TASC:Topic-Adaptive Sentiment Classification on Dynamic Tweets

Shenghua Liu, Xueqi Cheng, Fuxin Li, and Fangtao Li

Abstract—Sentiment classification is a topic-sensitive task, i.e., a classifier trained from one topic will perform worse on another. This is especially a problem for the tweets sentiment analysis. Since the topics in Twitter are very diverse, it is impossible to train a universal classifier for all topics. Moreover, compared to product review, Twitter lacks data labeling and a rating mechanism to acquire sentiment labels. The extremely sparse text of tweets also brings down the performance of a sentiment classifier. In this paper, we propose a semi-supervised topic-adaptive sentiment classification (TASC) model, which starts with a classifier built on common features and mixed labeled data from various topics. It minimizes the hinge loss to adapt to unlabeled data and features including topic-related sentiment words, authors' sentiments and sentiment connections derived from "@" mentions of tweets, named as topic-adaptive features. Text and non-text features are extracted and naturally split into two views for co-training. The TASC learning algorithm updates topic-adaptive features based on the collaborative selection of unlabeled data, which in turn helps to select more reliable tweets to boost the performance. We also design the adapting model along a timeline (TASC-t) for dynamic tweets. An experiment on 6 topics from published tweet corpuses demonstrates that TASC outperforms other well-known supervised and ensemble classifiers. It also beats those semi-supervised learning methods without feature adaption. Meanwhile, TASC-t can also achieve impressive accuracy and F-score. Finally, with timeline visualization of "river" graph, people can intuitively grasp the ups and downs of sentiments' evolvement, and the intensity by color gradation.

Index Terms—Sentiment classification, social media, topic-adaptive, cross-domain, multiclass SVM, adaptive feature

1 Introduction

The booming Microblog service, Twitter, attracts more people to post their feelings and opinions on various topics. The posting of sentiment contents can not only give an emotional snapshot of the online world but also have potential commercial [1], [2], financial [3] and sociological values [4]. However, facing the massive sentiment tweets, it is hard for people to get overall impression without automatic sentiment classification and analysis. Therefore, there are emerging many sentiment classification works showing interests in tweets [5], [6], [7].

Topics discussed in Twitter are more diverse and unpredictable. Sentiment classifiers always dedicate themselves to a specific domain or topic named in the paper. Namely, a classifier trained on sentiment data from one topic often performs poorly on test data from another [8]. One of the main reasons is that words and even language constructs used for expressing sentiments can be quite different on different topics. Taking a comment "read the book" as an example, it could be positive in a book review while negative in a movie review. In social media, a Twitter user may have different opinions on different topics [9]. Thus, topic adaptation is needed for sentiment classification of tweets on emerging and unpredictable topics. Previous works [10], [11], [12] explicitly borrowed a bridge to connect a topic-dependent feature to a known or common feature. Such bridges are built between product reviews by assuming that the parallel sentiment words exist for each pair of topics, such as books, DVDs, electronics and kitchen appliances. However, it is not necessarily applicable to topics in Twitter, especially the unpredictable ones. It is worth mentioning that detecting and tracking topics from tweets is another research topic. Ad-hoc Microblog search in Text REtrieval Conference (TREC) 2011 [13] and 2012 [14] is hopefully a choice for people to query tweets on emerging topics, and sentiment classification can be conducted afterwards.

Unlike product reviews that are usually companied with a scoring mechanism quantifying review sentiments as class labels, there lack labeled data and rating mechanism to generate them in Twitter service. Go et al. [15] used emoticons as noisy labels for Twitter sentiment classification. But the neutral class could not be labeled in this way, and unexpected noise may be introduced only relying on emoticons to label sentiment classes. Semi-supervised approaches [16], [17], [18], [19] have been used for sentiment classification with a small amount of labeled data for other medium than Microblog. They did not take the special nature of tweets, such as emoticons, users, and networks, to select unlabeled data for training. Tan et al. [9] explored the sentiment correlations affected by users who connect with each other, i.e., social network and @-network formed by users referring to each other with "@" mentions in tweets. However, the work focused on user-level sentiment analysis, while we analyze at the tweet-level.

Semi-supervised support vector machines [20], [21], a.k.a. S3VM, is one of the more promising candidates to

- *S. Liu, X. Cheng, and F.X. Li are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China 100190. E-mail: liu.shengh@gmail.com, cxq@ict.ac.cn, lifuxin@software.ict.ac.cn.*
- *F.T. Li is with Google Inc., Mountain View, CA 94043. E-mail: fangtao06@gmail.com.*

Manuscript received 25 June 2014; revised 20 Nov. 2014; accepted 28 Nov. 2014. Date of publication 17 Dec. 2014; date of current version 27 Apr. 2015. Recommended for acceptance by P.G. Ipeirotis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2382600

utilize unlabeled data combining with a small amount of labeled ones, since SVM minimizes the structural risk. And collaborative training (co-training) framework [22] is an alternative wrapper, and achieves a good performance, which is often used in the scenarios whose features are easily split into different views. Nigam and Ghani [23] demonstrated that algorithms that manufacture a feature split outperform algorithms not using a split, meanwhile algorithms leveraging a natural independent split of features perform better. Since of extremely sparse text, it needs to extract more features other than sentiment words for tweets, and use the independent split of features for co-training.

Therefore our work focuses on cross-domain sentiment analysis on tweets, and we propose a semi-supervised topic-adaptive sentiment classification model (TASC). It transfers an initial common sentiment classifier to a specific one on an emerging topic. TASC has three key components.

1) The semi-supervised multiclass SVM model is formalized. Given a small amount of mixed labeled data from topics, it selects unlabeled tweets in the target topic, and minimizes the structural risk of labeled and selected data to adapt the sentiment classifier to the unlabeled data in a transductive manner.
2) We set feature vector in the model into two parts: fixed common feature values and topic-adaptive feature variables. Topic-adaptive words as adaptive features are expanded and their values are updated in semi-supervised iterations to help transfer sentiment classifier. Seeing consistencies of users' sentiment and correlations between a tweet and the ones posted by its @-users, we add user-level and @-network-based features as another adaptive features to help sentiment classifier find more reliable and unlabeled tweets for adapting.
3) To tackle the content sparsity of tweets, more features are extracted, and split into two views: text and non-text features. Since the independent nature of two feature sets, co-training framework is employed to collaboratively transfer the sentiment classifier.

An iterative algorithm is proposed for solving TASC model by alternating among three steps: optimization, adapting to unlabeled data, and adapting to features. The algorithm iteratively minimizing the margins of two independent objectives separately on text and non-text features to learn coefficient matrices. With agreement, adapting to unlabeled data step selects unlabeled tweets from a given topic according to the confidence evaluated using the current estimation of coefficient matrices. And then topic-adaptive features including sentiment words are updated to help the first two steps find more reliable tweets to boost the final performance of sentiment classification on the target topic. Besides, since of the dynamics of tweets, TASC is extended to adapt along a timeline, named TASC-t. Compared with the well-known supervised and ensemble sentiment classifiers, namely, DT (Decision Tree), multiclass SVM and RF (Random Forest); and semi-supervised approaches, namely, multiclass S3VM and co-training multiclass SVM, TASC achieves improvements in the mean accuracies of five repeated runs on six topics from the public tweet corpuses. TASC-t also reports impressive accuracies and F-scores, even better than the mean values by TASC on some topics. To better evaluate our algorithm, we test it with some different ratios for randomly sampling training data, and controls parameter settings.

Furthermore, sentiment visualization plays an important role in helping people understand sentiment classification results. Most sentiment classification works were ended with a passively produced charts or tables, showing the final portions of each classes, i.e., ratios of positive, neutral and negative [24]. However, with poorly designed sentiment visualization, it prevents people from grasping the insights, without reading the large classified list of unstructured tweets. Thus, with the classification results on "President Debate", we use a "river" graph to visualize sentiment intensity by color gradation, and ups and downs of sentiment polarities by the river flow along a timeline. With the graph, people can clearly recognize the trend of the topic as well as the sentiment peak and bottom.

The rest of paper is organized as follows. Section 2 investigates the previous related works. We then analyze the data in hand, and give the observations and motivations in Section 3. The features, TASC model and learning algorithm are described in Section 4. Experiment results are reported and timeline visualization is demonstrated for the results in Section 5. Section 6 finally concludes the whole paper.

2 Related Work

Cross-domain sentiment classification is challenging and many works proposed their solutions. Blitzer et al. [10] proposed an approach called structural correspondence learning (SCL) for domain adaptation. It employed the pivot features as the bridge to help cross-domain classification. Pan et al. [12] proposed a spectral feature alignment (SFA) algorithm to bridge the gap between the domains with domain independent words. Li et al. [11] proposed the cross-domain sentiment lexicon extraction for sentiment classification. He et al. [25] and Gao and Li [26] employed probabilistic topic model to bridge each pair of domains in a semantic level. All the above studies focus on the review data set. However, Twitter data is much different from the product reviews. Twitter contains diverse topics from different domains, which are unpredictable and each topic need sufficient labeled data and features to train a topic-specific classifier. Besides the above document-level sentiment classification works, aspect-based sentiment analysis detects topic spans, and associations of topic aspects, sentiment words, and opinion holders in a document or even sentence [27], [28], [29], [30]. And SUIT model [31] considered the topic aspects and opinion holders for cross-domain sentiment classification via supervised learning. We also notice that [32] conducted the experiments on cross-media sentiment analysis with news, blogs and Twitter data sets. They also found Twitter data set was very different from other resources. Our work proposes a topic-adaptive sentiment classification model for tweet document-level analysis, transferring a common classifier to a specific one on an emerging topic.

