Model Sentiment Evolution For Social Incidents

Submission

No Institute Given

Abstract. Modeling sentiment evolution for social incidents in microblogs is of vital importance for both researchers and government officials. Existing work on sentiment tracking is 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 towards different entities. To address this problem, in this paper, we investigate the impact of proximity information to obtain more precise entity-level sentiment extraction. Furthermore,detecting sentiment shift is not a trivial problem because the evolution of the background sentiment can not be ignored. We propose to simultaneously model the evolution of sentiment and sentiment shift by a state space model on the time series of sentiment polarities. Experiments on a real data set demonstrates that the proposed methods outperform state-of-the-art methods.

Keywords: Sentiment Tracking · Dynamic Sentiment Model · Opinion Analysis · Microblog Mining

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 some incidents even change the investigation and judicial outcome [1]. For example, in 2010, Twenty-nine-year-old Li Changkui was originally condemned to immediate execution by a local court in Zhaotong because he killed a three-year-old boy and his teenager sister after raping her. The higher people's court of Yunnan later overruled the sentence and gave Li a two-year reprieve because he confessed his crime and gave compensation to the victims' family. The overruling caused great anger on microblogs with many arguing Li deserved to die for his brutality. Finally, the higher people's court of Yunnan overruled its previous decision and sentenced Li to death. The power of public opinion in Microblogging space makes it appealing to analyze sentiment evolution for social incidents in microblogs for individuals, enterprises, NGO organizations, researchers, government officials and so on.

In this paper, to facilitate understanding of public opinions, we focus on the problem of modeling sentiment evolution for social incidents. 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 the significant sentiment shifts. As shown in Fig. 1, analysis of online comments on the murder case of Jiang Ge[1] leads to visualization of the evolution pattern of public sentiment towards the victim's mother (Jiang) and the victim's roommate (Liu). A sentiment shift is also detected in the third time point.

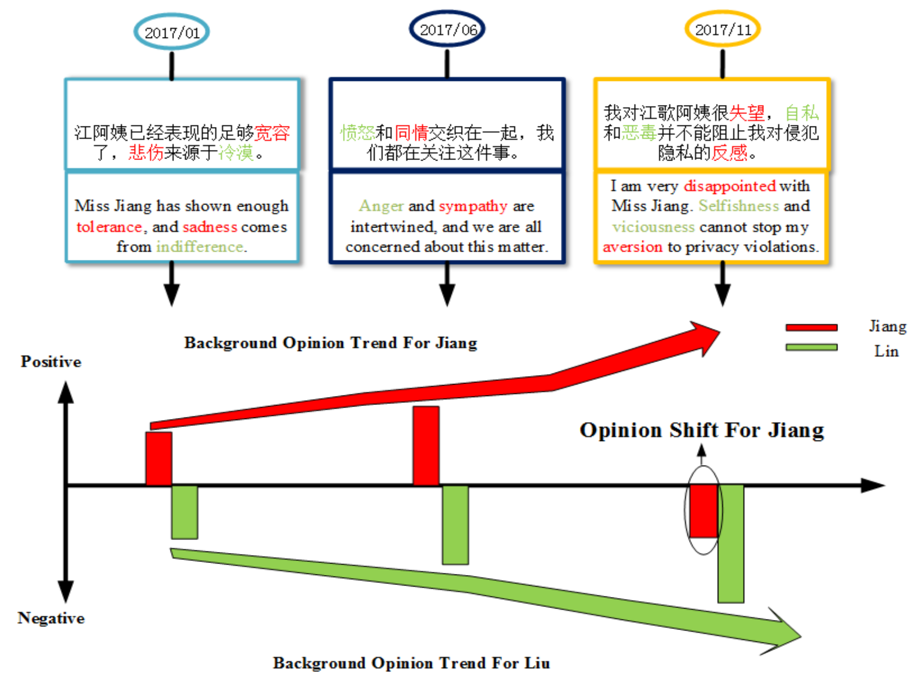


Fig. 1. Comments from "Jiang Ge" incident and reflected characteristics of social incident.

Recently there is an increasing interest in tracking microblogging sentiments for entities [2,3] or topics [4,5]. Most of them are based on a two stage framework, i.e. first adopt a sentiment extraction tool such as SentiStrength [6] 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 two challenges that haven't been addressed by previous research.

