Model Sentiment Evolution For Social Incedents

Yunjie Wang[1] and Chen Lin[1]

Department of Computer Science, Xiamen University, China chenlin@xmu.edu.cn

Abstract. Tracking opinion over time is a powerful tool that can be used to detect the possible reasons of a sentiment change. In this paper, we focus on the problem of tracking sentiment changes caused by social events and find the time point where the sentiment is mutated. For a social event, the traditional sentiment outlier detection method focuses on transforming the sentiment polarity of all tweets related to the event into sentiment value of the event itself. For example,we can use the ratio of tweets with positive sentiments or the average of tweet sentiment values. Then detect the sentiment outliers by different ways, such as calculating residuals. We try to model public opinion evolution in a holistic framework and use probability to indicate the likelihood of sentiment mutation at each time point. Experimental results on Weibo data set validate the effectiveness of the proposed model.

Keywords: Sentiment change · Outlier detection

1 Introduction

Nowadays Microblogging has become the major platform for Chinese people to publish information and share opinions about social incidents. Public opinion on Microblogging platforms has greatly influenced the Chinese society. For example, even change investigation and judicial outcome. In the famous "My father is Li Gang" incident, the son of an officer killed an innocent female college student with no regrets. This event caused great anger on the Microblogging, which made the criminals got the punishment he deserved. Public opinion is appealing for individuals, business organizations, researchers, and government officials.

In this paper, to facilitate understanding of public opinions, we focus on the problem of modeling sentiment evolution for social incidents. Given a sequence microblogging comments on one socia incident, our goal is to reveal the evolution patterns of public opinions, identify the significant opinion shifts, and explain cause of the opinion shifts. As shown in Fig. 1, for comments on the "My Dad is Li Gang", we anlayze the public opinion and visualize

Recently there is an increasing interest in tracking microblogging sentiments for entities [1,2] or topics [3,4]. Most of them are based on a two stage framework, i.e.first adopt a sentiment extraction tool such as SentiStrength [9] to compute the sentiment score for an entity or topic, then conduct statistical analysis such as outlier detection to obtain sentiment spikes. However, modeling public opinion for social incidents poses several challenges that haven't been addressed by previous research.

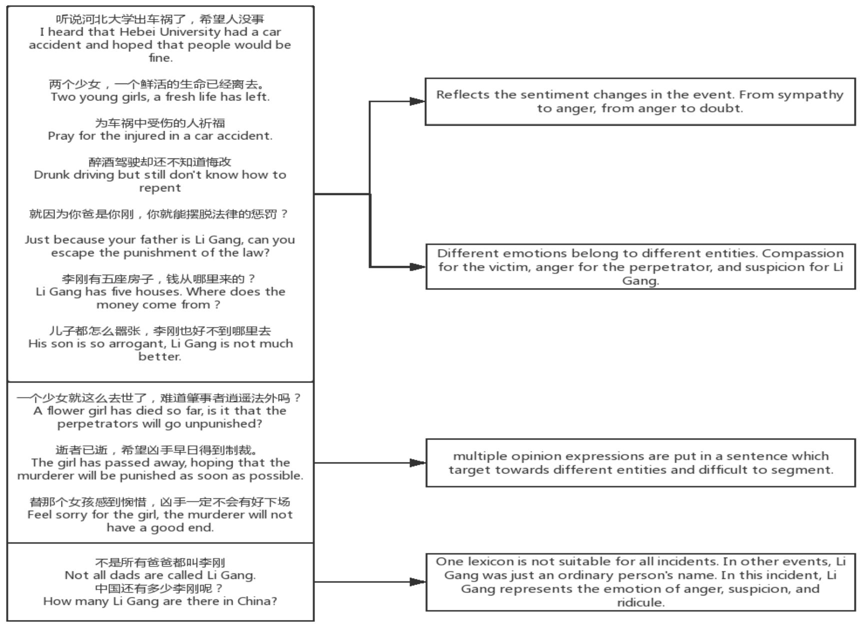

Fig. 1. Comments from "My father is Li Gang" incident and reflected characteristics of social incident