Microblogs as a social media have attracted many studies on sentiment analysis [7], [24], [33], [34]. Go et al. [15] introduced a distant supervised learning approach

TABLE 1
Corpus Statistics

Topics	Positive	Neutral	Negative	Total
Apple	191	581	377	1,149
Google	218	604	61	883
Microsoft	93	671	138	902
Twitter	68	647	78	793
Taco Bell	902	2,099	596	3,597
President Debate	1,465	1,019	729	3,213

for automatically classifying the sentiment of tweets using emoticons as noisy labels for training data. Tumasjan et al. [4] showed that Twitter can be considered as a valid indicator of political opinion. Kouloumpis et al. [35] leveraged the existing hashtags in tweets to build training data and demonstrated that part-of-speech features might not be useful for sentiment analysis of tweets. Mehta et al. [36] used the Twitter data as a corpus for sentiment analysis and tracking the influence of a particular brand activity on the social network. We focus on sentiment classification problem for tweets.

There lack sentiment labels in tweets for supervised learning of a sentiment classifier. Turney [37] presented an effective unsupervised learning algorithm, called semantic orientation, for classifying reviews. A web-kernel based measurement was proposed as PMI-IR to measure the weight of a sentiment word, which is independent to the corpus collection in hand. But such a measurement is suitable for common sentiment words which always have a stable weight and fixed sentiment polarity, i.e., negative or positive in contexts of diverse topics. Besides, the text content of tweets is too sparse to extract plenty of salient sentiment words. Except text feature of sentiment classification, there have been attracted many studies which considered other features to improve the classification result. Wang et al. [38] used the graph of hashtag co-occurrence for sentiment classification on hash-tag-level. Wang et al. [39] developed a semi-supervised factor graph model which incorporates both the single image features and the image correlations to better predict the emotional impact. In social networks, Jia et al. [40] showed that users generated the images which embedded users' emotional impact and influenced the emotional impact of each other. Zhang et al. [41] predicted users' emotions in a social network, based on a dynamic continuous factor graph model which modeled the users' historic emotion logs and their social network. Tan et al. [9] showed that social relationships could be used to improve user-level sentiment analysis. Their approach was based on that the users who connected with each other might be more likely to hold similar opinions. However, our work focuses on the tweet-level sentiment classification across diverse topics, considering rich non-text features, such as user's sentiment, and network of "@" mentions.

Visualizing themes and dynamics of complex data [42], [43], [44] even in the area of text mining [45] has been studied. However, there are only a few studies dealt with the sentiment visualization. As far as we know, Wu et al. [46] proposed the opinion triangle and ring to visualize the hotel reviews on different polarities, and the periodic pattern is not applicable to visualize the sentiment evolvement of event series. Alper et al. [47] gave an intuitive sentiment visual with OpinionBlocks on different aspects of a product, which was an analysis of overall reviews. Hao et al. [47] used pixel cell-based sentiment calendars and high density geo maps for visualization. Das et al. [48] presented simple directed paths to show the temporal relations between sentiment events. Nevertheless, those visualizations cannot show the dynamics and trend of sentiment along a timeline of a topic. In our sentiment visualization, we use a "river" graph to intuitively show the sentiment classification results and its dynamic process on a tweet topic.

3 Data Observations and Motivations

To get a convincing observations, we collect the publicly available tweets with sentiment labels on diverse topics, consisting of three corpuses. One is Sanders-twitter sentiment corpus consisting of $5{,}513$ manually labeled tweets. These tweets are divided into four different topics (Apple, Google, Microsoft, and Twitter). After removing the non-English and spam tweets, we have $3{,}727$ tweets left. Another one is the $9{,}413$ tweets mentioning "Taco Bell" during January $24\text{-}31, 2011$. And the last one is the first 2008 Presidential debate corpus [49] with sentiment judgments on $3{,}238$ tweets by AMT (Amazon Mechanical Turk). The detailed information of the data is shown in Table 1.

With necessary pre-processes, we use the Stanford POS (part-of-speech) tagger to tag the tweets on each topic, and select the frequent adjectives, verbs, nouns and adverbs as candidates of sentiment words. Table 2 lists a part of representative sentiment words on each topic in an alphabet order, extracted from the corpuses. It is seen that tweets on various topics may use quite different words for expressing sentiments, although there are some common ones, which agrees with [12]. To make matters worse, the same word in one topic may mean positive but in another topic may mean negative, e.g., the sentiment word "unpredictable" could be positive on the topic of a thriller but negative when describing the brake system of the Toyota. Thus such differences in

TABLE 2
Sentiment Words Extracted from Tweets on Various Topics

Topics	sentiment words
Apple	amazing, better, buy, dear, design, genius, great, hate, issues, support
Google	available, amazing, cool, improvements, infinite, more, new, really, sharing, unveil
Microsoft	available, celebrity, coming, deal, free, great, learning, love, review, using
Twitter	addicted, cool, follow, free, funny, lol, movement, occupy, tired, trending
Taco Bell	alternatives, crisis, defend, fights, issue, lol, marketing, officially, promoting, surprised
President Debate	agree, better, democracies, difference, experience, independent, lost, opposing, vote, winner

TABLE 3
Sentiment Statistics of Users' Tweets

	total users	users (≥2 tweets)	average var
Taco Bell	3,446	106	0.1008
President Debate	1,204	520	0.4168

sentiment lexicons stop a sentiment classifier from directly adapting to different topics. And considering the differences of those sentiment words in turn helps classifiers adapt to a specific topic. We name them as topic-adaptive sentiment words in our paper.

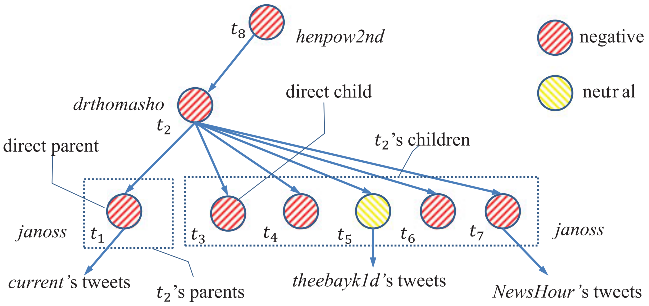
Besides, a user posting tweets on a topic aims to reflect his opinion. Intuitively the sentiment expressed by a user should be consistent in a context. Namely, the tweets with close posting time by a user probably belong to the same sentiment class. Since the corpuses of "Taco Bell" and "President Debate" contain authors' names, the statistics of users (posting no less than two tweets) in Table 3 shows that the average variances of sentiment labels (-1 for negative, 0 for neutral, and 1 for positive) are 0.10 and 0.42 separately. And if a user's sentiment is evenly distributed, that is, the chance of posting a negative, neutral or positive tweet is equal, i.e., $1/3$, the variance of his sentiment is 0.67, calculated as follows:

$$\begin{aligned}\mathbf{Var}(X) &= \mathbf{E}(X^2) - (\mathbf{E}(X))^2\\ &= \frac{2}{3} - 0 = 0.67,\end{aligned}$$

where X is a random variable to represent sentiment of a user's tweet, $\mathbf{E}(\cdot)$ is the mean and $\mathbf{Var}(\cdot)$ is the variance of sentiments of all the user's tweets. Thus the statistics on the two topics shows the consistency of users' sentiment based on the significantly smaller variances than 0.67. And the overall sentiment of a user's tweets can reflect his sentiment.

At last, "@" is a commonly used convention in a tweet, e.g. user a_i mentioned a_j via a tweet t containing "@a_j". Such a @-relation reflects the dependencies between the sentiments of tweet t and @-user a_j's tweets in the same context. Thus we build a graph to model @-network, with tweets as vertices and @-relations between tweets as directed edges. There are directed edges from tweet t to all a_j's tweets in the example. Those a_j's tweets posted earlier than tweet t are called t's parents, otherwise those tweets posted later are its children. The parent with the nearest posting time to tweet t is its direct parent, and similarly the nearest child is its direct child. Fig. 1 illustrates a small part of the network on a topic. The detailed information of tweets are listed bellow. Tweet t_2 posted by user *drthomasho* mentioned user *janoss*. User *janoss* successively posted tweets t_1, t_3, t_4, t_5, t_6 and t_7, among which t_1 was earlier than t_2, and the rest were later than t_2. So there are directed edges pointing from t_2 to them, with t_1 as direct parent, the rest five tweets as children, and t_3 as direct child. Besides, user *drthomasho* was mentioned by tweet t_8, so the example has a directed edge from t_8 to t_2.

With statistics of sentiment correlations in @-network, Fig. 2 shows the ratios that a tweet has the same sentiment to its parents, children, direct parent, and direct children. And it also calculates the probability in the case that sentiment labels are randomly assigned to the tweets in the network, keeping the total number of each class labels unchanged. The ratios show that a tweet gets a higher probability to have the same sentiment as its adjacent tweets than random assignment on "Taco Bell". On "President Debate", the results of parents and direct parent don't have significant differences comparing to that of random assignment, while the results of children and direct child show higher probability. It shows that users might prefer to post objective or neutral tweets at the beginning, and express the positive or negative opinions later. Or they might mention others for debates. Fig. 1 also illustrates an observed example agreeing to the statistics that adjacent tweets tend to have similar sentiment, where red (slash shaded) represents negative, and yellow (back slash shaded) represents neutral.



	post time	author	@ whom	sentiment
t_1	2:01	*janoss*	@current	negative
t_2	2:13	*drthomasho*	@janoss	negative
t_3	2:19	*janoss*	--	negative
t_4	2:29	*janoss*	--	negative
t_5	2:37	*janoss*	@theebayk1d	neutral
t_6	2:38	*janoss*	--	negative
t_7	2:55	*janoss*	@NewsHour	negative
t_8	2:57	*henpow2nd*	@drthomasho	negative

Fig. 1. Example of @-network.