The first challenge is to **identify entity-level sentiment**. As a social incident often involves several entities (i.e. people or organizations), it is clearly

[1] The murder case of Jiang Ge, a Chinese student killed in Japan in 2016 has attracted wide attention online. Jiang was stabbed to death in her apartment by her roommate's boyfriend. After the tragedy, Jiang's mother (Jiang) blamed her daughter's roommate (Liu) for her daughter's death by claiming Liu had locked Jiang out when she was attacked.

problematic to utilize coarse-grained analysis which obtains an averaged sentiment for an event without separating different entities. However, extracting entity level sentiment is not a trivial problem because the length limit of microblogs encourages people to use short and informal expressions. Multiple opinion expressions are put in a sentence which targets towards different entities. For example, as shown in Fig. 1, sentiment words for Jiang (in red) and for Liu (in green) are mixed together without a clear partition and a correct grammar structure. To address this challenge, it is helpful to **embed proximity information** to enhance entity-level sentiment extraction accuracy.

The second challenge is to **detect sentiment shift**. In previous work, researchers mostly depend on statistical analysis such as outlier detection to detect sentiment spikes [2, 3, 9]. Such a method is not sufficient, because the evolution of the background is largely overlooked. The fact that events are continuously changing causes changing responses in public opinions. Hence sentiment shift should be distinguished with the evolution patterns of background sentiment. For example, in Figure 1, the background sentiment towards Jiang is an increasing trend of positive sentiment. Revealing this evolution pattern marks the significance of sentiment shift at the third time point. In this article we propose a probabilistic model that **simultaneously model 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 Chinese 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 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. Furthermore, we investigate the impact of proximity information in obtaining entity-level sentiment extraction.

This paper is organized as follows. We briefly survey the related work in Sec. 2. In Sec. 3 to Sec. 4, we describe the methodology. We present and analyze the experimental results on a real data set in Sec. 5. We conclude our work and suggest future directions in Sec. 6.

2 Related Work

Sentiment Tracking on Microblogs has received considerable attention from both academy and industry [2, 3, 9–13]. Most of existing work adopt a cascade framework, i.e. in the first step sentiment of each tweet is extracted, in the second step sentiment shift is detected [2, 3, 9–12]. To extract sentiment, the collection of tweets are divided into numerous time slices, and the ratio of positive and negative sentiments is computed in a time slice [3, 9–11]. To detect sentiment shift, residual between actuarial and predicted sentiment value is the most commonly adopted measurement [2, 9]. Furthermore, topic information is incorporated in recent studies. Sentiment change is represented by topic changes in [12], an integrate framework based on empirical heuristics is utilized in [13] to identify the emotional spikes and locate causes of spikes.

A fundamental block in sentiment tracking systems is **sentiment analysis**. In the literature, there are two types of sentiment analysis algorithms: supervised learning and lexicon-based methods [14]. **Supervised learning method** creates a training model based on training data to classify the sentiment polarity of sentences. Obtaining training data and selecting features are the two most important parts of this methods. Emoji is often used to label sentiments of tweets [15, 18]. Hashtag is another major source to label training data [17]. But the accuracy of label by emoji is low. To overcome this issue, an ensemble of sentiment detection tools is employed to obtain the training data [16]. The goal of supervision based methods is to classify the polarity of sentiments. To obtain a high precision, lexical features such as unigrams and POS [15, 19], syntax features such as retweets, URLs, emoticons, and meta-features such as POS tags, words' polarity, [16] are obtained. Experiments have shown that classifiers benefit most from features which involve text polarity [20].

Due to the lack of training data, most researchers turn to **lexicon based methods**. SentiStrength is the first open domain large-scale lexicon, which is used as a baseline in most sentiment detection algorithms like SentiStrength2 [7], SentiStrength-SE [22], and VADER [21]. SentiStrength2 [7] improves the accuracy of SentiStrength by adding idioms. VADER [21] improves the accuracy of SentiStrength by grouping sentiment words on Twitter and manually filtering them. SentiStrength-SE improves the recall of SentiStrength by designing different lexicons for different domains [22]. The above methods are based on static lexicons. Recently,we've seen an emerging attempt to construct dynamic lexicon. For example, to enclose subtle dimensions of a word's sentiment, a seed lexicon is defined in [23] and connotation lexicon is retrieved based on PageRank and HITS. However, directly applying these lexicons does not guarantee the accuracy of entity level sentiment extraction, because opinion expressions towards different entities are usually mixed together in a microblogging post.