The first challenge is to **identify entity-level sentiment**. As a social incident often involves several entities (i.e. people or organizations), extracting entity level sentiment is important. For example, as shown in Fig 1, in "My father is Li Gang" incident, people negative sentiment to the perpetrator and Li Gang, positive sentiment to the girl who was killed. Existing research utilizes coarse-grained analysis to obtain an averaged sentiment for an event without seperating different entities, which is clearly problematic.This is not a trivial problem because the length limit of microblogs encourage people to use short and informal expressions. As in Fig. 1, multiple opinion expressions are put in a sentence which target towards different entities. To address this challenge, it is helpful to embed proximity information to enhance entity-level sentiment extraction accuracy.

The second challenge is to **build incident specific sentiment lexicon**. In the literature there are two types of sentiment extraction methods: supervised and lexicon based. Due to the dynamic nature of social incidents (i.e. happen suddenly, constantly changing), it is infeasible to label real-time training corpus. Lexicon based sentiment extraction tools are widely adopted [9, 18, 19]. They select a collection of words with negative and positive sentiments to compute the final polarity of sentiments. These methods do not require a training dataset. The accuracy of the classification depends on the quality of the lexicon and the rules for calculating sentiment polarity. The problem is that one lexicon is not suitable for all incidents. For example, in Figure 1, We found that So it is necessary to construct a "global" vocabulary and an incident specific lexicon without supervision.

The third challenge is to **simutanously model the evolution of opinion and opinion shift**. In previous work, researchers mostly depend on statistical analysis to detect opinion shift or sentiment spikes from the background [1,2,22]. We argue here that simply detecting outliers is not sufficient. The fact that events are continuously changing causes changing responses in public opinions. Thus the evolution of the background is largely overlooked. In this article we propose a probabilistic model that simultaneously model the evolution of background opinion and the opinion shifts.

Our contributions are three folds. We investigate the impact of proximity information in obtaining entity-level sentiment extraction. We design an efficient algorithm to extract global lexicon and incident specific lexicon. In the model aspect, we propose to simutanously model the evolution of opinion and opinion shift

This paper is organized as follows. We briefly survey the related work in Sec. 2. In Sec. 3 to Sec. 5, we describe the methodology. We present and analyze the experimental results on a real data set in Sec. 6. We conclude our work and suggest future directions in Sec. 7.

2 Related Work

Sentiment Tracking on Microblogs has received considerable attention from both academy and industry [].Most of existing work adopt a cascade framework, i.e. first sentiment extraction, second shift detection. [].To extract sentiment, the collection of tweets are divided into numerous time slices, and compute the ratio of positive and negatie sentiments in a time slice [], or . To detect sentiment shift, residual between actural and predicted sentiment value is computed in xxx, xxx, xxx [22]Topical residual is incorporated inxx []A few utilizes an integrate framework []

A fundamental block in sentiment tracking systems is **sentiment analysis**. In the literature, there are two types of sentiment analysis algorithms: supervised and lexicon-based [11]. Preprocessing syntatic and semantic features have been proved to be the most critical part in training a good classifier []. Morover, since it is difficult to access large-scale training data, recently a few recent research explores the possibility of distant supervision [].