Therefore, in a semi-supervised setting, the consistency of user's tweets and correlation of adjacent tweets in @-network are useful for finding more reliable unlabeled tweets, insuring the training adaptive to a target topic.

4 Topic Adaptive Sentiment Classification

Sentiment classification in the work is treated as a multiclass classification problem, with positive, neutral, and negative expressions. The training data is given in (x_i, y_i), $y_i \in \{1, \ldots, K\}$, where x_i is the feature vector with each

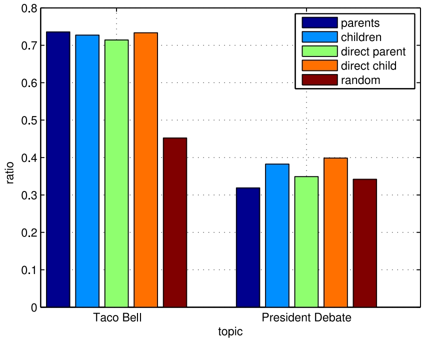


Fig. 2. The ratios that adjacent tweets in @-network have the same sentiment.

element as the value of the corresponding feature, y_i is the class that the data belongs to, and k is the number of classes. As we analyze sentiment at tweet document-level, it assumes that a tweet t_i belongs to one and only one overall sentiment class, expressing on a single topic from a single opinion holder [8]. Sometimes opinions are expressed on more than one topics in a post by different holders, so we can divide it into different posts by detecting topic spans and opinion holders [27], [28].

In this section, multiclass SVM model is first introduced as preliminaries. And then we discuss the features used for classification model, and some of feature values are not finalized which are topic-adaptive. Three key components for TASC model, namely, adapting to unlabeled data, adapting to features, and collaboratively training are described. An iterative procedure is proposed to solve TASC model afterwards. At last, the scheme adapting along a timeline is designed for TASC model, since of the dynamics of tweets on a topic.

4.1 Multiclass SVM

SVMs model is originally built for binary classification. And there are intuitive ways to solve multiclass with SVMs. The most common technique in practice has been to build K "one-versus-rest" classifiers, and to choose the class which classifies the test data with greatest margin. Or we build $K(K-1)/2$ "one-versus-one" classifiers, and choose the class that is selected by the most classifiers. In our work, we choose the "one-versus-rest" strategy, and the multiclass SVMs model is as follows:

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{K} w_i^T w_i + \frac{C}{n}\sum_{i=1}^{n} \max_{y \neq y_i} \{0, 1 - w_{y_i}^T x_i + w_y^T x_i\}, \quad (1)$$

where the bold symbol of $\mathbf{w}$ in (1) is a matrix with column w_i as the coefficient vector corresponding to the features for class $i \in 1, \ldots, K$. As $w_y^T x_i$ is the confidence score of tweet t_i belonging to class y, the second summation item is the loss of every tweet t_i belonging to class y other than y_i. And C is a constant coefficient. To better capture how C scales with the training set, C is divided by the total number of labeled tweets. The model (1) shows structural risk is considered to optimize, and the result of the model is single support vector machine, instead of multiple SVMs for binary classification.

Then the class label of tweet t_i is predicted by formula (2),

$$y_i' = \arg\max_{y} \{w_y^T x_i\}. \quad (2)$$

4.2 Features

Features are extracted and split into two views, i.e., text feature set χ_1 of sentiment words, and non-text feature set χ_2, including emoticons, temporal features, punctuation, user-level features and @-network-based features.

4.2.1 Text Features

Common and topic-adaptive sentiment words form text features. We collect common sentiment words from WordNet Affect [50] and public sentiment lexicon [51], denoted as set P. To evaluate the feature values, we use the Pointwise Mutual Information and Information Retrieval (PMI-IR) [37]. With Google search engine as the kernel, we get the number of querying hits, and then calculate two PMI values of a common sentiment word separately with two sentiment orientation words, such as "excellent" and "poor", showing the strength of semantic associations. Afterwards, the feature value of the common sentiment word is calculated as the difference between the PMI values.

With POS tagging for tweets on a topic and removing the common sentiment words, we select the frequent adjectives, verbs, nouns and adverbs as candidates of topic-adaptive sentiment word set Π. The initial feature values $x_\pi = 0, \pi \in \Pi$. And the dimensions of text feature set χ_1 is $v + u$, where $v = |P|$ and $u = |\Pi|$.

4.2.2 Non-Text Features

Temporal features. Tweets are real-time, which are different from traditional web documents, and people express their sentiment dynamically over time. Onur et al. [52] shows that users' opinions are correlated well with their biological clock. Thus we extracted the post time from tweets as temporal features. As for tweets posted in the different time zones, we map the post time into their local periods.

Emoticon features. A set of emoticons from Wikipedia are collected as a dictionary,[1] such as :-), :), (∼_∼;),(> _ <), >: [, :-(, :(, etc. They are labeled with positive $(+1)$, neutral (0), or negative (-1) emoticons. The corresponding values of emoticons in a tweet are summed up as its emoticon feature value.

Punctuation features. As users often expressed their emotion with punctuation marks, such as exclamation mark (!), question mark $(?)$, and their combinations and repeats, etc, we counted them for features.

User-level features. Based on the previous observations, a user's sentiment on a topic is consistent, which could be determined by his tweets. It in turn means the sentiment of tweets posted by a user is correlated to his opinion. As the sentiment of a user can belong to any one of K classes, we define user-level features of a tweet to be $\nu_k \in \chi_2, k = 1, \ldots, K$, simply denoted as $\nu_{1\sim K}$. The corresponding non-negative values $x_{\nu,1\sim K}$ indicate the weights that its author belongs to the corresponding classes, and at most one of K feature values is non-zero due to the exclusiveness of sentiment classes. Since the user-level features are related to topics, we just initialize their values as zeros before a topic is given.

@-network-based features. Recall that we build a directed graph on tweets to model @-network. We have two alternative @-network-based features for a tweet. One is direct-parent and -child features, assigning sentiments of the tweet's direct parent and child as feature values. The other is all-parents and -children features, assigning total sentiments of all the parents and children of the tweet separately as feature values. We denote parent and child features be $\rho_{1\sim K} \in \chi_2$ and $\sigma_{1\sim K} \in \chi_2$ separately. The feature values are $x_{\rho,1\sim K}$ and $x_{\sigma,1\sim K}$, initialized as zeros before specifying a topic, similar to the user-level features.

[1] http://en.wikipedia.org/wiki/List_of_emoticons

4.3 Adapting to Unlabeled Data

The sentiment classifiers trained using labeled tweets from one topic are not usually adaptive to another one. Thus we choose labeled data set L from an even mixture of various topics. The initial sentiment classifier trained with set L can be viewed as a common and weak classifier, since of the small amount of labeled data on mixed topics and common sentiment features. To adapt to sentiment classification on topic e, the unlabeled tweet set U on the topic can be used for transductive learning without any more costs of manually labeling.

With adding those unlabeled tweets as an optimization term in the optimization model (1), the semi-supervised multiclass SVM model adapting to unlabeled tweets on topic e is formed as model (3),

$$\begin{aligned}\min_{\mathbf{w},\xi} \frac{1}{2}\sum_{i=1}^{K} w_i^T w_i + \frac{C}{|L|}\sum_{t_i\in L}\max_{y\neq y_i}\left\{0, 1 - w_{y_i}^T x_i + w_y^T x_i\right\} \\ + \frac{C'}{|U|}\sum_{t_j\in U}\max_{y\neq y_j}\left\{0, 1 - w_{y'_j}^T x_j + w_y^T x_j\right\},\end{aligned} \tag{3}$$

where $|L|$ and $|U|$ indicate the number of elements in sets L and U separately. Furthermore, the selected unlabeled tweet t_j is predicted to be class y'_j as formula (2). Thus the following equation satisfied:

$$w_{y'_j}^T x_j = \max_{:y}\left\{w_y^T x_j\right\}.$$

We define the function submax$\{\cdot\}$ as the second largest value in set $\{\cdot\}$. So the slack for unlabeled data in the third minimization term of model (3) is as follows:

$$\begin{aligned}&\max_{y\neq y'_j}\left\{0, 1 - \left(w_{y'_j}^T x_j - w_y^T x_j\right)\right\} \\ &= \max\left\{0, 1 - \max_y\left\{w_y^T x_j\right\} + sub\max_y\left\{x_y^T x_j\right\}\right\} \\ &= h_l\left(\max_y\left\{w_y^T x_j\right\} - sub\max_y\left\{w_y^T x_j\right\}\right).\end{aligned}$$

It is the hinge loss $h_l(\cdot) = \max\{0, 1 - \cdot\}$ of the difference between the largest and the second largest confidence score of tweet t_i among all the sentiment classes. In practice, not all the unlabeled tweets are picked for the adaptive training. Only the most confident classifying results are preferred to be added to avoid bringing much noise. We define the normalized confidence score S_j of tweet t_j with predicted class y'_j as

$$S_j = \frac{w_{y'_j}^T}{\sum_y w_y^T x_j} = \frac{\max_y\left\{w_y^T x_j\right\}}{\sum_y w_y^T x_j}.$$

Therefore given a confidence threshold τ, the unlabeled tweets t_j satisfying $S_j \geq \tau$ are selected. Thus the optimization model with confidence threshold is as follows:

$$\begin{aligned}\min_{\mathbf{w},\xi} \ \frac{1}{2}\sum_{i=1}^{K} w_i^T w_i + \frac{C}{|L|}\sum_{t_i\in L}\max_{y\neq y_i}\left\{0, 1 - w_{y_i}^T x_i + w_y^T x_i\right\} \\ + \frac{C'}{|\alpha|}\sum_{t_j\in U}\alpha_j h_l\left(\max_y\left\{w_y^T x_j\right\} - sub\max_y\left\{w_y^T x_j\right\}\right)\end{aligned} \tag{4a}$$

$$s.t. \quad \alpha_j = \mathbf{I}(S_j \geq \tau) \ \forall t_j \in U, \tag{4b}$$

where C' is a constant coefficient for unlabeled optimization term, α is binary vector, and each value indicates whether the unlabeled tweet is selected for optimization. $\mathbf{I}(\cdot)$ is indicator function, outputting 1 if the argument is true, and 0 otherwise. $|\alpha|$ is the number of ones in binary vector α.