3 Proximity-based Entity-level Sentiment Extraction

3.1 Problem Definition

In this section, we describe Proximity-based Entity-level Sentiment Extraction (PESE). 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, $o = \{(w_i, l_i, v_i)\} \bigcup \{(e_j, l_j)\}$, where i is the index for sentiment words, w_i is the word which is extracted from a sentiment lexicon $w_i \in D$, l_i is the location and v_i is the sentiment value of w_i, which is also extracted from a sentiment lexicon, $v_i \in \mathcal{R}$, j is the index for entity occurences, $e_j \in E$ is the name of the entity, l_j is the location of the entity in the post. Our aim is to output the sentiment polarity p for the entity e_j in the post $p_e(o) \in \{0, 1\}$.

3.2 Distance Function

Our basic assumption 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 the sentiment word is to describe the entity. Inspired by [25], given two locations l_i, l_j, we use four distance kernel functions to compute influence of sentiment words on entities, namely Gaussian, Triangle, Cosine, and Circle:

1. Gaussian kernel

$$k(l_i, l_j) = \exp\left[\frac{-(l_i - l_j)^2}{2\sigma^2}\right], \tag{1}$$

2. Triangle kernel

$$k(l_i, l_j) = \begin{cases} 1 - \frac{|l_i - l_j|}{\sigma} & \text{if } |l_i - l_j| \leq \sigma \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

3. Cosine (Hamming) kernel

$$k(l_i, l_j) = \begin{cases} \frac{1}{2}\left[1 + cos\left(\frac{|l_i - l_j| \cdot \pi}{\sigma}\right)\right] & \text{if } |l_i - l_j| \leq \sigma \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

4. Circle kernel

$$k(l_i, l_j) = \begin{cases} \sqrt{1 - \left(\frac{|l_i - l_j|^2}{\sigma}\right)} & \text{if } |l_i - l_j| \leq \sigma \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

Note that all four of these kernel functions are governed by one parameter σ, which is tuned in the experiment. To obtain the proximity influence between an entity e and a sentiment word w, we compute the average distance over its multiple occurrences, that is $d(e, w) = \sum_{l_i, l_j} k(l_i, l_j)/(n_i \times n_j)$, where l_i is the location of each occurrence of sentiment word w, l_j is the location of each occurrence of entity e, n_i is the number of occurrences of w and n_j is the number of occurrences of e

3.3 Entity Level Sentiment Polarity Classification

To classify the polarity $p_e(o)$ of sentiment towards an entity e in a post o, we first obtain an entity-level sentiment value by calculating the average of the influence on the entity from different sentiment words. n_i is the number of anti-words and d_i is the sum of value of degree words between ith sentiment word and $(i-1)$th sentiment word. N is the number of sentiment words.

$$s = \frac{\sum_{i=1}^{N} (-1)^{n_i} \cdot d_i \cdot v_i \cdot k(l_i, l_j)}{N} \tag{5}$$

if the sentiment value $s > 0$, the sentiment polarity of this sentence is positive $p_e(o) = 1$. if the sentiment value $s < 0$, the sentiment polarity of this sentence is negative $p_e(o) = 0$.

4 Public Sentiment Evolution Model

4.1 Problem Definition

In this section, we describe how to model public sentiment evolution. For a social incident which involves several entities $E = \{e\}$, we first divide the collection of tweets to T time slices. Suppose in each time slice, there are N_t posts, where each post is pre-processed by the PESE to observe an entity-level sentiment polarity on each post $p \in \{0, 1\}$. We build a public sentiment evolution model for each entity. Our assumptions are: (1) there is a background opinion distribution, i.e. how users normally react to the entity. (2) The background is smoothly and slowly changing. (3) However sometimes a sudden shift on public opinions appears, i.e. triggered by sometimes a new piece of evidence.

Therefore, we present the following generation process, as illustrated in Figure 2.