Due to the lack of training data, most researchers turn to lexicon based methods.The first SentiStrength, which is used as a baseline in xxxx [].However, SentiStrength is not accurate enough due to ...xx improves the accuracy of sentistrength by [], xx increases recall by xx [] Recently, a few researchers xxx. Feng et al. proposed using connotation lexicons to enclose subtle dimensions of a words sentiment. They first define a seed lexicon and then they learn the connotation lexicon based on PageRank and HITS [21].However,

3 Problem Definition

Social events are planned by people, attended by people and that the media illustrating the events are captured by people [10]. So social events involve interactions between different entities. Social events are dynamic and occur in a limited time. Emotional changes contained in social events tend to be regular. These features result in sentiment analysis and sentiment tracking methods previously used on entities and topics that no longer apply to social events. We use a new opinion model to simulate the evolution of public opinion in social events and to find the time points when sentiments shift suddenly during the evolution. In order to achieve the best results of our model, we consider the impact of the entity on the results of sentiment analysis, and use the core lexicon and feature lexicon to construct a unique lexicon of each event to improve the accuracy of sentiment classification.

4 Sentiment Classification

We have improved two aspects to increase the accuracy of sentiment classification: (1)We use the core lexicon and feature lexicon to construct a unique lexicon of each event. (2) We add entities to the process of sentiment classification.

4.1 Core Lexicon And Feature Lexicon

Here, we describe the process of generating the core lexicon and feature lexicon. The core lexicon is a collection of emotional words shared by all events. The words in the core lexicon are not only shared by all events, but also represent the strongest and most explicit sentiment polarity. The feature lexicon consists of sentiment words that only included in one event. We follow the steps below to find the core lexicon and feature lexicon:

1. Extract the words in the corpus and remove the neutral words, low frequency words and stop words to get the candidate lexicon.

2. Use a graph to represent the candidate lexicon. Each word in the candidate lexicon is used as a node. If two words appear in the same sentence, then there is an edge between the nodes corresponding to the two words. Traversing the corpus of several events, getting the graph corresponding to these events.

3. Look for the core nodes and feature nodes in the graph, and their corresponding words form the core lexicon and feature lexicon.

We use Tiantian Zhang's method to find the core node in the graph [26]. This method starts from the source node. Nodes connected to the source node and having a greater degree than the source node constitute a candidate nodes set. Then select the node in the candidate set that is closest to the source node. Repeat this process to get closer to the core node step by step. This method uses the following formula to calculate the intimacy between two nodes A and B :

$$intimate(A,B) = \frac{2 * |Edeg(adj(A) \cap adj(B))|}{|(adj(A) \cap adj(B)| * (|(adj(A) \cap adj(B)| - 1)} \tag{1}$$

adj(A) is the set of nodes adjacent to A, adj(A) ∩ adj(B) is the common nodes of adjacent nodes of A and adjacent nodes of B. Edge(adj(A) ∩ adj(B)) stands for the edges linked by the nodes in adj(A) ∩ adj(B).

We start with each node with the degree equal to one in the graph to find the core node, and the corresponding words form the core lexicon. In order to ensure that the feature words belong to the sentiment words, we regard the three nodes closest to the core node on the path of finding the core node as feature nodes, and the corresponding words form the feature lexicon.

4.2 Distance Function

Here, we describe how to add entities to the process of sentiment classification. We use entities as the granularity of sentiment classification. We believe that in one sentence, the sentiment words in different positions have different effects on the entity. Sentiment words that are close to the entity have a greater influence on the sentiment polarity of the sentence than the sentiment words far from the entity. We use the average of all sentiment words on the entity as the sentiment polarity of the sentence.

We use the distance function to calculate influence of sentiment words on entities. Following some previous work, We use four kernel functions as the form of the distance function in turn: Gaussian, Triangle, Cosine, and Circle:

1. Gaussian kernel

$$k(i,j) = \exp\left[\frac{-(i-j)^2}{2\sigma^2}\right] \tag{2}$$

2. Triangle kernel

$$k(i,j) = \begin{cases} 1 - \frac{|i-j|}{\sigma} & \text{if } |i-j| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

3. Cosine (Hamming) kernel

$$k(i,j) = \begin{cases} \frac{1}{2}\left[1 + cos\left(\frac{|i-j| \cdot \pi}{\sigma}\right)\right] & \text{if } |i-j| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

4. Circle kernel

$$k(i,j) = \begin{cases} \sqrt{1 - \left(\frac{|i-j|^2}{\sigma}\right)} & \text{if } |i-j| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

i is the position of the sentiment word in the sentence, and j is the position of the entity in the sentence. All four of these kernel functions have only one parameter. We will determine the best form of the distance function and best value of the parameter in the experiment.

