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 sentimen shift is not a trivial problem because the evolution of the background sentiment can not be ignored. We propose to simutanously 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

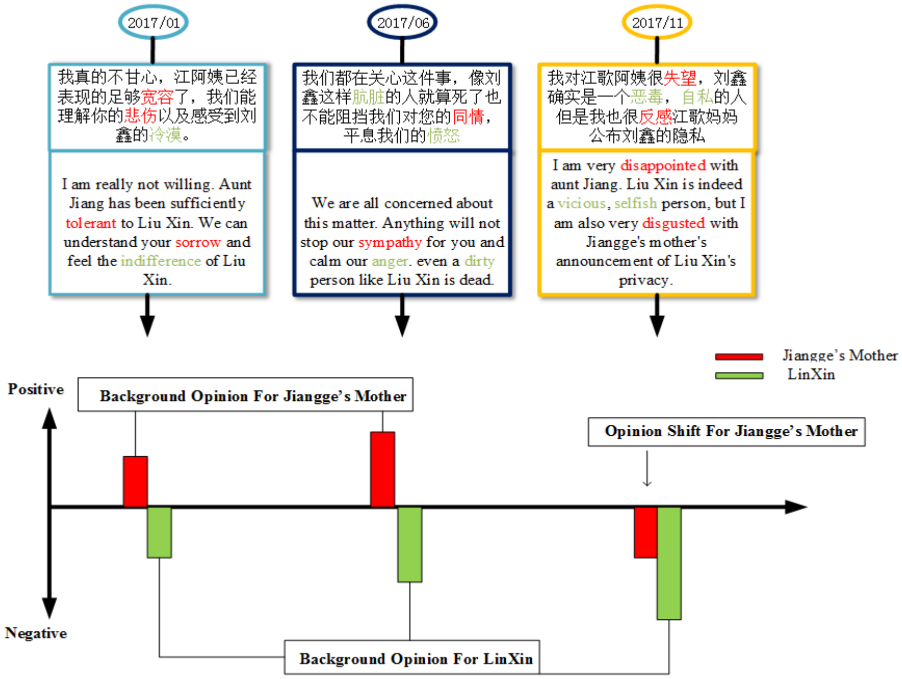
1 Introduction

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, 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, and 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 murdur case of Jiang Ge[1] leads to visualization of the

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

evolution pattern of public sentiment towards the victim's mother (Jiang) and the victim's roomate (Liu). A sentiment shift is also detected in the third time point.


2017/01
2017/06
2017/11
🟥 Jiangge's Mother
🟩 LinXin
Positive
Negative
Background Opinion For Jiangge's Mother
Opinion Shift For Jiangge's Mother
Background Opinion For LinXin

我真的不甘心，江阿姨已经表现的足够宽容了，我们能理解你的悲伤以及感受到刘鑫的冷漠。

I am really not willing. Aunt Jiang has been sufficiently tolerant to Liu Xin. We can understand your sorrow and feel the indifference of Liu Xin.

我们都在关心这件事，像刘鑫这样肮脏的人就算死了也不能阻挡我们对您的同情，平息我们的愤怒

We are all concerned about this matter. Anything will not stop our sympathy for you and calm our anger. even a dirty person like Liu Xin is dead.

我对江歌阿姨很失望，刘鑫确实是一个恶毒，自私的人但是我也很反感江歌妈妈公布刘鑫的隐私

I am very disappointed with aunt Jiang. Liu Xin is indeed a vicious, selfish person, but I am also very disgusted with Jiangge's mother's announcement of Liu Xin's privacy.

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 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 encourage 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 simutanously 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 negatie 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 measurment [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 asu nigrams 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]. SentiSterength2 [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. SenciStrength-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 words 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 toghether in a microblogging post.

3 Proximity-based Entity-level Sentiment Extraction

In this section, we describe Proximity-based Entity-level Sentiment Extraction (PESE). Given xxx, our aim is to xxx.

3.1 Distance Function

Our basic assumption is that the position of a sentiment word influences the performance of 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 [], we use four distance kernel functions to compute influence of sentiment words on entities, namely Gaussian, Triangle, Cosine, and Circle:

1. **Gaussian kernel**

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

2. **Triangle kernel**

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

3. Cosine (Hamming) kernel

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

4. Circle kernel

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

where 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 are governed by one parameter σ, which is tunned in the experiment.