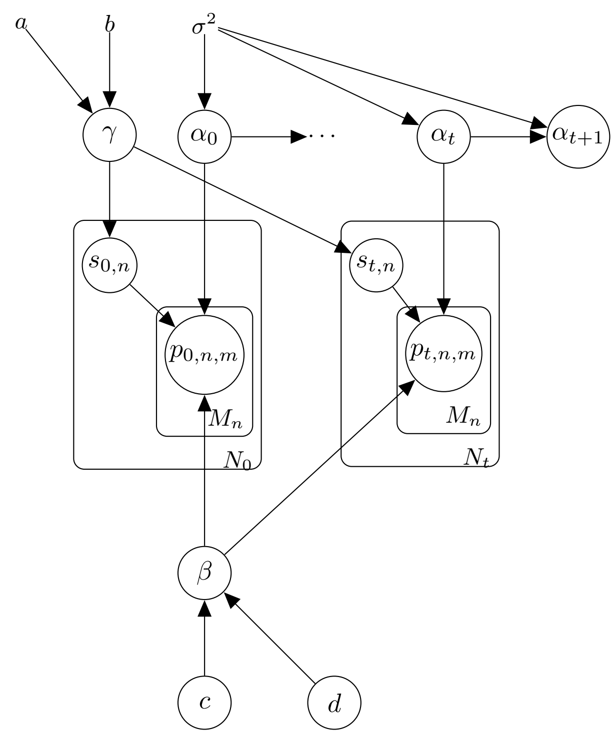


Fig. 2. Plate notation of the proposed opinion evolution model

- 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 $\gamma \sim Beta(a, b)$

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

4.2 Inference

The joint probability is given by

$$\begin{aligned} &p(\gamma, \beta, \alpha_0, \cdots, \alpha_T, \boldsymbol{s}, \boldsymbol{p}, |a, b, c, d, \sigma^2) \\ &\quad = p(\gamma|a, b)p(\beta|c, d)p(\alpha_{0:T}|\sigma^2) \prod_t \prod_n p(s_{t,n}|\gamma) \prod_m p(p_{t,n,m}|s_{t,n}, \alpha_t, \beta) \end{aligned} \tag{6}$$

In the nutshell, the optimization algorithm follows the framework of variational inference. Thus we make the following assumptions.

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

Then we implement the iterations over all hidden variable, which is described in Algorithm 1.

Algorithm 1: Iteration Process For All Hidden Variable

Input: Initial value of a, b, c, d, α
Output: Stable value of a, b, c, d, α, e
while $\hat{e_{t,n_1}}$ and $\hat{e_{t,n_0}}$ not changed **do**
  $\hat{e_{t,n_1}} = \phi(\hat{b}) - \phi(\hat{a}+\hat{b}) + \sum_m y_{t,n,m}(\phi(\hat{c}) - \phi(\hat{c}+\hat{d})) + \sum_m (1-p_{t,n,m})(\phi(\hat{d}) - \phi(\hat{c}+\hat{d}))$;
  $a \leftarrow a + \sum_{t,n} e_{t,n}$;
  $\hat{b} = b + \sum_{t,n}(1 - e_{t,n})$;
  $\hat{c} = c + (1 - e_{t,n}) \sum_{t,n,m} p_{t,n,m}$;
  $\hat{d} = d + (1 - e_{t,n}) \sum_{t,n,m}(1 - p_{t,n,m})$;
  $\hat{e_{t,n_0}} = \phi(\hat{a}) - \phi(\hat{a} + \hat{b}) + \sum_m p_{t,n,m}\mathbb{E}[\alpha_{t,0}] + \sum_m (1 - p_{t,n,m})\mathbb{E}[\alpha_{t,1}])$;
  $\frac{1}{\sigma^2}(\widetilde{m}_t - \widetilde{m}_{t-1})(\frac{\partial \widetilde{m}_t}{\partial \hat{\alpha}_{t,0}} - \frac{\partial \widetilde{m}_{t-1}}{\partial \hat{\alpha}_{t,0}}) = (\sum_t (\sum_n \hat{e_{t,n_0}} \sum_m p_{t,n,m})$
  $\frac{1}{\sigma^2}(\widetilde{m}_t - \widetilde{m}_{t-1})(\frac{\partial \widetilde{m}_t}{\partial \hat{\alpha}_{t,1}} - \frac{\partial \widetilde{m}_{t-1}}{\partial \hat{\alpha}_{t,1}}) = (\sum_t (\sum_n \hat{e_{t,n_0}} \sum_m (1 - p_{t,n,m}))$
end while;
Use $\hat{e_{t,n_0}}$ and $\hat{e_{t,n_1}}$ to re-normalize $\hat{e_{t,n}}$;
return $\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{e_{t,n}}, \hat{\alpha_t}$

---

5 Experiment

5.1 Experimental Setup

The data set used in our experiment is crawled through Microblogging API between 2016 and 2018 using keyword matching. The corpus includes six incidents which all gained great attention on the Microblogging platform. Details of the data set, including the description of each event, the number of comments are shown in Tab. 1. We will make the dataset public upon acceptance.