4.3 Calculate Sentiment polarity

Based on the above work, here we give the formula for calculating the sentiment value of a sentence. We first calculate the influence of one of the sentiment words on the entity.

$$s_i = (-1)^n \cdot d_i \cdot v_i \cdot k(p_i, j) \tag{6}$$

i is the number of this sentiment word. n is the number of gainsay words between this emotional word and the sentiment word before it. d_i is the sum of the weights of the degree words between this sentiment word and the sentiment word before it. v_i is the emotional value of this emotional word. k is the distance function. p_i is the location of this emotional word. j is the location of the entity.

We calculate the influence of all sentiment words on the entity according to this formula, and take the average value as the sentiment value of the sentence. If the sentiment value is greater than 0, we think that the sentence has a positive emotion. If the sentiment value is less than 0, we think that the sentence has a negative emotion.

5 Public Opinion Model

Dramatic real-world events are known to have the power to impact on public opinions and to cause shift on public attitudes. For example, assassinations of Martin Luther King created a cultural shift in attitudes on race issues. Thus we are motivated to analyze news and user generated contents on social networking platforms, model the evolution and shifts of public opinions, exact common patterns and explain anomalies.

We first explore our options on modeling the evolution of public opinions. Before we go into the details, we have to list a few assumptions here. (1) We only consider bi-polarized opinions, i.e. each unit piece of social comment is pre-processed by some opinion classifiers to be either positive or negative. The unit to be processed can be a phrase about an entity, a sentence, or a minimal length semantic unit. (2) We consider there is a background opinion distribution, i.e. how users normally react to a certain entity or a particular event. The background is smoothly and slowly changing, e.g. the public opinion for civil rights is constantly changing. (3) However, sometimes a new piece of evidence might trigger a sudden shift on public opinions, e.g. the assassination of Martin Luther King boosts public supports to civil rights.

First we generate the prior distribution for the background opinion distributions.

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

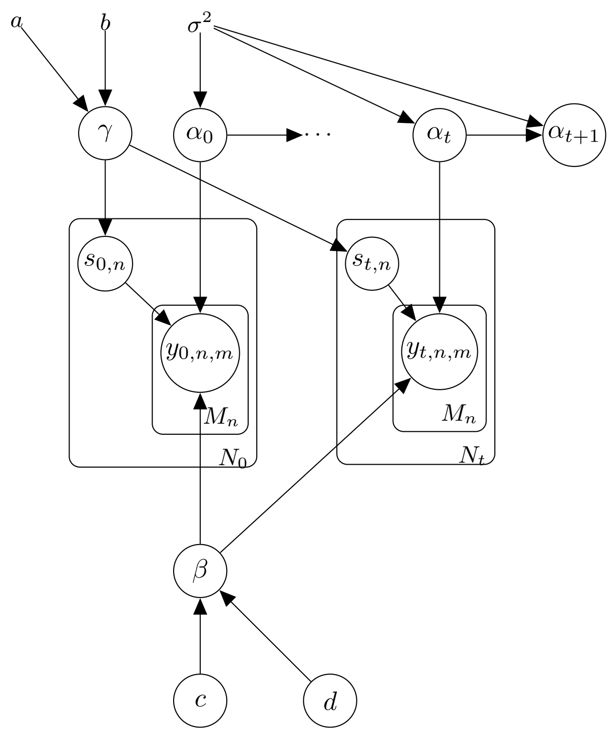

Fig. 2. Plate notation of the proposed opinion evolution model

Note that all αs are vectors of real numbers, $\alpha \in \mathcal{R}^2$. Next to generate the opinion distribution, we adopt $\pi(\cdot)$ function to convert α to a value within $(0,1)$, $\pi(\alpha_t) = \frac{\exp \alpha_{t,0}}{\exp \alpha_{t,0} + \exp \alpha_{t,1}}$.

