Modeling Sentiment Evolution for Social Incidents*

Yunjie Wang
Xiamen University
23020171153058@stu.xmu.edu.cn

Hui Li
Xiamen University
hui@xmu.edu.cn

Chen Lin
Xiamen University
chenlin@xmu.edu.cn

ABSTRACT

Modeling sentiment evolution for social incidents in Microblogs is of vital importance for both enterprises and government officials. Existing works on sentiment tracking are not satisfying, due to the lack of entity-level sentiment extraction and accurate sentiment shift detection. Identifying entity-level sentiment is challenging as Microbloggers often use multiple opinion expressions in a sentence which targets different entities. Moreover, the evolution of the background sentiment, which is essential to shift detection, is ignored in the previous study. To address these issues, we leverage the proximity information to obtain more precise entity-level sentiment extraction. Based on it, we propose to simultaneously model the evolution of background opinion and the sentiment shift using a state space model on the time series of sentiment polarities. Experiments on real data sets demonstrate that our proposed approaches outperform state-of-the-art methods on the task of modeling sentiment evolution for social incidents.

CCS CONCEPTS

• **Information systems → Social networks**.

KEYWORDS

Sentiment Tracking, Dynamic Sentiment Model, Opinion Analysis, Microblog Mining

ACM Reference Format:

Yunjie Wang, Hui Li, and Chen Lin. 2019. Modeling Sentiment Evolution for Social Incidents. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM’19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358136

1 INTRODUCTION

Nowadays, Microblogging has become the primary platform for people to publish information and share opinions about social incidents. Public opinions about social events on Microblogging platforms have greatly influenced society, changed the results of the inadequate investigation and unjust sentence [1]. The power of public opinion in Microblogging space makes it appealing to analyze sentiment evolution for social incidents in microblogs for enterprises and government officials.

To facilitate the understanding of public opinions, we study the problem of modeling sentiment evolution for social incidents in this paper. Given a sequence of microblogging comments related to any social incident, our goal is to reveal the *sentiment evolution pattern* related to the *involved entities* in this incident and identify *significant sentiment shifts*. Recently there is an increasing interest in tracking microblogging sentiments for entities [2] or topics [3]. However, identifying entity-level sentiment and detecting sentiment shift for modeling sentiment evolution of social incidents are two challenges that have not been addressed so far. A social incident often involves several entities (i.e. people or organizations), and it is problematic to utilize coarse-grained analysis which obtains an averaged sentiment for an event without separating different entities. Moreover, conventional approaches leverage statistical analysis such as outlier detection to detect sentiment spikes when modeling sentiment evolution [2]. Such methods are not sufficient for detecting sentiment shift since the evolution of the background is largely overlooked. Events are continuously changing, which causes the changes of responses in public opinions at the same time. Consequently, sentiment shift should be distinguished together with the evolution patterns of background sentiment.

To handle the first problem, we propose the *Proximity-based Entity-level Sentiment Extraction* (PESE) to embed the proximity information when extracting entity-level sentiment and enhance accuracy. For the second issue, we design a probabilistic model called *Public Sentiment Evolution Model (PSEM)* which simultaneously models the evolution of background opinion and the opinion shifts. Our contributions are two folds. In the application aspect, we explore the feasibility of tracking sentiment evolution for social incidents on microblogs. Our work sheds insights into better understanding public opinions and provides a solid foundation for future applications such as explaining the causes of sentiment shifts. In the model aspect, we investigate the impact of proximity information in obtaining entity-level sentiment extraction. Based on it, we propose to simultaneously model the evolution of background sentiment and sentiment shift by state space models on the natural parameters of the binomial distributions that represent the sentiment polarity. Experimental results demonstrate that our proposal shows promising performance on the task of modeling sentiment evolution for social incidents.

2 PROXIMITY-BASED ENTITY-LEVEL SENTIMENT EXTRACTION

In this section, we describe our proposed Proximity-based Entity-level Sentiment Extraction (PESE) which leverages the proximity information to obtain more precise entity-level sentiment extraction.

*The work is supported by the National Natural Science Foundation of China (no.61472335, 61972328).