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

Table 1. Statistics of the data set

In pre-processing, repeated tweets, emoji expressions, http links and mentions (@somebody) are removed. For Chinese word segmentation, we use the jieba NLP tool[2]. We get sentiment words and values through the HOWNET lexicon[3].

5.2 Evaluation of Entity-level Sentiment Extraction

Our first research question is whether incorporating proximity information enhances entity-level sentiment extraction. To answer this question, we generate a ground truth of entity-level sentiment polarity for each tweet by first randomly sampling tweets for all incidents. Next, five human volunteers are asked to judge the sentiment polarity of each tweet on each relevant entity. In order to make the ground truth as accurate as possible, the tweet is added to the ground truth if only five volunteers agree with each other. As a result, we create a sentiment polarity standard data set containing 2000 tweets. We will make the ground truth publiclly available upon acceptance.

We compared our method to three state-of-the-art methods. (1) SentiStrength [6]: a classic algorithm for sentiment extraction, (2) SentiStrength-SE [22]: a different lexicon is designed for a different domain, (3) SentCR [14]: a supervised learning method designed for code review comments. We also provide results obtained by our proposed method with four distance kernels, namely (4) PESE-G: sentiment extraction with Gaussian distance kernel: (5) PESE-T: sentiment extraction with Triangle distance kernel: (6)PESE-C: sentiment extraction with Cosine (Hamming) distance kernel: (7)PESE-I: sentiment extraction with Circle distance kernel. After ten-fold cross-validation, parameter σ is set to be $\sigma = 21$ for PESE methods.

The evaluation metric is accuracy, which is the ratio of number of tweets that are correctly judged versus total number of tweets.

[2] https://github.com/fxsjy/jieba
[3] http://www.keenage.com/

As shown in Table 2, all PESE variants outperform the comparable methods. PESE-G achieves the highest accuracy averaged over all incidents. We also observe that positive polarities are usually more difficult to identify, with lower accuracies by most methods. To gain some insights about the effect of text length, we split our dataset into three divisions: tweets with less than 20 words, tweets with 20 $\sim$ 40 words, and long tweets with more than 40 words. We observe that, as the tweet gets longer, the accuracy of our proposed method achieves better results. Our observation is consistent to our assumption that our proposed method is more effective for long text.

Methods	Comments length					
	0-20		20-40		40+	
	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
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**

Table 2. Averaged accuracy by different sentiment extraction methods.

5.3 Parameter Influence Of Distance Function

In this subsection we study the effect of parameter σ to the proposed PESE method. We use the same ground truth and evaluation metric as in Sec 5.2. We tune $\sigma = 1, 2 \cdots, 30$ and report the average accuracy over all incidents in Fig. 3

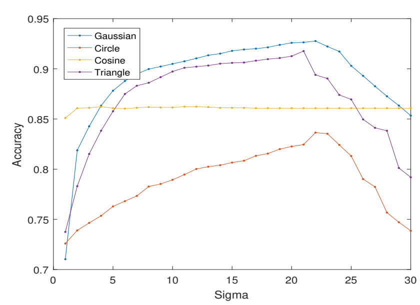


Fig. 3. For different kernel functions, the relationship between sentiment analysis accuracy and σ

As shown in Fig 3, PESE-C is more stable and less affected by σ. Other three methods are more sensitive to the value of σ. For example, PESE-G achieves the highest accuracy of 92.77% with $\sigma = 21$.

5.4 Evaluation of Public Opinion Model

First, we analyze the performance of shift detection. The ground truth of shift points for each incident is manually generated. Five volunteers observed the tweets at each time point and judged whether belongs to shift point. The final shift point gold standard is selected by taking a simple majority vote on each time points.