For time t, with N_t pieces of evidence, i.e. N_t piece of news report published at time t, each evidence n contains M_n observations, e.g. user comments, we mimic the following generation process.

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

The joint probability is given by

$$
\begin{aligned}
& p(\gamma, \beta, \alpha_0, \cdots, \alpha_T, \boldsymbol{s}, \boldsymbol{y}, |a, b, c, d, \sigma^2) \\
= p(\gamma|a,b)p(\beta|c,d)p(\alpha_{0:T}|\sigma^2) & \prod_t \prod_n p(s_{t,n}|\gamma) \prod_m p(y_{t,n,m}|s_{t,n}, \alpha_t, \beta)
\end{aligned}
$$

However, as the above complete likelihood involves natural parameters $\pi(\alpha)$, we apply the lower-bound $\forall t, \ln(\exp \alpha_{t,0} + \exp \alpha_{t,1}) \le \ln \hat{\xi}_t + \frac{\exp \alpha_{t,0} + \exp \alpha_{t,1} - \hat{\xi}_t}{\hat{\xi}_t}$ to $p(y_{t,n,m}|s_{t,n}, \alpha_t, \beta) = [\pi(\alpha_t)^{y_{t,n,m}}(1-\pi(\alpha_t))^{1-y_{t,n,m}}]^{s_{t,n}}[\beta^{y_{t,n,m}}(1-\beta)^{1-y_{t,n,m}}]^{1-s_{t,n}}$ whenever neccessary. In the nutshell, the optimization algorithm is variational inference. Thus we make the following assumptions.

$$
q(Z|\boldsymbol{y}, a, b, c, d, \sigma^2) = q(\gamma|\hat{a}, \hat{b})q(\beta|\hat{c}, \hat{d})q(\alpha_{0:T}|\hat{\alpha_{0:T}}) \prod_{t,n} q(s_{t,n}|\hat{e_{t,n}}),
$$

where Z includes all hidden variables, $\hat{e_{t,n}} \in \mathcal{R}^2$ is a vector.

Then we implement the iterations over all hidden variables.

6 Experiment

6.1 Experimental Setup

The data set used in our experiment is crawled through weibo API, collected between 2016 and 2018.through the microblogging API using keyword matching. The corpus includes 6 incidents, The tweets for each event are divided into news and corresponding comments. Details of the data set, including the description of each event, the number of news and comments are shown in Tab. 1.

In pre-processing, repeated tweets, emoji expressions, http links and mentions (@somebody) are removed from the data set. Segmentation

Table 1. Statistics of the data set

Abbrevation	Tweets	Time period (start end)	Event description
JGmurder	368037	2016/11/02 2018/01/01	Chinese female student Jiang Ge was killed in Japan
CFfall	35081	2017/08/31 2017/10/16	A maternal woman jumped died in the hospital
RYBabused	35927	2017/11/23 2017/12/27	Many children were abused in a kindergarten
HZbabysitter	167225	2017/06/22 2017/11/01	A nanny in Hangzhou burned his employers
SDhumiliation	17607	2017/03/25 2017/08/31	A mother in Shandong was humiliated because she owed money.
WZXhospital	59501	2016/04/21 2016/09/11	Wei Zexi died of fake medical information

6.2 Sentiment Lexicon Extraction

We first verify that the performance of lexicon construction.

The ground truth is the polarity (positive v.s. negative) of sentiment words. in manually generated.