CIKM ’19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358136

Firstly, we formulate the task of entity-level sentiment extraction. Suppose we have a collection of incident relevant microblog posts $O = \{o\}$. As an incident involves several entities $E = \{e\}$, we can represent each post o as a set of sentiment triples and entity triples, $o = \{(w_i, l_i, v_i)\} \bigcup \{(e_j, l_j)\}$. In the sentiment triple (w_i, l_i, v_i), i is the index for sentiment words, $w_i \in D$ is the word which is extracted from the sentiment lexicon D, l_i is the location of sentiment words and v_i is the sentiment value of w_i. In the entity triples (e_j, l_j), j is the index for entity occurrences, $e_j \in E$ is the name of the entity, l_j is the location of the entity in the post. The aim of entity-level sentiment extraction is to output the sentiment polarity y for the entity e_j in the post $y_{e_j}(o) \in \{0, 1\}$, where 0 represents a negative emotion, and 1 otherwise.

The basic assumption of PESE is that the position of a sentiment word influences the performance of entity-level sentiment extraction. Intuitively, the closer a sentiment word is to an entity, the more likely that the sentiment word is describing the entity. Inspired by [4], given two locations l_i and l_j, the proximity information can be captured by four distance kernel functions to compute the influence of sentiment words on entities, namely Gaussian, Triangle, Cosine (Hamming), and Circle:

- **Gaussian:** $k(l_i, l_j) = \exp\left[\frac{-(l_i - l_j)^2}{2\sigma^2}\right]$
- **Triangle:** $k(l_i, l_j) = \begin{cases} 1 - \frac{|l_i - l_j|}{\sigma} & \text{if } |l_i - l_j| \le \sigma \\ 0 & \text{otherwise} \end{cases}$
- **Cosine:** $k(l_i, l_j) = \begin{cases} \frac{1}{2}\left[1 + cos\left(\frac{|l_i - l_j| \cdot \pi}{\sigma}\right)\right] & \text{if } |l_i - l_j| \le \sigma \\ 0 & \text{otherwise} \end{cases}$
- **Circle:** $k(l_i, l_j) = \begin{cases} \sqrt{1 - \left(\frac{|l_i - l_j|^2}{\sigma}\right)} & \text{if } |l_i - l_j| \le \sigma \\ 0 & \text{otherwise} \end{cases}$

Note that all four of these kernel functions are governed by one parameter σ, which is tuned in the experiment. To classify the polarity $y_{e_j}(o)$ of sentiment towards an entity e_j in a post o, PESE first obtains an entity-level sentiment value by calculating the average of the influence on the entity from different sentiment words. Then, PESE incorporates the information of negative words and adverbs of degree to output the sentiment polarity value s:

$$s = \frac{\sum_{i=1}^{N} (-1)^{q_i} \cdot z_i \cdot v_i \cdot k(l_i, l_j)}{N}, \tag{1}$$

where q_i is the number of negative words, z_i is the sum of degree value (it can be obtained from the lexicon) for adverbs of degree between $(i-1)^{th}$ sentiment word and i^{th} sentiment word, v_i is the sentiment value of ith sentiment word, l_i is the location of i^{th} sentiment word, l_j is the location of entity e_j, N is the number of sentiment words in this sentence. If the sentiment value $s > 0$, the sentiment polarity towards entity e_j in post o is positive, i.e., $y_{e_j}(o) = 1$. if the sentiment value $s < 0$, the sentiment polarity towards entity e_j in post o is negative. i.e., $y_{e_j}(o) = 0$.

3 PUBLIC SENTIMENT EVOLUTION MODEL

Now we describe in detail the Public Sentiment Evolution Model (PSEM) which utilizes PESE. PSEM models the evolution of background opinion and the sentiment shift in parallel.

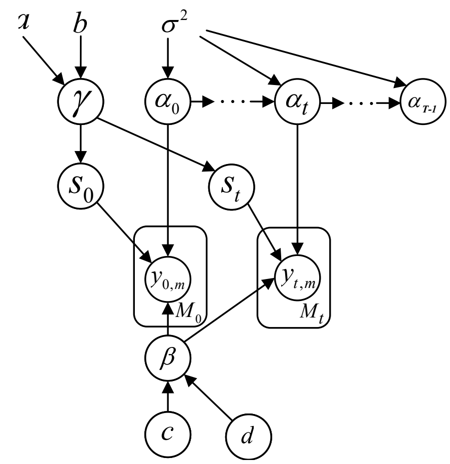


Figure 1: Plate notation of the proposed PSEM model

For a social incident which involves several entities $E = \{e\}$, PSEM first groups the posts based on entities and divides the collection of posts associated with an entity into T time slices. Suppose there are M_t posts in each time slice t, where each post m is preprocessed by the PESE to observe an entity-level sentiment polarity $y_{t,m} \in \{0, 1\}$.