4.4 Adapting to Features

Topic-related features may have different values on topics. We subsequently treat feature values x_π of topic-adaptive sentiment words, user-level feature value $x_{\nu,1\sim K}$, and @-network-based feature values $x_{\rho,1\sim K}$ and $x_{\sigma,1\sim K}$ as variables, initialized as zeros. The rest feature values in vector x keep constant for a tweet, which are shared among various topics. Thus the feature value vector x can be written as

$$\begin{aligned}x = \overbrace{x_1, \ldots, x_v, \overbrace{x_{v+1}, \ldots, x_{v+u}}^{x_\pi, \pi\in\Pi}}^{\text{text feature set } \chi_1}, \\ \underbrace{x_{v+u+1}, \ldots, x_{\nu,1\sim K}, x_{\rho,1\sim K}, x_{\sigma,1\sim K}}_{\text{non-text feature set } \chi_2}.\end{aligned} \tag{5}$$

In the following, we introduce the way our model to adapt to those features on topic e.

4.4.1 Topic-Adaptive Sentiment Words

In the adaptive training, the feature values of topic-adaptive sentiment words are evaluated in the selected unlabeled tweets on topic e. The weight of a topic-adaptive sentiment word π belonging to a class y is defined as equation (6),

$$\varphi_y(\pi) = \sum_{j, y'_j = y}\alpha_j f_j(\pi)\cdot w_{y'_j}^T x_j, \tag{6}$$

where $f_j(\pi)$ is the term frequency of word π in tweet t_j. The equation shows that $\varphi_y(\pi)$ is the weighted summation of the term frequency of word π in the tweets t_j with predicted class y'_j being y. The sentiment class that word π belongs to is

$$\arg\max_y\{\varphi_y(\pi)\}.$$

So the feature value

$$x_\pi = \max_y\{\varphi_y(\pi)\}.$$

Similar to add unlabeled data, we do not hope all the topic-adaptive sentiment words are added. Thus we define the significance ϖ_π of sentiment word π belonging to a predicted class as follows:

$$\varpi_\pi = \frac{\max_y\{\varphi_y(\pi)\}}{\sum_y \varphi_y(\pi)}.$$

The normalized ϖ_π indicates the significance of sentiment word π in the class $\arg\max_y\{\varphi_y(\pi)\}$ than that in other classes. Finally given a significance threshold θ, the selection vector β is defined as equation (7),

$$\beta_\pi = \mathbf{I}(\varpi_\pi \geq \theta). \tag{7}$$

Thus the feature values of topic-adaptive sentiment words in objective function (4a) are calculated as follows:

$$x_\pi = \beta_\pi \cdot \max_y \{\varphi_y(\pi)\}, \ \pi \in \Pi. \tag{8}$$

4.4.2 User-Level Features

As for user-level features $\nu_{1\sim K}$ of a tweet, let tweet set N_ν be the selected unlabeled tweets from the tweet's author ν in the step of adapting to unlabeled data. The weight of the author's overall sentiment belonging to a class y is defined as follows:

$$\psi_y(\nu) = \sum_{t_j \in N_\nu, y'_j = y} \alpha_j w_{y'_j}^T x_j.$$

Thus we calculate the feature values as formula (9),

$$\begin{aligned} x_{\nu,k} = \psi_k(\nu)\mathbf{I}(\mathbf{k} = \arg\max_{\mathbf{y}}\{\psi_{\mathbf{y}}(\nu)\}) \\ k = 1, \ldots, K. \end{aligned} \tag{9}$$

4.4.3 @-Network-Based Features

Similarly, @-network-based features are evaluated in the selected unlabeled tweets on topic e as well. Let N_ρ and N_σ separately be the tweet's parents and children that are selected from unlabeled data. We define the weights of overall sentiments of the parents and children belonging to a class y as follows:

$$\begin{aligned} \varsigma_y(\rho) &= \sum_{t_j \in N_\rho, y'_j = y} \alpha_j w_{y'_j}^T x_j \\ \varsigma_y(\sigma) &= \sum_{t_j \in N_\sigma, y'_j = y} \alpha_j w_{y'_j}^T x_j. \end{aligned}$$

And the features values are

$$x_{\rho,k} = \varsigma_k(\rho)\mathbf{I}(k = \arg\max_y\{\varsigma_y(\rho)\}) \tag{10}$$

$$\begin{aligned} x_{\sigma,k} = \varsigma_k(\sigma)\mathbf{I}(k = \arg\max_y\{\varsigma_y(\sigma)\}) \\ k = 1, \ldots, K. \end{aligned} \tag{11}$$

4.5 Adaptive Co-Training Algorithm

We naturally adopt the text and non-text features, χ_1 and χ_2 as independent views for co-training. In co-training scheme, two classifiers C_1 and C_2 are trained based on χ_1 and χ_2 separately using initial labeled data L. The corresponding feature values are denoted as x and x' respectively for text and non-text feature values, instead of x in equation (5). The unlabeled data are selected collaboratively to augment labeled data set L, which is used for the next iteration. And the final sentiment classification result is obtained by the classifier trained on the combining features $\{\chi_1, \chi_2\}$ using augmented labeled data set L.

In order to make adaptive multiclass S3VMs model (4) in a co-training scheme, we define another selection vector α' of unlabeled data for another multiclass S3VMs,

$$\alpha'_j = \mathbf{I}(S'_j \geq \tau). \tag{12}$$

And classification confidence S'_j is calculated with non-text feature values x'_j as follows:

$$S'_j = \frac{\max_y\{(w'_y)^T x'_j\}}{\sum_y (w'_y)^T x'_j}.$$

Algorithm 1. Algorithm of learning TASC on topic e

```
Given:
    Text feature set $\chi_1$ and feature values $x$, initializing $x_\pi = 0, \pi \in \Pi$;
    Non-text feature set $\chi_2$ and feature values $x'$, initializing $x_{\nu,1\sim K} = 0, x_{\rho,1\sim K} = 0, x_{\sigma,1\sim K} = 0$;
    $L$: labeled tweets containing $K$ sentiment classes on mixing topics;
    $U$: unlabeled data on topic $e$;
    $\tau$: classification confidence threshold;
    $\theta$: threshold for topic-adaptive words expansion;
    $l$: the maximum number of unlabeled tweets selected in one iteration;
    $c$: the maximum number of topic-adaptive sentiment words selected in one iteration for each class;
    $M$: the maximum number of co-training iterations.

loop
    repeat
        [Optimization]
        Minimize objective (13) for $\mathbf{w}$ on feature set $\chi_1$;
        Minimize the objective for $\mathbf{w}'$ on feature set $\chi_2$;
        [Adapting to unlabeled data]
        Calculate confidence scores $S_j$ and $S'_j$ by equations (4b) and (12) with $\mathbf{w}$ and $\mathbf{w}'$ separately;
        Select the $l$ most confident and unlabeled tweets $t_j$ in each sentiment class, such that $\alpha_j \cdot \alpha'_j = 1$;
        Move them with predicted class labels from $U$ into $L$;
    until $\forall t_j \in U$ such that $\alpha_j \cdot \alpha'_j = 0$ or number of iterations $> M$.
    [Adapting to features]
    Calculate the significance $\varpi$ with $\mathbf{w}, \alpha, \alpha'$ and the current feature vector $x$;
    Select the $c$ most significant and topic-adaptive sentiment words $\pi$ for each class, such that $\beta_\pi = 1$;
    Update feature values $\{x_\pi | \pi \in \Pi\} \subset x$ by equation (8);
    Update the user-level feature values $x_{\nu,1\sim K} \subset x'$ by equation (9);
    Update the @-network-based feature values $x_{\rho,1\sim K}$, $x_{\sigma,1\sim K} \subset x'$ by equations (10) and (11);
end loop
Train multiclass SVM $C^*$ on the features consist of $x$ and $x'$ using augmented $L$.
return $L, x, x'$ and $C^*$.
```

To avoid noise, we follow "agreement" strategy [53], and only select the confident and unlabeled data that both classifiers agree most. Thus we replace α_j with $\alpha_j \cdot \alpha'_j$ as the selection coefficient in the third optimization term of objective (4a) resulting in the objective of text view χ_1 as formula (13). Similarly, in the equations for adaptive features, α should be substituted by $\alpha_j \cdot \alpha'_j$ as well. And the objective of non-text view χ_2 has the same form as formula (13) by replacing coefficient matrix $\mathbf{w}$ and feature value vector x with $\mathbf{w}'$ and x',

$$\min_{\mathbf{w},\xi} \frac{1}{2}\sum_{i=1}^{K} w_i^T w_i + \frac{C}{|L|}\sum_{t_i \in L} \max_{y \neq y_i} \{0, 1 - w_{y_i}^T x_i + w_y^T x_i\} + \frac{C'}{\boldsymbol{\alpha}^T \boldsymbol{\alpha}'} \sum_{t_j \in U} \boldsymbol{\alpha}_j \boldsymbol{\alpha}'_j \cdot h_l\big(\max_y \{w_y^T x_j\} - sub\max_y \{x_y^T x_j\}\big). \tag{13}$$

The joint objectives of text and non-text views compose our TASC model. Coefficient matrices **w** and **w′** separately are used for linear combinations of text and non-text feature value vectors x and x'. The adaptive feature variables $\{x_\pi | \pi \in \Pi\} \subset x, \{x_{\nu,1\sim K}, x_{\rho,1\sim K}, x_{\sigma,1\sim K}\} \subset x'$, and selection vectors α and α' for unlabeled data to be estimated. Thus we can solve this model using an iterative procedure that alternates among the following three steps.