We choose precision, recall as evaluation metrics. The precision help us judge that how many words in the lexicon belong to sentiment words. The recall tell us the proportion of sentiment words in the lexicon to all sentiment words in the ground truth. We compare our method with PMI on sentiment lexicon extraction. PMI is a commonly used method that extends the seed lexicon by calculating pointwise mutual information of candidate words and sentiment seeds. The results are shown in Tab. 2

Table 2. Comparative of Core lexicon and PMI

Events	Tweet	PMI			Core Lexicon		
		Precision	Recall	Size of Lexicon	Precision	Recall	Size of Lexicon
JGmurder	368037	0.7179	0.7188	176	0.7936	0.7604	165
HZbabysitter	167225	0.7097	0.7750	101	0.7766	0.7833	94
WZXhospital	59501	0.6508	0.8289	60	0.8548	0.8158	62
RYBabused	35927	0.7083	0.7619	77	0.8077	0.8254	52
CFfall	35081	0.4468	0.7015	75	0.8103	0.8557	58
SDhumiliation	17607	0.4074	0.7714	56	0.8484	0.9429	33

Observing from Tab. 2, we can see that our proposed approach is more effective for sentiment lexicon extraction than PMI. Among all six events, the core lexicon approach has higher precision than PMI, and among the five events, the core lexicon approach has a higher recall than PMI. More importantly, as the corpus of events becomes smaller, the precision of PMI shows a downward trend. PMI's precision on the two events with the smallest corpus is only 44 percent and 40 percent. At the same time, the PMI recall remained stable. This shows that as the event corpus becomes smaller, PMI sacrifices precision in order to extract emotional words as much as possible, resulting in a much larger lexicon than the lexicon obtained by the core lexicon method. Compared with PMI, as the corpus of events becomes smaller, the precision and recall of the lexicon obtained by the core lexicon method is increasing. Therefore, the lexicon extracted by the core lexicon method has better quality, especially when the corpus of events is small.

6.3 Parameter Selection For Distance Function

Here we will get the form of the distance function and determine the parameters of the it through experiments. The experimental data set consisted of 2,000 tweets randomly selected from the corpus of six events. Manually mark the polarity of these tweets and determine the location of the entity by keyword. We use different distance functions to calculate the sentiment classification accuracy for the data set and systematically test a set of fixed σ values from 1 to 30 in increments of 1. The results are shown in Fig. 3

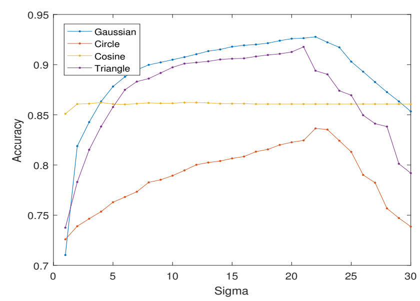

Fig. 3. For different kernel functions, the relationship between sentiment analysis accuracy and σ

First, it is clear that adding entity to the process of sentiment recognition can improve the accuracy of sentiment classification. Among the four kernel functions, the Cosine kernel function is less affected by sigma. The accuracy of sentiment classification corresponding to the other three kernel functions increases first and then decreases, and all three kernel functions get the optimal value when sigma is equal to 21. The Gaussian kernel function performs best. So we take the Gaussian kernel function as a form of the distance function and set the value of sigma to 21. Under such conditions, the sentiment classification accuracy of the data set reached 92.77 percent.

6.4 Evaluation of Sentiment Extraction

Here, we show the impact of using the core lexicon and distance functions on sentiment classification. Before the experiment, we suspected that the distance function is more advantageous for long texts, because if a sentence contains few sentiment words, the distance of the sentiment words to the entity no longer has a strong influence on the classification results. In order to validate our conjecture, we randomly extracted three different lengths of comments from the event corpus to form three data sets. Manually annotate the emotional polarity of tweets in the dataset.