Then, PSEM is built for each entity based on the following two assumptions: (1) There is a background sentiment distribution, i.e., how users typically react to the entity. (2) The background is smoothly and slowly changing, and the evolution of background sentiment distribution is modeled by a dynamic state model. To handle the situation when a sudden shift on public opinions appears (e.g., a sentiment shift is triggered by a new piece of evidence), PSEM leverages a switch variable to simulate the trigger. If the switch is on, the observed sentiment is drawn from the background sentiment distribution. Otherwise, it is drawn from the distribution of "outlier" sentiment.

Let α_t denote the distribution of background opinion specific to time slice t. The generation process of PSEM (shown in Fig. 1) is as follows:

- For time slice $t = 0$, draw $\alpha_0 \sim \mathcal{N}(0, \sigma^2\mathbf{I})$.
- For time slice $t = 1 : T-1$, draw $\alpha_t \sim \mathcal{N}(\alpha_{t-1}, \sigma^2)$. As $\mathcal{N}$ is a continuous and differentiable distribution, the evolution of background opinions is smooth and slow.
- Generate a global prior γ for the switch s_0, i.e. a variable that controls how likely the public sentiment is to change, by $\gamma \sim Beta(a, b)$.

- For each time slice
  - Generate a switch $s_t \sim Bern(\gamma)$
  - For each observation, generate
    $y_{t,m} \sim \begin{cases} Bern\big(\pi(\alpha_t)\big) & \text{if } s_t = 1 \\ Bern(\beta) & \text{if } s_t = 0 \end{cases}$

Note that α is a vector of real numbers, i.e., $\alpha \in \mathcal{R}^2$. To generate the opinion distribution, we adopt $\pi(\cdot)$ function to convert α to a value within $(0, 1)$, $\pi(\alpha_t) = \frac{\exp(\alpha_{t,0})}{\exp(\alpha_{t,0}) + \exp(\alpha_{t,1})}$, where $\alpha_{t,0}$ indicates the first dimension of α_t.

Table 1: Statistics of the data set

No.	Abbreviation	#Posts	Description
1	Jiang Ge Murder	368,037	Chinese female student Jiang Ge was killed in Japan.
2	Maternal in Yulin Fall	35,081	A maternal woman in Yulin jumped to death. Family members and hospitals blame each other.
3	RYB Kindergarten Child Abuse	35,927	Beijing RYB kindergarten was accused of abuse, and the internet erupted in Fury.
4	Nanny Arson	167,225	Hangzhou nanny sets house on fire, killing a mother and her three children.
5	Yu Huan Murder	17,607	Yu Hunan killed a creditor who had harassed his mother.
6	Death of Wei Zexi	59,501	Wei Zexi died because of the fake medical information from Baidu.
7	Unqualified Vaccine scandal	20,682	Changsheng Bio-tech was found to have falsified data and sold ineffective vaccines for children.
8	MH370 Missing	65,585	Malaysia Airlines Flight 370 disappeared on 8 March 2014 while flying to Beijing.
9	DIdi driver murder	45,965	A woman was raped and killed allegedly by a Didi driver.
10	Bus crashing into the river	13,326	A bus in Chongqing plunged off a bridge killing 15 after woman attacks the driver.
11	Programmer Suicide	3,242	A Beijing programmer suicide by jumping from the top of an apartment after becoming depressed during his acrimonious divorce.
12	Professor sexual harassment	1,276	A professor accused of sexually harassing students has been removed from teaching posts.
13	Tiger Attack	35,830	A mother was attacked by a tiger when aiding her daughter who left the vehicle and was exposed to monsters.
14	Beijing Hotel Attack	19,824	A man attacked a woman inside the Yitel Hotel in Beijing.
15	Germanwings Plane Crash	70,188	A young pilot crashed a German airliner into the remote French Alps.
16	Nepal Earthquake	401,889	A severe earthquake struck near the city of Kathmandu in central Nepal on April 25, 2015. About 9,000 people were killed.
17	Paris Attack	732,145	Terror attacks in Paris killed 130 people and wounded 494.
18	UK Leaving The EU	67,481	United Kingdom voted to leave the EU in a bitterly fought referendum in June 2016.
19	Brussels Airport Explosion	184,783	Two suicide bombers, carrying explosives in large suitcases, attacked a departure hall at Brussels Airport.
20	Ecuador Earthquake	3,022	A 7.8-magnitude earthquake struck northern Ecuador on 16 April 2016.