We compare our method with three state-of-the-art sentiment tracking methods. (1)POMS [11]: measure sentiment polarity and calculate shift points with residuals. (2)FB-LDA [12]: extract foreground topics from tweets in the variation period. (3)LDA & KL-divergence [2]: extract topics in the time window and tank the topic based on their contribution.

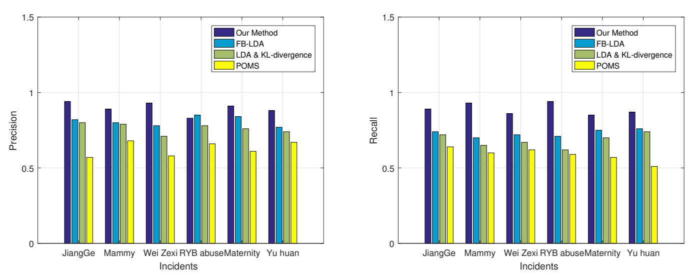
We choose precision, recall and perplexity as evaluation metrics. The precision metric is shown as follows:

$$Precision = \frac{time\ points\ with\ correct\ judgment}{time\ points\ with\ judgment} \tag{7}$$

The purpose of precision can help us to judge how many time points are correctly judged.

$$Recall = \frac{time\ points\ with\ judgment}{time\ points\ with\ judgment\ in\ the\ ground\ truth} \tag{8}$$

Recall tells us how many time points we can judge whether belongs to shift point in the ground-truth.



(a) Precision (b) Recall

Fig. 4. Comparative performance of shift detection

As shown in Fig 4, Our model has achieved the best results in detecting shift points. For the six events we selected, our model can achieve an average of 86% precision and 90% recall. In contrast, the average precision and recall of FB-LDA are 81% and 73%. For LDA & KL-divergence are 76% and 68%. POMS performs the worst which the average precision and recall are 52% and 59%. These results show that our model can detect the public opinion shift in the incident more accurately.

Then, we analyze the performance of predictive. We choose perplexity as evaluation metric.

$$perplexity_{pw} = \exp\left\{-\frac{\sum_{d\in D} logp(w_d)}{\sum_{d\in D} N_d}\right\} \tag{9}$$

Perplexity is a measurement of how well a probability distribution or probability model predicts a sample. A low perplexity indicates the probability distribution is good at predicting the sample. $p(w_d)$ is the probability of the d-th word. N is the length of the sentence.

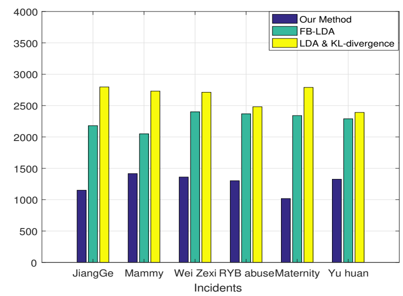
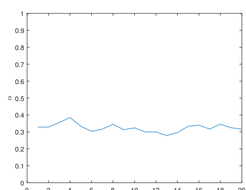


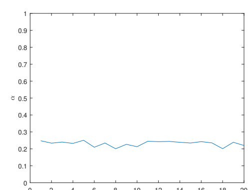
Fig. 5. Averaged per-word predictive perplexity comparison

As shown in Fig 5, Compared to the other two methods, our model has a smaller averaged per-word perplexity in all six incidents. This result indicates our model has better predictive performance.

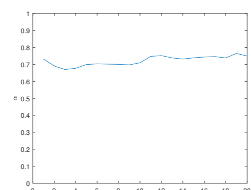
Finally, we offer visualization of the background sentiment evolution for the six incidents.



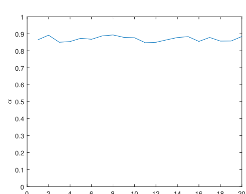
(a) Maternity Fall



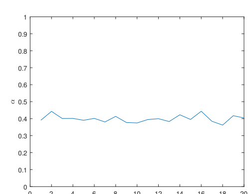
(b) Kindergarten Abuse



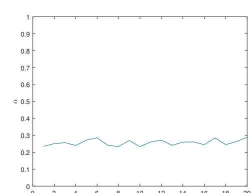
(c) Mammy Arson



(d) Jiang Ge Murder



(e) Yu Huan Murder



(f) Death Of Wei Zexi

Fig. 6. Changes in the α value of six incidents

As shown in Fig 6, the α which represents the distribution of background opinion in the incidents is smoothly and slowly changing in all six incidents as we considered before. The α of Death Of Wei Zexi, Kindergarten Abuse, Maternity Fall, and Yu Huan Murder is less than 0.5, indicating that negative sentiment is the back ground opinion to these incidents. The α of Jiang Ge Murder and

Mammy Arson are greater than 0.5, indicating that positive sentiment is the back ground opinion to these incidents.