We compared our method to SentiStrength, SentiStrength-SE [27] and SentCR. SentiStrength is a classic algorithm for sentiment classification. SentiStrength-SE has improved SentiStrength, this method designs different lexicon for events in different fields. SentCR is a supervised learning method designed for code review comments. We compared the sentiment classification accuracy of these four methods on positive and negative data set, and compared the impact of the length of the tweets on the classification results. The results are shown in Tab. 3

According to Table 3, Our approach has the best performance. On all dataset, our method's sentiment classification accuracy exceeds 80 percent, especially for negative dataset with tweets between 40 and 60 in length, the accuracy of our method is over 90 percent. There are many details in the table. We can see that except for SentCR, the accuracy of the other three methods on the positive dataset is lower than the accuracy on the negative dataset. This is because in our event, most of the comments contain a large number of negative words, but some of them are expressed in support of another person by expressing dislike of someone. Although there are many negative words in the tweet, the tweet still expresses positive emotions. Therefore, in our corpus, judging positive emotions is difficult to judge negative emotions. Despite this, our approach has achieved high accuracy on positive data sets. The accuracy of SentiCR remains stable because supervised learning methods are not affected by tweet semantics and structure. As the tweet grows longer, the accuracy of the our method using the distance function is significantly improved compared to the our method without using the distance function. So the distance function is more effective for long comments.

Table 3. Comparison of sentiment classification results

Methods	Sentence length					
	0-20		20-40		40-60	
	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
xxx	0.7279	0.7558	0.7517	0.7792	0.7477	0.7641
xxx	0.8477	0.8588	0.8539	0.8771	0.8862	0.9289

6.5 Evaluation of Public Opinion Model

Here, we show the advantages of our public opinion model in simulating the evolution of public opinion. We set 19 time points for each event, with the same interval between each two time points. We use the core lexicon and feature lexicon to build a lexicon for each event, and calculate the sentiment polarity of each tweet in the corpus corresponding to each time point by the distance function.

We show the changes of the parameter e in the Tab. 4 The parameter e represents the probability of a sudden change in public opinion at each time point. So we use the parameter e to find the shift points in the sentiment change. For each event, we use the time point where the value of e is greater than 0.8 as the shift point. The ground truth of sentiment changes is obtained manually, we analyze the events at all time points and select the time points we believe can cause a sudden change in public opinion. We compare our model to the ground truth. The results are shown in Tab. 5. Y represents this time point is considered to be a sentiment shift point. N is the opposite. As shown in Table 5, The ground truth selected a total of 23 time points as the sentiment shift point. In the case that our model only selected 20 time points, 18 time points of them were considered correct by ground truth. This proves the validity of our model.

Table 4. Comparative performance of shift detection

7 Conclusion

In this paper, we study the problem of tracking the evolution of public opinion in social events. We analyze the differences between social events and entities in sentiment analysis, and propose a new opinion evolution model to track the changes in public opinion in social events. We consider the existence of background opinion distribution in the model, and use probability to indicate the likelihood of sudden changes in sentiments at each time point. To improve the performance of our model, we have improved the method of sentiment analysis based on the lexicon. We obtain the core lexicon and feature lexicon by finding the core nodes and feature nodes in the graph, which improves the quality of the lexicon. We add entities in the process of sentiment analysis, use the distance function to calculate the influence of emotional words on entities, and experimentally prove that the distance function is more effective for long comments. In the future, we plan to track changes in public opinion about different entities in social events. Changes in opinions of different entities can reflect the relationship between entities. We also consider adding topics to our model to improve the accuracy of the tracking opinions evolution.

References

1. Anastasia Giachanou, Ida Mele and Fabio Crestani. Explaining Sentiment Spikes in Twitter. In CIKM 16, PP. 2263-2268, 2016.