The joint probability in PSEM is given by:

$$p(\gamma, \beta, \vec{\alpha}, \vec{s}, \vec{y}, |a, b, c, d, \sigma^2) = p(\gamma|a, b)\, p(\beta|c, d)\, p(\vec{\alpha}|\sigma^2) \prod_t p(s_t|\gamma) \prod_m p(y_{t,m}|s_t, \alpha_t, \beta), \quad (2)$$

where $\vec{\alpha}, \vec{s}, \vec{y}$ are sequences of variables, i.e., $\vec{\alpha} = \{\alpha_0, \cdots, \alpha_{T-1}\}$, $\vec{s} = \{s_0, \cdots, s_{T-1}\}$ and $\vec{y} = \{y_{0,m}, \cdots, y_{T-1,m}\}$.

The optimization algorithm of PSEM follows the framework of variational inference [5]. It presumes that the probability of all hidden variables Z, given the observed sequence $\vec{y}$ and hyper-parameters a, b, c, d, factorizes with respect to variables as follows:

$$q(Z|\vec{y}, a, b, c, d, \sigma^2) = q(\gamma|\hat{a}, \hat{b})\, q(\beta|\hat{c}, \hat{d})\, q(\vec{\alpha}|\hat{\vec{\alpha}}) \prod_t q(s_t|\hat{e}_t), \quad (3)$$

where the symbols with hats are parameters for the factorized distributions and $\hat{e}_t \in \mathcal{R}^2$ is a vector. PSEM iterates over all hidden variables for inference. We use the variational Kalman filter [5] to infer sentiment distribution α.[1]

4 EXPERIMENT

4.1 Data

The statistics of the data we used is shown in Tab. 1. We crawled data sets 1-14 which contain Chinese posts between 2014 and 2018, and they are related to 14 incidents from Weibo. Data sets 15-20 contain English tweets related to 6 incidents from Twitter, which were collected from Twitter event data set [6]. These incidents have gained great attention on the Microblogging platform. In pre-processing, repeated posts, emoji expressions, http links and mentions (@somebody) were removed. For Chinese word segmentation, we utilized the jieba NLP tool[2]. The lexicon we used to extract sentiment words and sentiment values is the HOWNET lexicon[3].

[1] We provide the details of the inference in the supplementary material.
[2] https://github.com/fxsjy/jieba
[3] http://www.keenage.com

Table 2: Average accuracy of different sentiment extraction methods. + indicates improvement with significance level $p < 0.001$.

Methods	Comments length					
	0-20 words		20-40 words		40+ words	
	Positive	Negative	Positive	Negative	Positive	Negative
SentiStrength	0.3774	0.5808	0.2254	0.3906	0.3938	0.3622
SentiStrength-SE	0.6014	0.6951	0.5040	0.5843	0.5752	0.6467
SentiCR	0.7953	0.7855	0.7911	0.7005	0.7404	0.7861
MCNN	0.8199	0.8068	0.8019	0.8060	0.8003	0.8082
RCNN	0.8284	0.8243	0.8291	0.8211	0.8393	0.8353
PESE-I	0.8038	0.8170	0.8011	0.8032	0.8034	0.8166
PESE-C	0.8242	0.8212	0.8269	0.8229	0.8249	0.8291
PESE-T	0.8302	0.8342	0.8398	0.8479	0.8470	0.8486
PESE-G	**0.8477**$^+$	**0.8588**$^+$	**0.8539**$^+$	**0.8771**$^+$	**0.8862**$^+$	**0.9289**$^+$

4.2 Evaluation of Entity-level Sentiment Extraction

To investigate whether incorporating proximity information enhances entity-level sentiment extraction, we generated the ground truth of entity-level sentiment polarity for each post. We first randomly sampled posts for each incident. Then, seven human volunteers were asked to judge the sentiment polarity of each post on each relevant entity. In order to make the ground truth as accurate as possible, the post was added to the ground truth only if at least five volunteers agreed with each other. Sampling and polarity judgment for one incident terminated if we obtained 100 posts. As a result, we created a ground truth data set with sentiment polarity for 2,000 posts (100 posts per incident).