We use the average of all sentiment words on the entity as the sentiment polarity of the sentence.

3.2 Calculate Entity level Sentiment polarity

To calculate the sentiment value for an entity given a post, we first calculate the influence of each sentiment words on the entity:

$$s_i = (-1)^n \cdot d_i \cdot v_i \cdot k(p_i, j), \tag{5}$$

where i is the number of this sentiment word. n is the number of gainsay words between this sentiment 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 sentiment value of this sentiment word. k is the distance function. p_i is the location of this sentiment word. j is the location of the entity.

We calculate the influence of all sentiment words on the entity according to Equ. 5, 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 entity has a negative emotion.

4 Public Opinion Model

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 preprocessed 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

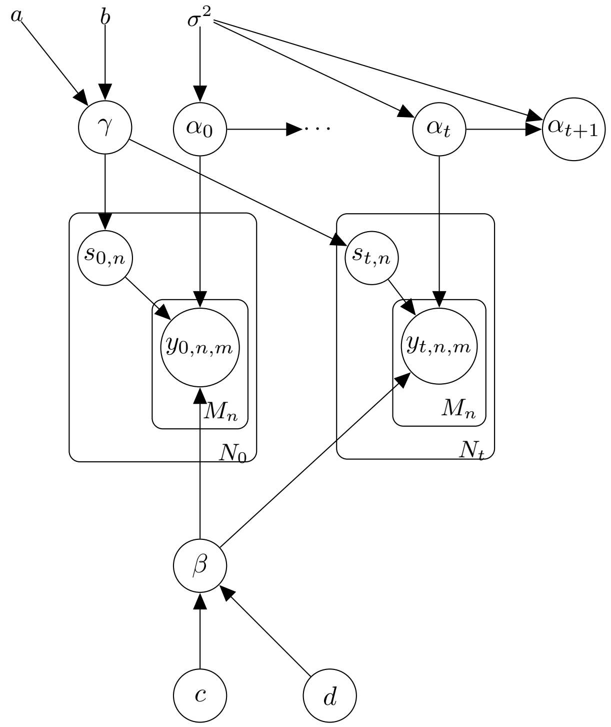


Fig. 2. Plate notation of the proposed opinion evolution model

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

- 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

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

In the nutshell, the optimization algorithm is variational inference. Thus we make the following assumptions.

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

We implement variational inference. In each iteration, the parameters α, are updated by:

(6)

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 6 incidents, which are all top-level incidents at the time. 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.

In pre-processing, repeated tweets, emoji expressions, http links and mentions (@somebody) are removed. For Chinese word segmentation, we use the jieba NLP tool [].

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

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 randomly xxx.

Ground truth is still manually marked by five people.

We compared our method to three state-of-the-art methods. (1) SentiStrength []: a xxx tool which xxx, (2) SentiStrength-SE []: and (3) SentCR []:. We also provide results obtained by our proposed method with four distance kernels, namely (4) PESE-G: sentiment extraction with Gaussian distance kernel: (5) ... Parameter σ is tuned by cross-validation, where $\sigma =$ for PESE-G ...

The evaluation metrics are accuracy, which is computed by *xxxequation*.

As shown in Table 2, xxx achieves the highest accuracy on all incidents. xxx is better than xxx. 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, Our observation is consistent to our assumption that,

Table 2. Comparison of sentiment extraction results

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-G	0.8477	0.8588	0.8539	0.8771	0.8862	0.9289
PESE-T	0.8477	0.8588	0.8539	0.8771	0.8862	0.9289
PESE-C	0.8477	0.8588	0.8539	0.8771	0.8862	0.9289
PESE-I	0.8477	0.8588	0.8539	0.8771	0.8862	0.9289

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 extraction results. In order to verify our conjecture,

5.3 Parameter Selection For Distance Function

Here we will get the form of the distance function and determine the parameters of the it through experiments. We randomly selected tweets from the corpus of six events, and then manually marked by five people. If five people have inconsistent judgments on the sentiment polarity of a tweet, delete the tweet. We got the position of the entity by keyword matching. Finally we got an experimental data set containing 2000 tweets. We use different distance functions to calculate the sentiment extraction 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