2. Anastasia Giachanou, Ida Mele and Fabio Crestani. A Collection for Detecting Triggers of Sentiment Spikes. In SIGIR 17, PP. 1249-1252, 2017.
3. Mikalai Tsytsarau, Themis Palpanas and Malu Castellanos. Dynamics of news events and social media reaction. In KDD 14, PP. 901-910, 2014.
4. Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in Twitter Events. In Journal, pp. 406-418, 2011.
5. M. Hu and B. Liu. SMining and summarizing customer reviews. In SIGKDD 04, 2014.
6. P. D. Turney. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In 40th Annual Meeting on Association for Computational Linguistics. 2002.
7. John ONeil. Entity Sentiment Extraction Using Text Ranking. In SIGIR 12, pp. 1024-1024, 2012.
8. Diana MaynardAdam Funk. Automatic Detection of Political Opinions in Tweets. In ESWC 11, pp. 88-89, 2011.
9. Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. Journal of the Association for Information Science and Technology 61, 12 (2010), 25442558.
10. S. Papadopoulos, E. Schinas, V. Mezaris, R. Troncy, and I. Kompatsiaris. s. Social event detection at mediaeval 2012: challenges, dataset and evaluation. In Proceedings of MediaEval 2012, 2012.
11. Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, Shahram Rahimi. SentiCR: A Customized Sentiment Analysis Tool for Code Review Interactions. In ASE 17, 2017.
12. Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification Using Distant Supervision. Technical Report. Standford.
13. Luciano Barbosa and Junlan Feng.2010. Robust sentiment detection on twitter from biased and noisy data. In COLING 10, pp. 36-44, 2010.
14. Dmitry Davidov, Oren Tsur, and Ari Rappoport.2010. Enhanced sentiment learning using twitter hashtags and smileys. In COLING 10, pp. 241-249, 2010.
15. Akshat Bakliwal, Piyush Arora, Senthil Madhappan, Nikhil Kapre, Mukesh Singh, and Vasudeva Varma.2012. Mining sentiments from tweets. In WASSA 12, pp. 11-18, 2012.
16. Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. J. Artif. Intell. Res. 50 (2014), 723762.
17. Nathan Aston, Jacob Liddle, and Wei Hu. 2014. Twitter sentiment in data streams with perceptron. J.Comput. Commun. 2 (2014), 1116.
18. Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. J. Am. Soc. Inform. Sci. Technol. 63, 1 (2012), 163173.
19. Reynier Ortega, Adrian Fonseca, and Andres Montoyo. 2013. SSA-UO: Unsupervised twitter sentiment analysis. In Proceedings of the 7th International Workshop on Semantic Evaluation.
20. Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. 2013. SSA-UO: Unsupervised sentiment analysis with emotional signals. In WWW 13, pp. 607-618, 2013.
21. Song Feng, Ritwik Bose, and Yejin Choi. 2011. Learning general connotation of words using graph-based algorithms. In EMNLP 11, pp. 1092-1103, 2011.
22. Anastasia Giachanou, Fabio Crestani. 2011. Tracking Sentiment by Time Series Analysis. In SIGIR 16, pp. 1037-1040, 2016
23. Shulong Tan, Yang Li, Huan Sun, Ziyu Guan, Xifeng Yan, Jiajun Bu, Chun Chen, Xiaofei He. Interpreting the Public Sentiment Variations on Twitter. In IEEE14, pp. 1158-1170, 2014.

24. X. An, R. A. Ganguly, Y. Fang, B. S. Scyphers, M. A. Hunter, and G. J. Tracking Climate Change Opinions from Twitter Data. In KDD14: Workshop on Data Science for Social Good, 2014.
25. J. Bollen, A. Pepe, and H. Mao. Modeling Public Mood and Emotion : Twitter Sentiment and Socio-Economic Phenomena. In ICWSM11, pages 450-453, 2011.
26. Tiantian Zhang, Bin Wu. A Method for Local Community Detection by Finding Core Nodes. In Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.
27. Md Rakibul Islam and Minhaz F Zibran. 2017. Leveraging automated sentiment analysis in software engineering. In Proceedings of the 14th International Conference on Mining Software Repositories. IEEE Press, 203214.