We compared our method PESE with five state-of-the-art methods including lexicon based approaches (SentiStrength [7], SentiStrength-SE [8] and SentiCR [9]) and deep learning based models (MCNN [10] and RCNN [11]). The results of PESE are indicated as PESE-G, PESE-T, PESE-C and PESE-I. G, T, C and I represent Gaussian, Triangle, Cosine and Circle kernel functions, respectively. After tuning using ten-fold cross-validation, parameter σ was set to be 21 for all PESE variants. We used accuracy as the evaluation metric, which is the ratio of number of posts that are correctly judged versus total number of posts.

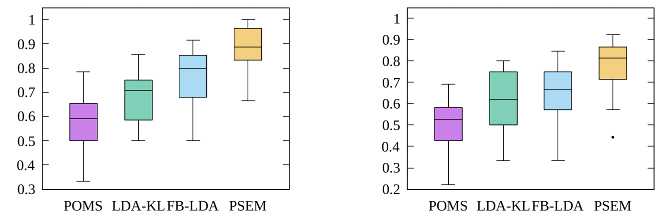
The comparison of results is shown in Tab. 2. As we can see, all PESE variants perform better than competitors. PESE-G achieves the highest average accuracy over all incidents. It noticeably outperforms the second best method (PESE-T) with significance level $p < 0.001$. Moreover, we can observe that positive polarities are usually more challenging to identify, with lower accuracies by most methods. To gain some insights about the effect of text length, we further split the data into three divisions depending on the length of the post. We observe that, as the post gets longer, the accuracy of our proposed method increases.

4.3 Evaluation of Public Sentiment Evolution

For the task of modeling public sentiment evolution, we first investigated the performance of PSEM for detecting shift points of social incident. The ground truth of shift points for each incident was manually generated. Five volunteers were asked to read all posts at each time point and judged whether the time point contains a sentiment shift. The final golden standard for the shift point was selected by taking a majority vote.

Table 3: Average precision and recall of different shift detective methods. + indicates improvement with significance level $p < 0.001$.

	POMS	LDA-KL	FB-LDA	PSEM
Precision	0.5950	0.7000	0.7750	**0.8950**$^+$
Recall	0.5265	0.6195	0.6858	**0.7920**$^+$



(a) Precision **(b) Recall**

Figure 2: Comparison of performance on shift detection

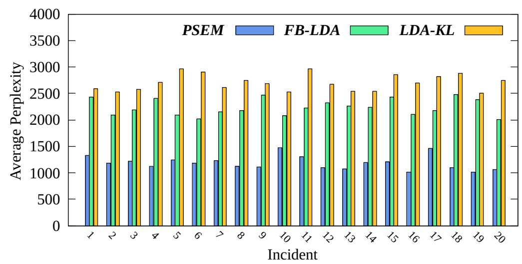


Figure 3: Average per-word predictive perplexity

We compared our method with three state-of-the-art sentiment tracking methods for detecting shift points: POMS [12], FB-LDA [13] and LDA-KL [2]. As this is a binary classification task, we use the standard evaluation metrics: precision and recall. We can observe from Fig. 2 and Tab. 3 that our model PSEM achieves the best results in detecting shift points and significantly exceeds the second best method (FB-LDA) with significance level $p < 0.001$.