Optimization step for solving coefficient matrices **w** and **w′**: by fixing selection vectors α and α', and feature values x and x', we solve two minimization of multiclass SVM separately, since the feature sets χ_1 and χ_2 are designed conditionally independent.

Adapting to unlabeled data step for solving selection vectors α and α': by fixing matrices **w, w′** and feature values x, x', the unlabeled data are selected by comparing the confidence scores S_j and S'_j with threshold τ according to constraint (4b) and formula (12) separately in a transductive way.

Adapting to features step for solving adaptive feature variables $x_\pi, x_{\nu,k}, x_{\rho,k}$ and $x_{\sigma,k}$: by fixing **w, w′**, and selection vectors α and α', topic-adaptive sentiment words are expanded according to equation (7), and the corresponding feature values are evaluated as equation (8). User-level and @-network-based feature values are calculated as equations (9) and ((10), (11)) respectively.

In the details of the algorithm, unlabeled tweets and topic-adaptive sentiment words are added gradually to avoid misleading of the overwhelming unlabeled data and adaptive features at early iterations. As shown in Algorithm 1, it is given a common feature set split into independent views: text view χ_1 and non-text view χ_2, an even mixture of labeled tweets from various topics, and a specific topic e with its unlabeled tweet set U. The candidates of topic-adaptive sentiment words are extracted from unlabeled set U, which is described in Section 4.4. In feature sets χ_1 and χ_2, we initialize the values of all topic-adaptive features to zeros, which only common features take effects. By fixing topic-adaptive features, optimization and adapting to unlabeled data steps alternates until confidence scores of the rest tweets $t_j \in U$ are below threshold τ, or the number of iterations is more than M to control the execution time. In an iteration, at most l unlabeled tweets are selected from each sentiment classes. The iterations between optimization and adapting to unlabeled data agree with the co-training framework. Furthermore, adapting to features is executed for expending at most c topic-adaptive sentiment words and updating user-level and @-network-based feature values. It gradually transfers the sentiment classifier in the aspect of features. Such a step is alternated with the above co-training iterations, until there are no adaptive features can be expanded or updated. Finally, at the end of the algorithm the augmented labeled set L, updated feature values x, x' and a combined sentiment classifier C^* is output. Classifier C^* is trained on the expanded topic-adaptive sentiment words, user-level features, @-network-based features and common features, using the augmented set L.

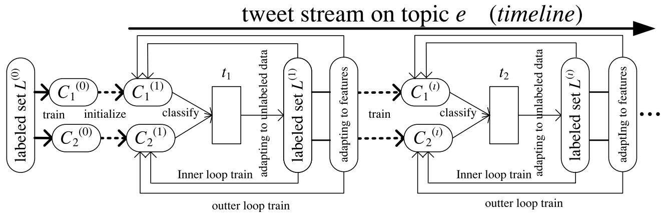


Fig. 3. TASC-t: adapting along a timeline.

4.6 Adapting along a Timeline

In practice, tweets on a topic are not available at one time, which are dynamic and come up in a streaming way. The incoming tweets always bring more significant unlabeled data and topic-adaptive features for transferring the classifier in the future. Thus we modify our TASC algorithm to adapt along a timeline, named TASC-t, which procedure is shown as Fig. 3. For topic e, we start with mixed labeled data $L = L^{(0)}$, and classifiers $C_1^{(1)}$ and $C_2^{(1)}$ are initialized separately by common classifiers $C_1^{(0)}$ and $C_2^{(0)}$. The whole timeline is divided into time intervals equally or with empirical knowledge. Classifiers $C_1^{(1)}$ and $C_2^{(1)}$ are used for collaboratively selecting unlabeled data from topic e in the first interval of a timeline. The selected data are used to augment the labeled data L for the next inner loop co-training of classifiers. Besides adapting to unlabeled data from topic e, we extend topic-adaptive sentiment words to features χ_1 and update user-level and @-network-based feature values at the end of each outer loop training. At the end of first interval, we output the classification results with a combined TASC classifier. In the next interval ι, sentiment classifiers $C_1^{(\iota)}$ and $C_2^{(\iota)}$ are trained on the latest adaptive feature values x and x' using the augmented labeled set $L^{(\iota)}$ inherited from last interval. The inner loops of co-training and outer loops of adapting to features are executed as well. At last, all the tweets in the stream of topic e are classified interval by interval, paying more attention to the local context of dynamic tweets. Therefore, TASC-t can be viewed as an upgraded version for adapting to the evolvement on a topic, while TASC is for adapting common classifier to a specific topic.

5 Experiments

In the experiments, we use the corpuses in Table 1 for evaluation. There are $K = 3$ sentiment class labels in the corpuses, i.e., negative, neutral and positive. In order to show how our algorithm performs with a small amount of labeled data, we randomly sample some ratio p of labeled tweets on a topic, keeping proportions of sentiment classes. The sampled tweets from various topics are then mixed together as initially labeled data set L. And the rest tweets on each topic are used for testing.

5.1 Baselines and Evaluation Metrics

We use the well-known supervised, ensemble and semi-supervised approaches as the baselines.

TABLE 4
@-Network-Based Feature Comparison

$p = 10\%$		without		direct		all	
		MSVM	TASC	TASC	Incr	TASC	Incr
Taco Bell	Acc.	0.6040	0.6143	0.7117	15.85%	**0.7145**	16.31%
	F-s.	0.4422	0.4752	0.6191	30.28%	**0.6264**	31.81%
President	Acc.	0.4463	0.4505	**0.5493**	21.93%	0.5445	20.86%
Debate	F-s.	0.3603	0.3468	**0.5422**	56.35%	0.5383	55.22%

- *DT.* It is a Weka [54] implementation of Decision Tree, which is a tree-like model in which internal node represents test on an attribute.
- *MSVM [55].* It is a multiclass SVM classification based on Structural SVM and it is an instance of SVMstruct.
- *RF.* It is a Weka [54] implementation of Random Forest, which is an ensemble learning method for classification of a multitude of decision trees, and we tune the number of trees to be 10 for better performance.
- *MS3VM.* It is multiclass semi-supervised SVM which is our implementation of augmenting unlabeled tweets without adaptive features.
- *CoMS3VM.* It is MS3VM algorithm in a co-training scheme, by naturally splitting the common features into text and non-text views.

The accuracy, precision, recall and F-score are used for evaluation. As for sentiment class y, let tp_y, fp_y, tn_y and fn_y separately be the numbers of true positive, false positive, true negative and false negative tweets. Thus the accuracy denoted as "Acc" is calculated as follows:

$$Accuracy = \frac{\sum_y (tp_y + tn_y)}{\sum_y (tp_y + fp_y + tn_y + fn_y)}. \tag{14}$$

The precision and recall are averaged for all the sentiment classes, and F-score is the mean value of precision and recall, denoted as "F-s".

5.2 Results and Comparison

5.2.1 Comparison of @-Network-Based Features

In Table 4, we compare TASC without @-network-based features, direct-parent/-child connections, and all-parents/-children connections of @-network-based features, denoted as "without", "direct" and "all" respectively. Since there is no author information in Sanders-Twitter Sentiment Corpus, we only compare the results on topics of "Taco Bell" and "President Debate". It is seen that TASC with @-network-based features achieves accuracy increase by at least 15.85 percent and F-score increase by at least 30.28 percent, compared to "without". Furthermore, "direct" performs a little better than "all" on topic "President Debate", since direct-parent/-child connections may have a very close context and less noise in such cases. In the following experiments, we use all-parents/-children connections of a tweet as its @-network-based features, since sentiment correlations to @-users' overall expressions may have less variances. And such features can also be interpreted as user-level sentiments of @-users, estimated by all their tweets in a context.