6 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 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.

References

1. Cheung A S Y. The Battle of Microblogging for Legal Justice in China. Ssrn Electronic Journal, 2014.
2. Anastasia Giachanou, Ida Mele and Fabio Crestani. Explaining Sentiment Spikes in Twitter. In CIKM '16, PP. 2263-2268, 2016.
3. Anastasia Giachanou, Ida Mele and Fabio Crestani. A Collection for Detecting Triggers of Sentiment Spikes. In SIGIR '17, PP. 1249-1252, 2017.
4. Mikalai Tsytsarau, Themis Palpanas and Malu Castellanos. Dynamics of news events and social media reaction. In KDD '14, PP. 901-910, 2014.
5. Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in Twitter Events. In Journal, pp. 406-418, 2011.
6. Mike Thelwall,Kevan Buckley,Georgios Paltoglou,Di Cai,Arvid Kappas. Sentiment strength detection in short informal text. In Journa, pp. 2544–2558, 2010.
7. Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. J. Am. Soc. Inform. Sci. Technol. 63, 1 (2012), 163–173.
8. Reynier Ortega, Adrian Fonseca, and Andres Montoyo. 2013. SSA-UO: Unsupervised twitter sentiment analysis. In Proceedings of the 7th International Workshop on Semantic Evaluation.
9. Anastasia Giachanou, Fabio Crestani. 2011. Tracking Sentiment by Time Series Analysis. In SIGIR '16, pp. 1037-1040, 2016
10. X. An, R. A. Ganguly, Y. Fang, B. S. Scyphers, M. A. Hunter, and G. J. Tracking Climate Change Opinions from Twitter Data. In KDD'14, 2014.
11. J. Bollen, A. Pepe, and H. Mao. Modeling Public Mood and Emotion : Twitter Sentiment and Socio-Economic Phenomena. In ICWSM'11, pages 450-453, 2011.
12. Shulong Tan, Yang Li, Huan Sun, Ziyu Guan. Interpreting the Public Sentiment Variations on Twitter. In Journal'14, 2014.
13. C. S. Montero, H. Haddad, M. Mozgovoy, and C. B. Ali. Detecting the likely causes behind the emotion spikes of influential twitter users. In CICLing'16, 2016.
14. Toufique Ahmed,Amiangshu Bosu,Anindya Iqbal,Shahram Rahimi. SentiCR:A Customized Sentiment Analysis Tool for Code Review Interactions. In ASE '17.

15. Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification Using Distant Supervision. Technical Report. Standford.
16. Luciano Barbosa and Junlan Feng.2010. Robust sentiment detection on twitter from biased and noisy data. In COLING '10, pp. 36-44, 2010.
17. Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In ICWSM'11, pp. 538-541.
18. Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In LREC '10, pp. 1320-1326, 2010.
19. Dmitry Davidov, Oren Tsur, and Ari Rappoport.2010. Enhanced sentiment learning using twitter hashtags and smileys. In COLING '10, pp. 241-249, 2010.
20. Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2010. Sentiment analysis of twitter data. In LCM'11, pp. 30–38.
21. C. J. Hutto and E. Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In AAAI'14.
22. Md Rakibul Islam and Minhaz F Zibran. 2017. Leveraging automated sentiment analysis in software engineering. In Proceedings of IEEE Press, 203-214.
23. Song Feng, Ritwik Bose, and Yejin Choi. 2011. Learning general connotation of words using graph-based algorithms. In EMNLP '11, pp. 1092-1103, 2011.
24. Tiantian Zhang, Bin Wu. A Method for Local Community Detection by Finding Core Nodes. In Proceedings of the 2012 IEEE/ACM
25. Yuanhua Lv, ChengXiang Zhai. Positional language models for information retrieval. In SIRIG '09, pp. 299-306, 2009.