0.61	0.55	0.59	0.57	0.61	0.56	0.59
	male	0.06	0.06	0.49	0.13	0.49	0.38	0.42	0.29
Age	1-18	0.62	0.70	0.73	0.65	0.73	0.73	0.49	0.69
	19-30	0.51	0.37	0.53	0.46	0.48	0.48	0.58	0.49
	31-45	0.44	0.34	0.60	0.49	0.61	0.52	0.45	0.49
	45+	0.50	0.16	0.54	0.29	0.38	0.59	0.41	0.41
Location	abroad	0.34	0.44	0.44	0.21	0.54	0.45	0.13	0.36
	first-tier	0.43	0.39	0.53	0.51	0.47	0.48	0.53	0.48
	second-tier	0.45	0.44	0.54	0.50	0.51	0.55	0.48	0.50
	third-tier	0.51	0.45	0.61	0.53	0.58	0.46	0.56	0.53
	fourth-tier	0.59	0.55	0.52	0.52	0.65	0.55	0.51	0.56
Fan	true	0.87	0.72	0.73	0.69	0.71	0.71	0.87	0.76
	false	0.31	0.20	0.37	0.20	0.39	0.44	0.37	0.33

Fig. 3. The normalized weight of user attribute feature in *Base+UAF* (Table 4). Value (from 0.0 to 1.0) shows how possible a user with the attribute feature might like the specific movie. We use different colors to fill in the box. High value with dark color and low value with light color.

For *gender*, we find that the average feature weight of male users is only 0.29 (less than 0.3), which shows male users always give negative comments. The reason is that compared with plain movies, such as love movies, male users may like adventure and excitement ones, while from Table 1 we can find this kind of movies take up a large of proportion in our datasets. Female users may like love movies, therefore the average feature weight of female users is about 0.6, which shows they always write positive comments.

With increasing age (1-18, 19-30, 31-45, 45+), the average feature weight (0.69, 0.49, 0.49, 0.41) decreases, which shows young users often give positive comments and middle-aged ones always write negative comments. We think the reason is that with increasing age, users are increasingly demanding.

For *fan*, we can find users following the main actor or actress of a movie will get high average feature weight about 0.76 and always like the movie, while users being not the fan always give negative comments, which is broadly in line with what we expected.

5 Related Work

Sentiment classification has been studied for years. Lots of researches follow [12] and use machine learning algorithms to build sentiment classifier from reviews with sentiment labels [5,19,6,13,18].

[4,15,3] make use of user interactions (such as following, retweeting etc.) to improve the performance. Their main idea is that sentiments of two messages posted by friends are more likely to be similar than those of two randomly selected messages. Incorporating this information into a graph-based model [4]

or a supervised method [3] gets good results in the task. Other studies [1,14,16] also incorporate the user itself to improve sentiment classification accuracy.

6 Conclusion and Future Work

In this paper, we exploit user attributes to help sentiment classification on social media text. We propose two methods to incorporate user attributes: 1) take them as features; 2) use them to construct user attribute graph and design a graph-based model to handle it. We conduct experiments on seven datasets from Sina Weibo. Experimental results show that incorporating user attributes can significantly boost sentiment classification accuracy.

Since many researchers have proven the effectiveness of user interactions in social media on the sentiment classification task and we have also demonstrated user attributes can be useful for the task. In the future, we would like to investigate how to combine these two kinds of information together.

Acknowledgments

We thank the three anonymous reviewers for their helpful comments and suggestions.

References

1. Al Boni, M., Zhou, K.Q., Wang, H., Gerber, M.S.: Model adaptation for personalized opinion analysis. *Proceedings of Annual Meeting of the Association for Computational Linguistics* pp. 769–774 (2015)
2. Angelova, R., Weikum, G.: Graph-based text classification: learn from your neighbors. In: *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 485 – 492 (2006)
3. Hu, X., Tang, L., Tang, J., Liu, H.: Exploiting social relations for sentiment analysis in microblogging. In: *Proceedings of the international conference on Web Search and Data Mining*. pp. 537–546 (2013)
4. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: *Proceedings of Annual Meeting of the Association for Computational Linguistics*. pp. 151–160 (2011)
5. Li, S., Xia, R., Zong, C., Huang, C.R.: A framework of feature selection methods for text categorization. In: *Proceedings of Annual Meeting of the Association for Computational Linguistics*. pp. 692–700 (2009)
6. Li, S., Xue, Y., Wang, Z., Zhou, G.: Active learning for cross-domain sentiment classification. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. pp. 2127–2133 (2013)
7. Liu, B.: Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1), 1–167 (2012)
8. Liu, K.L., Li, W.J., Guo, M.: Emoticon smoothed language models for twitter sentiment analysis. In: *Proceedings of the international joint conference on Artificial Intelligence*. pp. 1678–1684 (2012)

9. Mohammad, S.M., Kiritchenko, S., Zhu, X.: Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *Proceedings of the International Workshop on Semantic Evaluation Semeval* pp. 321–327 (2013)
10. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Proceedings of Advances in Neural Information Processing Systems 28(3)*, 169–187 (2001)
11. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and trends in information retrieval 2(1-2)*, 1–135 (2008)
12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: *Proceedings of the international conference on Empirical Methods in Natural Language Processing*. pp. 79–86 (2002)
13. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the conference on empirical methods in natural language processing*. p. 1631–1642 (2013)
14. Song, K., Feng, S., Gao, W., Wang, D., Yu, G., Wong, K.F.: Personalized sentiment classification based on latent individuality of microblog users. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. pp. 2277–2283 (2015)
15. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: *Proceedings of the international conference on Knowledge Discovery and Data mining*. pp. 1397–1405. ACM (2011)
16. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. In: *Proceedings of the international conference on Annual Meeting of the Association for Computational Linguistics*. pp. 1014–1023 (July 2015), <http://www.aclweb.org/anthology/P15-1098>
17. Wang, S., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: *Proceedings of the international conference on Annual Meeting of the Association for Computational Linguistics*. pp. 90–94. Association for Computational Linguistics (2012)
18. Xia, R., Zong, C., Hu, X., Cambria, E.: Feature ensemble plus sample selection: domain adaptation for sentiment classification. *Intelligent Systems, IEEE 28(3)*, 10–18 (2013)
19. Xia, R., Zong, C., Li, S.: Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences 181(6)*, 1138–1152 (2011)
20. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: *Proceedings of the international conference on research and development in Information Retrieval*. pp. 42–49. ACM (1999)