TABLE 5
Performance on Different Step Lengths on President Debate

(a)			
$\tau = \theta = 0.5$ $l = 5; c =$	5	10	20
Acc.	0.5311	0.5531	0.5445
Precision	0.5446	0.5716	0.5504
Recall	0.5115	0.5354	0.5268
F-s.	0.5275	0.5529	0.5383
(b)			
$\tau = \theta = 0.5$ $c = 20; l =$	5	10	20
Acc.	0.5445	0.5400	0.5438
Precision	0.5504	0.5456	0.5666
Recall	0.5268	0.5265	0.5217
F-s.	0.5383	0.5359	0.5432

5.2.2 Our Results with Different Step Lengths

There are four controls parameters for TASC algorithm. We try different step lengths l and c separately for selecting unlabeled data and topic-adaptive sentiment words to see how those empirical parameters impact our algorithm. According to the definition of confidence and significance scores, the thresholds τ and θ take effects when larger than $1/K \approx 0.33$. As shown in Table 5a, with step length l fixed and thresholds $\tau = \theta = 0.5$, different step length c of expanding topic-adaptive words are tested for TASC algorithm. Table 5b shows the performances with step length $c = 20$ and different l for unlabeled data selection. We can see that step lengths c and l are not sensitive in our TASC learning algorithm with the guarantee of thresholds. In the following experiments, we choose thresholds $\tau = \theta = 0.5$, step length $l = 5$ and $c = 20$ together to control the quality of selected data and topic-adaptive words.

5.2.3 Different Sample Ratios p

To show the adaptive ability, we test our TASC algorithm with labeled set L in different sample ratios p. Table 6 lists the accuracies and F-scores of classification results on the six topics. The average accuracy and F-score increases of TASC are separately reported compared to MSVM, i.e., the initially common classifier. It is seen that TASC averagely increases by 11.76 and 40.18 percent in accuracy and F-score separately, with sample ratio $p = 1\%$ and labeled data size $|L| = 113$. The best accuracies and F-scores among different sample ratios on each topic are given in boldfaced digits. It is seen that TASC does not necessarily perform better when sampling out more tweets from the six topics as training data. Such a result implies that more labeled data mixed from a small number of topics can on the one hand benefit TASC by introducing more supervised information, and on the other hand hurt it by making TASC bias to some specific topics.

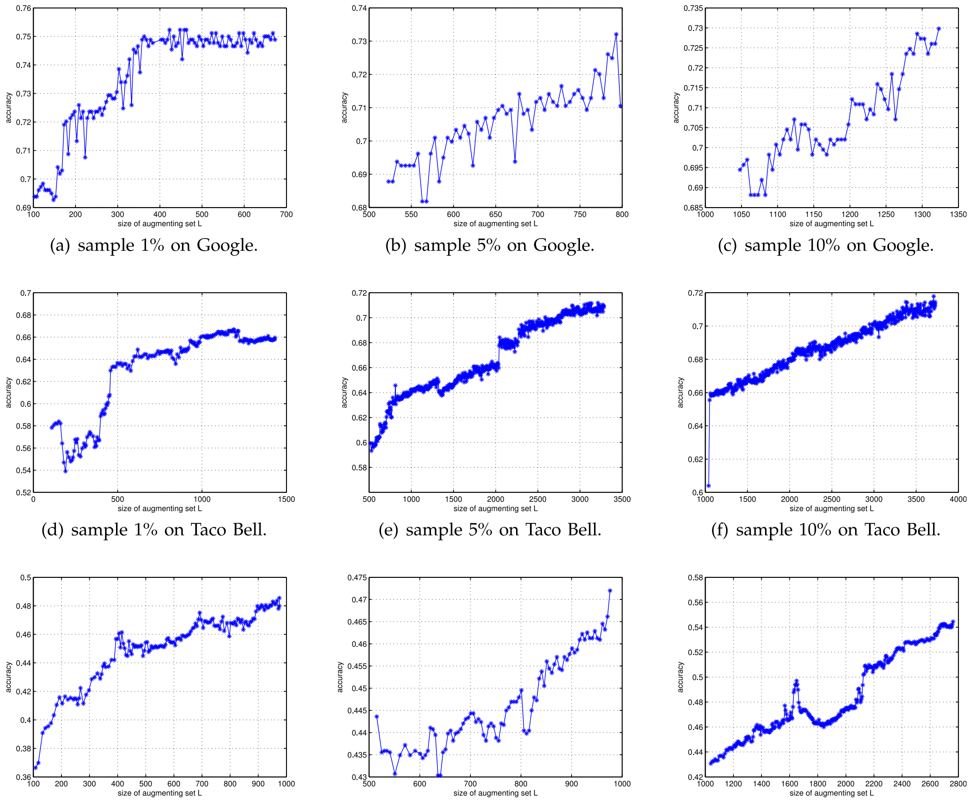
The accuracy curves with the size of augmenting set L in iterations are illustrated in Fig. 4 for the runs with ratios $p = 1\%, 5\%, 10\%$ on Google, Taco Bell and President Debate. It intuitively shows the accuracy increases with each iteration of optimization, adapting to unlabeled data and adapting to features.

TABLE 6
Performance on Different Sample Ratios

$p =$		1%		5%		10%		20%	
$\|L\|$		113		523		1,048		2,099	
		MSVM	TASC	MSVM	TASC	MSVM	TASC	MSVM	TASC
Apple	Acc.	0.5089	0.5508	0.5146	0.5693	0.5284	0.5559	0.5424	**0.5700**
	F-s.	0.4196	0.4660	0.4340	0.5110	0.4356	0.4793	0.4812	**0.5275**
Google	Acc.	0.6938	**0.7489**	0.6877	0.7105	0.6944	0.7300	0.6946	0.7230
	F-s.	0.4033	0.5598	0.4852	0.5225	0.4865	0.5696	0.4813	**0.5931**
Microsoft	Acc.	0.7412	0.7593	0.7420	**0.7608**	0.7339	0.7500	0.7395	0.7534
	F-s.	0.4585	**0.5512**	0.4536	0.5254	0.4484	0.5039	0.4593	0.5178
Twitter	Acc.	0.8111	**0.8568**	0.8127	0.8317	0.8065	0.8252	0.8096	0.8242
	F-s.	0.4316	**0.6704**	0.4546	0.5701	0.4175	0.6646	0.4730	0.5222
Taco Bell	Acc.	0.5783	0.6581	0.5994	0.7083	0.6040	**0.7145**	0.6278	0.7109
	F-s.	0.3436	0.5518	0.4321	0.6096	0.4422	**0.6264**	0.5248	0.6162
President	Acc.	0.3664	0.4855	0.4436	0.4720	0.4307	**0.5445**	0.4684	0.4851
Debate	F-s.	0.3147	0.4881	0.3394	0.4788	0.4170	**0.5383**	0.4187	0.4997
Average	Acc.	–	11.76%	–	7.23%	–	9.93%	–	4.94%
Incr	F-s.	–	40.18%	–	24.80%	–	28.24%	–	15.46%

5.2.4 *Comparisons Between Baseline*

We compare our adaptive algorithm TASC with five baseline algorithms, i.e., DT, MSVM, RF, MS3VM and CoMS3VM in Table 7. We sample five times with $p = 10\%$ from the whole data resulting in five labeled and testing data sets for repeated runs. The means of accuracies and F-scores of five runs are given separately for each baseline, and the standard deviations are listed below. The results report that TASC outperforms other baselines in mean accuracies on all the topics. Although ensemble learning method RF achieves best mean F-scores on three topics of IT companies, it loses the advantages on "Taco Bell" and "President Debate". Therefore, our TASC model achieves more reliable performances than the baselines. In addition, we simulate the realtime characteristic of tweets, and divide the tweets by post time into different time intervals. With the



(a) sample 1% on Google.
(b) sample 5% on Google.
(c) sample 10% on Google.
(d) sample 1% on Taco Bell.
(e) sample 5% on Taco Bell.
(f) sample 10% on Taco Bell.
(g) sample 1% on President Debate.
(h) sample 5% on President Debate.
(i) sample 10% on President Debate.

Fig. 4. Accuracy curves with the size of augmenting set L during iterations of TASC algorithm.

TABLE 7
Comparisons with Baselines in 10% Sample Ratio

Topics	DT		MSVM		RF		MS3VM		CoMS3VM		TASC		TASC-t	
	Acc	F-s.	Acc.	F-s.	Acc.	F-s.	Acc.	F-s.	Acc.	F-s.	Acc	F-s.	Acc	F-s.
Apple	0.5063 ±0.0000	0.3400 ±0.0000	0.5036 ±0.0106	0.4624 ±0.0085	0.5403 ±0.0161	0.5063 ±0.0123	0.5116 ±0.0205	0.4344 ±0.0283	0.6795 ±0.0126	0.6440 ±0.0133	**0.6882** ±0.0114	**0.6528** ±0.0123	0.5461	0.4617
Google	0.6835 ±0.0000	0.5550 ±0.0000	0.7016 ±0.0068	0.6386 ±0.0141	0.7614 ±0.0255	**0.7414** ±0.0301	0.7614 ±0.0168	0.6350 ±0.0142	0.7662 ±0.0124	0.6201 ±0.0200	**0.7725** ±0.0119	0.6371 ±0.0218	0.7054	0.5155
Microsoft	0.7429 ±0.0000	0.6330 ±0.0000	0.7300 ±0.0186	0.6716 ±0.0057	0.7411 ±0.0123	**0.6894** ±0.0156	0.7315 ±0.0089	0.4355 ±0.0141	0.7884 ±0.0171	0.6058 ±0.0400	**0.7896** ±0.0176	0.6072 ±0.0363	0.7416	0.4809
Twitter	0.8112 ±0.0000	0.7270 ±0.0000	0.7976 ±0.0021	0.7260 ±0.0020	0.8097 ±0.0148	**0.7645** ±0.0065	0.8054 ±0.0086	0.5226 ±0.0108	0.8126 ±0.0157	0.5343 ±0.0225	**0.8176** ±0.0165	0.5472 ±0.0240	0.8196	0.4867
Taco Bell	0.5836 ±0.0000	0.4300 ±0.0000	0.6500 ±0.0075	0.5796 ±0.0120	0.5976 ±0.0088	0.5744 ±0.0073	0.6974 ±0.0240	0.5911 ±0.0463	0.7105 ±0.0034	0.6181 ±0.0077	**0.7126** ±0.0015	**0.6206** ±0.0058	0.7151	0.6297
President Debate	0.3845 ±0.0047	0.2422 ±0.0624	0.4365 ±0.0081	0.4228 ±0.0057	0.4848 ±0.0103	0.4858 ±0.0107	0.5167 ±0.0289	**0.5189** ±0.0212	0.5185 ±0.0287	0.5162 ±0.0240	**0.5216** ±0.0289	0.5175 ±0.0246	0.5901	0.5824

evaluation of TASC-t, the results are listed in the last two columns in Table 7. It is surprising to see that TASC-t achieves better accuracies, i.e., 0.8196, 0.7151 and 0.5901, than the mean values by TASC on the last three topics. It gives the evidence that TASC-t may get some benefits when adding unlabeled tweets and @-network connections in chronological order.