Next, we analyze the predictiveness of PSEM. For probabilistic models, researchers usually choose to measure perplexity for predictiveness, which is defined as follows:

$$perplexity = \exp\left\{-\frac{\sum_{D_j \in D}\sum_{w_d \in D_j}\log p(w_d)}{\sum_{D_j \in D} N_{D_j}}\right\}. \quad (4)$$

When computing perplexity, we considered each post as a document of sentiment polarities and D is the document set, $p(w_d)$ is the probability of the d-th word computed by the target model, N_{D_j} is the length of the document D_j. Perplexity is a measurement of how well a probability model predicts a sample. A low perplexity indicates that the model is good at predicting the sample. As shown in Fig. 3, our model has a smaller averaged per-word perplexity in all twenty incidents, which illustrates that our model has a better predictiveness compared to the other methods.

Finally, we offered a visualization of the background sentiment evolution for the twenty incidents modeled by PSEM in Fig. 4 to illustrate its interpretability. In Fig. 4, the number in the legend indicates the number of social incident in Tab. 1, e.g., "1: Miss Jiang" denotes the entity "Miss Jiang" in incident 1. As shown in Fig. 4, α which represents the distribution of background opinion in the incidents is smoothly and slowly changing in all twenty incidents.

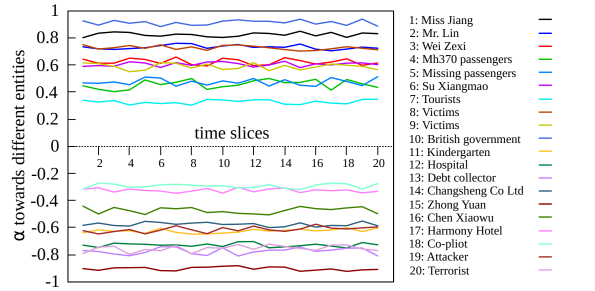


Figure 4: α towards entities are smoothly and slowly changing in all twenty incidents, different α indicating the different back ground opinion to entity.

We also observe that α towards victims like Miss Jiang (Jiang Ge Murder) and MH370 passengers (MH370 Missing) are all greater than 0, indicating that positive sentiment is the background opinion to these entities. Conversely, α towards perpetrators like attackers (Paris Attack) and terrorists (Brussels Airport Explosion) are all less than 0, indicating that negative sentiment is the background opinion to these entities.

5 CONCLUSION

In this paper, we study the problem of tracking public sentiment for social events. We use distance kernels to calculate the influence of sentiment words on entities and propose a novel sentiment evolution model which is based on state space models. We consider the existence of background sentiment distribution and simultaneously model the evolution pattern of background sentiment and sentiment shift. As future work, we plan to endow the model with the ability to explain the causes of sentiment shifts.

REFERENCES

[1] Anne S. Y. Cheung. The battle of microblogging for legal justice in china. *University of Hong Kong Faculty of Law Research Paper*, 2014. URL https://ssrn.com/abstract=2397060.
[2] Anastasia Giachanou, Ida Mele, and Fabio Crestani. Explaining sentiment spikes in twitter. In *CIKM*, pages 2263–2268, 2016.
[3] Mikalai Tsytsarau, Themis Palpanas, and Malú Castellanos. Dynamics of news events and social media reaction. In *KDD*, pages 901–910, 2014.
[4] Yuanhua Lv and ChengXiang Zhai. Positional language models for information retrieval. In *SIGIR*, pages 299–306, 2009.
[5] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.
[6] Arkaitz Zubiaga. A longitudinal assessment of the persistence of twitter datasets. *JASIST*, 69(8):974–984, 2018.
[7] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment in short strength detection informal text. *JASIST*, 61(12):2544–2558, 2010.
[8] Md Rakibul Islam and Minhaz F. Zibran. Leveraging automated sentiment analysis in software engineering. In *MSR*, pages 203–214, 2017.
[9] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. Senticr: a customized sentiment analysis tool for code review interactions. In *ASE*, pages 106–111, 2017.
[10] Hyunjun Ju and Hwanjo Yu. Sentiment classification with convolutional neural network using multiple word representations. In *IMCOM*, pages 9:1–9:7, 2018.
[11] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273, 2015.
[12] Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, pages 450–453, 2011.
[13] Shulong Tan, Yang Li, Huan Sun, Ziyu Guan, Xifeng Yan, Jiajun Bu, Chun Chen, and Xiaofei He. Interpreting the public sentiment variations on twitter. *IEEE Trans. Knowl. Data Eng.*, 26(5):1158–1170, 2014.