At last we illustrate some of the initial common sentiment words, and the expanded topic-adaptive sentiment words of some iterations in Table 8 on the topic of "President Debate". With topic-adaptive sentiment words from each iteration, we can easily grasp the adaption procedure to the sub-topics under "President Debate". For example, in the third expansion, the model adapts more to a sub-topic "Financial Crisis" with "rapid" and "top" as positive, and "again", "less", and "far" as negative expressions on financial recovery. And Table 9 shows the augmented tweets for different sentiment classes on the topic of "Apple". It is seen that the steps of adaptive feature expansion and unlabeled data selection pick reasonable topic-adaptive sentiment words and unlabeled tweets from positive, negative, and neutral classes in a semi-supervised way.

5.3 Timeline Visualization

TASC or TASC-t outputs the probabilities that a tweet t_i belongs to each class, denoting as p_i, m_i and n_i separately for positive, neutral and negative classes. The probabilities are separately the normalized confidence scores for each class, satisfying $p_i + m_i + n_i = 1$. The higher probability can reflect stronger intensity of a sentiment polarity that tweet t_i expresses.

5.3.1 Visualization Graph

We design a kind of "river" graph to visualize the intensities, ups and downs of sentiment polarities on a topic. The graph is composed of layers, with each one being colored and representing a sentiment class. The layers are ordered from the positive down to the negative with the neutral in the middle. And the geometries and colors of layers are defined in the following.

First, we define density function q_k as the distribution of the number of tweets belonging to sentiment class k. The parameter of function q_k is time, and its integral of the whole life cycle equals to the total number of tweets of class k. Thus the upper boundary of layer geometry, corresponding to sentiment class k, is calculated as follows:

TABLE 8
Adaptive Sentiment Words Expanded on the Topic of President Debate

sentiment class	initial	first expansion	second expansion	third expansion	fourth expansion	fifth expansion
positive	beautiful moderate sound adorable awesome	totally impressed uneventful condescending strongest	directly presidential debate dumb federal	defensive rapid top economic ready	overall OK defensive current much	enough middle very most over
neutral	want completely sustained understanding keep	next never instead live last	different probably post-debate tortured own	terrorist specific disrespectful polar ever	republican kinda especially sudden crazy	ahead present public bizarre yeah
negative	hate black alarming stupid dislike	tired No Second Global short	overall emotional corporate professorial dead	national again there far less	up ago maybe anymore nicely	away forward anyway wow social

TABLE 9
Selected Unlabeled Tweets on the Topic of Apple

tweets	class
Today I was introduced as BigDealDawson at #LGFW ! O #twitter and #social media I love you! Teehee xx	positive
"I'm starting to get really concerned, sending hashtags in emails :P #twitter is taking over our lives :D"	
I've pretty much abandoned Facebook for Twitter. #twitter'slegit	
Gotta love #Twitter—shit goes round the World like lightning-on-speed. . .	
#I #am #so #good #at #twitter ;)	
@codytigernord Just a reminder that you fail on #twitter	neutral
"And by the way, why did my iPhone 4 have to loose #Siri to get #Twitter?"	
Going in :(#work. Break at 2:30 and 5:30 #twitter time. See ya'll in the #am	
My #twitter age is 1 year 19 days 13 hours 37 minutes 17 seconds. Find out yours at http://t.co/XhRUA9Dz #twittertime	
@D_REALRogers BE SLEEP N WHO NOEZ WHERE I BE BUT HOW IT's it U so called sleep but every 5 sec u got a new #twitter post up #btfu	
RT @FuckingShinez: #Twitter = #Dead "this is why im never on it now"	negative
Just hit my hourly usage limit on #twitter. How does that even happen? All I'm doing is listing people . . . and I was almost done! #ugh	
"RT @mainey_maine: RT @ItalianJoya i better be able to see my RT's tomorrow #twitter and tell that lil blue ass bird, (cont) http://t.co/xGHoev8k"	
Not really. . . I rather study my notes than studying #twitter. . .	
#Twitter are you freaking kidding me #wth . . . http://t.co/zKn2bu5R	

$$S_k = -\frac{1}{2}\sum_{j=1}^{n} q_j + \sum_{j=1}^{k} q_j. \tag{15}$$

And S_{k-1} is the lower boundary of the layer. Let n be the number of classes. The bottom and top curve functions of the whole "river" graph are separately S_0 and S_n,

$$S_0 = -\frac{1}{2}\sum_{j=1}^{n} q_j \quad \text{and} \quad S_n = \frac{1}{2}\sum_{j=1}^{n} q_j.$$

It is seen that the graph is symmetric along x-axis of time.

Second a map function (16) between colors and sentiment probabilities is proposed,

$$RGB(t_i) = \begin{cases} ((1-p_i)\cdot 255, 255, 0)) & p_i \geq n_i \\ (255, (1-n_i)\cdot 255, 0) & p_i < n_i, \end{cases} \tag{16}$$

where the additive RGB color model is adopted, in which red, green, and blue are added together in various ways to reproduce a broad array of colors. That's to say, each color is determined by three parameters as a triple. To color the layers for showing sentiment intensities, we divide the sentiment probabilities p_i and n_i into fine-grained segments, and use the mid-value of each segment to calculate its color by equation (16). It is seen that $RGB(t_i)$ is only determined by positive and negative probabilities. When tweet t_i is neutral, i.e., m_i is the maximal, we also sub-classify it into different intensities of positively neutral, purely neutral and negatively neutral. To make the function value continuous, when p_i equals to n_i, we simply set the $m_i = 1$, and $p_i = n_i = 0$. It agrees with our common sense that if the probabilities to positive and negative classes become equal, the tweet should belong to a neutral class. And the middle layer color is pure yellow with a RGB triple $(255, 255, 0)$.

At last, we choose the sentiment word labels by considering the frequency $f(\bar{w})$ and feature values $x_{\bar{w}}$ of word $\bar{w}$. The font size $F(\bar{w})$ in the "river" graph is determined by $F(w) \propto x_{\bar{w}} f(\bar{w})$.

5.3.2 Visualizing Results

We use "river" graph to show the sentiment evolvement on "Present Debate" by TASC-t in Fig. 5. It is seen that we could easily grasp the ups and downs of sentiment trends along the timeline. The sentiment intensity is shown intuitively by the gradation of color from green to red, passing through yellow. The debate started from 1:00 GMT, and lasted for 97 minutes. After 2:37 GMT, the corpus lasted for another 53 minutes. We can see that there is a wide current in the first half hours. And there actually were discussions on solving financial crisis during that time, which attracted participants expressing their sentiments online. In the full view of the timeline, people have paid more attention to the debate, and less to post their tweets online during the live debate. And a large-scale discussions bursted in Twitter just after that, which was grasped by the suddenly width change of "river" graph after 2:37 GMT. It is also interesting to see that more people expressed negative tweets in the period.

6 Conclusions

Diverse topics are discussed in Microblog services. Sentiment classifications on tweets suffer from the problems of lack of adapting to unpredictable topics and labeled data, and extremely sparse text. Therefore we formally propose

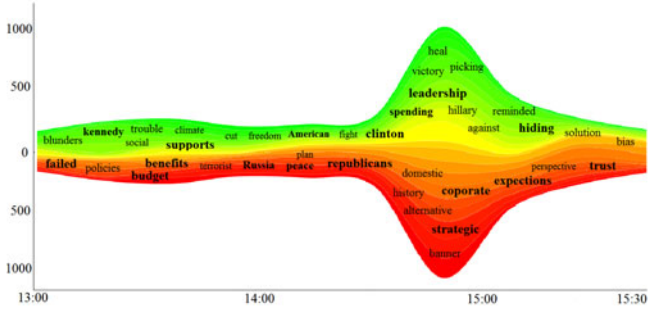


Fig. 5. The timeline visualization results of President Debate.

an adaptive multiclass SVM model in co-training scheme, i.e., TASC, transferring an initial common sentiment classifier to a topic-adaptive one by adapting to unlabeled data and features. TASC-t is designed to adapt along a timeline for the dynamics of tweets. Compared with the well-known baselines, our algorithm achieves promising increases in mean accuracy on the six topics from public tweet corpuses. Besides a well-designed visualization graph is demonstrated in the experiments, showing its effectiveness of visualizing the sentiment trends and intensities on dynamic tweets.

Acknowledgments

This work was funded by the National Basic Research Program of China (No. 2012CB316303, 2014CB340401), National Natural Science Foundation of China (No. 61202213, 61232010, 61472400, 61100083), National Key Technology R&D Program (2012BAH46B04). S. Liu is the corresponding author.

References

[1] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," *J. Am. Soc. Inform. Sci. Technol.*, vol. 60, no. 11, pp. 2169–2188, 2009.

[2] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Micro-blogging as online word of mouth branding," in *Proc. Extended Abstr. Human Factors Comput. Syst.*, 2009, pp. 3859–3864.

[3] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, 2011.

[4] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proc. 4th Int. AAAI Conf. Weblogs Soc. Media*, 2010, vol. 10, pp. 178–185.

[5] L. T. Nguyen, P. Wu, W. Chan, W. Peng, and Y. Zhang, "Predicting collective sentiment dynamics from time-series social media," in *Proc. 1st Int. Workshop Issues Sentiment Discovery Opinion Mining*, 2012, p. 6.

[6] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," *J. Am. Soc. Inform. Sci. Technol.*, vol. 62, no. 2, pp. 406–418, 2011.

[7] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proc. Workshop Lang. Soc. Media*, 2011, pp. 30–38.

[8] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lect. Human Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[9] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li, "User-level sentiment analysis incorporating social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1397–1405.

[10] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, vol. 7, pp. 440–447.

[11] F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain co-extraction of sentiment and topic lexicons," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics: Long Papers*, 2012, pp. 410–419.

[12] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 751–760.

[13] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff, "Overview of the trec-2011 microblog track," in *Proc. 20th Text Retrieval Conf.*, 2011, http://trec.nist.gov/pubs/trec20/t20.proceedings.html

[14] I. Soboroff, I. Ounis, J. Lin, and I. Soboroff, "Overview of the trec-2012 microblog track," in *Proc. 21st Text REtrieval Conf.*, 2012.

[15] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Computer Science Department, Stanford, USA, pp. 1–12, 2009.

[16] S. Li, C.-R. Huang, G. Zhou, and S. Y. M. Lee, "Employing personal/impersonal views in supervised and semi-supervised sentiment classification," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 414–423.

[17] S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee, "Semi-supervised learning for imbalanced sentiment classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1826–1831.

[18] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Language Process. AFNLP: Volume 1-Volume 1*, 2009, pp. 235–243.

[19] N. Yu and S. Kübler, "Filling the gap: Semi-supervised learning for opinion detection across domains," in *Proc. 15th Conf. Comput. Natural Language Learn.*, 2011, pp. 200–209.

[20] K. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Adv. Neural Inform. Proc. Syst.*, 1999, pp. 368–374.

[21] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machines for unlabeled data classification," *Optim. Methods Softw.*, vol. 15, no. 1, pp. 29–44, 2001.

[22] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.

[23] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. 9th Int. Conf. Inform. Knowl. Manage.*, 2000, pp. 86–93.

[24] S. Liu, W. Zhu, N. Xu, F. Li, X.-Q. Cheng, Y. Liu, and Y. Wang, "Co-training and visualizing sentiment evolvement for tweet events," in *Proc. 22nd Int. Conf. World Wide Web Companion*, 2013, pp. 105–106.

[25] Y. He, C. Lin, and H. Alani, "Automatically extracting polarity-bearing topics for cross-domain sentiment classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Language Technol.-Volume 1*, 2011, pp. 123–131.

[26] S. Gao and H. Li, "A cross-domain adaptation method for sentiment classification using probabilistic latent analysis," in *Proc. 20th ACM Int. Conf. Inform. Knowl. Manage.*, 2011, pp. 1047–1052.

[27] V. Stoyanov and C. Cardie, "Topic identification for fine-grained opinion analysis," in *Proc. 22nd Int. Conf. Comput. Linguistics*, 2008, pp. 817–824.

[28] D. Das and S. Bandyopadhyay, "Emotion co-referencing-emotional expression, holder, and topic," *Int. J. Comput. Linguistics Chin. Language Process.*, vol. 18, no. 1, pp. 79–98, 2013.

[29] D. Das and S. Bandyopadhyay, "Extracting emotion topics from blog sentences: Use of voting from multi-engine supervised classifiers," in *Proc. 2nd Int. Workshop Search Mining User-Generated Contents 19th ACM Int. Conf. Inform. Knowl. Manage.*, 2010, pp. 119–126.

[30] D. Das and S. Bandyopadhyay, "Identifying emotion topican unsupervised hybrid approach with rhetorical structure and heuristic classifier," in *Proc. 6th Int. Conf. Natural Language Process. Knowl. Eng.*, 2010, pp. 1–8.

[31] F. Li, S. Wang, S. Liu, and M. Zhang, "Suit: A supervised user-item based topic model for sentiment analysis," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1636–1642.

[32] Y. Mejova and P. Srinivasan, "Crossing media streams with sentiment: Domain adaptation in blogs, reviews and twitter," in *Proc. 6th Int. AAAI Conf. Weblogs Soc. Media*, 2012, pp. 234–241.

[33] S. Liu, F. Li, F. Li, X. Cheng, and H. Shen, "Adaptive co-training SVM for sentiment classification on tweets," in *Proc. 22Nd ACM Int. Conf. Inform. &#38; Knowl. Manage.*, 2013, pp. 2079–2088.

[34] K.-L. Liu, W.-J. Li, and M. Guo, "Emoticon smoothed language models for twitter sentiment analysis." in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1678–1684.

[35] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *Proc. 5th Int. Conf. Weblogs Soc. Media*, 2011, pp. 538–541.

[36] R. Mehta, D. Mehta, D. Chheda, C. Shah, and P. M. Chawan, "Sentiment analysis and influence tracking using twitter," *Int. J. Adv. Res. Comput. Sci. Electron. Eng.*, vol. 1, no. 2, p. 72, 2012.

[37] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 417–424.

[38] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach," in *Proc. 20th ACM Int. Conf. Inform. Knowl. Manage.*, 2011, pp. 1031–1040.

[39] X. Wang, J. Jia, P. Hu, S. Wu, J. Tang, and L. Cai, "Understanding the emotional impact of images," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 1369–1370.

[40] J. Jia, S. Wu, X. Wang, P. Hu, L. Cai, and J. Tang, "Can we understand van gogh's mood?: Learning to infer affects from images in social networks," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 857–860.

[41] Y. Zhang, J. Tang, J. Sun, Y. Chen, and J. Rao, "MoodCast: Emotion prediction via dynamic continuous factor graph model," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 1193–1198.

[42] S. Havre, E. G. Hetzler, and L. T. Nowell, "ThemeRiver: Visualizing theme changes over time," in *Proc. IEEE Symp. Inform. Vis.*, 2000, pp. 115–124.

[43] S. L. Havre, E. G. Hetzler, P. D. Whitney, and L. T. Nowell, "ThemeRiver: Visualizing thematic changes in large document collections," *IEEE Trans. Visualization Comput. Graph.*, vol. 8, no. 1, pp. 9–20, Jan. 2002.

[44] L. Byron and M. Wattenberg, "Stacked graphs—geometry & aesthetics," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1245–1252, Nov. 2008.

[45] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang, "TIARA: A visual exploratory text analytic system," in *Proc. Knowl. Discovery Data Mining*, 2010, pp. 153–162.

[46] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu, "OpinionSeer: Interactive visualization of hotel customer feedback," *IEEE Trans. Visualization Comput. Graph.*, vol. 16, no. 6, pp. 1109–1118, Nov. 2010.

[47] M. Hao, C. Rohrdantz, H. Janetzko, U. Dayal, D. A. Keim, L.-E. Haug, and M.-C. Hsu, "Visual sentiment analysis on twitter data streams," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2011, pp. 277–278.

[48] D. Das, A. K. Kolya, A. Ekbal, and S. Bandyopadhyay, "Temporal analysis of sentiment events–a visual realization and tracking," in *Computational Linguistics and Intelligent Text Processing*. New York, NY, USA: Springer, 2011, pp. 417–428.

[49] N. A. Diakopoulos and D. A. Shamma, "Characterizing debate performance via aggregated twitter sentiment," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2010, pp. 1195–1198.

[50] C. Strapparava and A. Valitutti, "Wordnet affect: An affective extension of wordnet," in *Proc. 4th Int. Conf. Lang. Resources Eval.*, 2004, vol. 4, pp. 1083–1086.

[51] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2004, pp. 168–177.

[52] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A large-scale sentiment analysis for yahoo! answers," in *Proc. 5th ACM Int. Conf. Web Search Data Min.*, 2012, pp. 633–642.

[53] M. Collins and Y. Singer, "Unsupervised models for named entity classification," in *Proc. Joint SIGDAT Conf. Empirical Methods Natural Lang. Process. Very Large Corpora*, 1999, pp. 189–196.

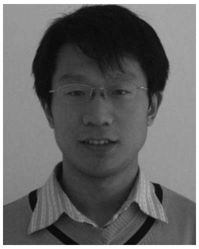
[54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[55] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2002.

Shenghua Liu graduated from Tsinghua University and is an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include data mining, social network, and sentiment analysis.

Xueqi Cheng is a professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include network science and social computing, web search and mining.

Fuxin Li is currently working towards the MS degree at the Institute of Computing Technology, Chinese Academy of Sciences. His research interest is data mining.



Fangtao Li received the PhD degree at Tsinghua University, and is currently a R&D at Google Inc. His research interests are computational linguistics, applied machine learning, and social network.