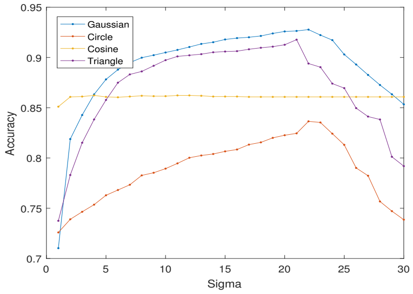


Fig. 3. For different kernel functions, the relationship between sentiment analysis accuracy and σ

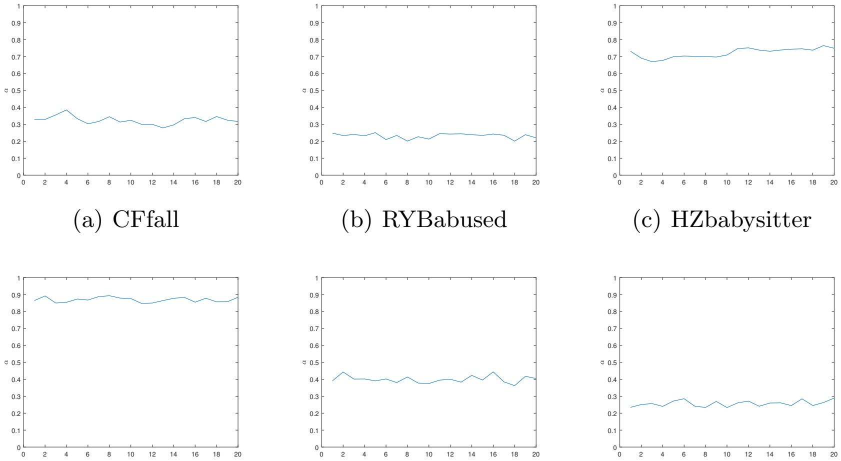
Among the four kernel functions, the Cosine kernel function is less affected by sigma. The accuracy of sentiment extraction corresponding to the other three kernel functions increases first and then decreases, and all three kernel functions get the optimal value when the value of sigma is approximately 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 extraction accuracy of the data set reached 92.77 percent.

5.4 Evaluation of Public Opinion Model

Here, we show the advantages of our model in simulating the evolution of public opinion.

Parameter α represents the distribution of background opinion in the incidents. As shown in Fig 4, the α is smoothly and slowly changing in all six incidents as we considered before. The α of WZXHospital, RYBabused, CFfall, and SDhumiliation is less than 0.5, indicating that negative sentiment is the publics backgronund opinion to these incidents. The α of JGmurder and HZbabysitter are greater than 0.5, indicating that positive sentiment is the publics backgronund opinion to these incidents.

We choose precision and recall as evaluation metrics to evaluate the performance of finding shift points. We compare our model with POMS, FB-LDA and RCB-LDA. POMS use the sentiment extraction tool to measure the sentiment polarity at numerous time points and use residual to extract shift points. FB-LDA and RCB-LDA are topic-based methods. The ground truth of shift points

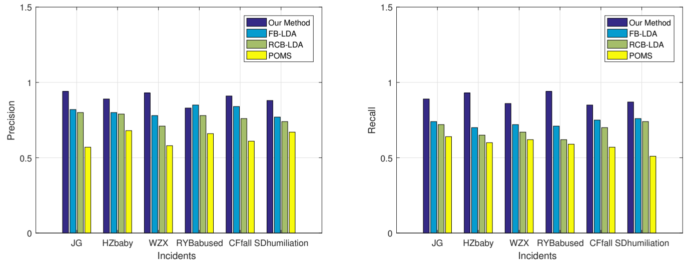


(a) CFfall (b) RYBabused (c) HZbabysitter

(d) JGmurder (e) SDhumiliation (f) WZXHospitaln

Fig. 4. Changes in the α value of six events

are manually determined. We have five people voting for Whether each time point of the six incidents belongs to the shift point.



(a) Precision (b) Recall

Fig. 5. Comparative preformance of shift detection

As shown in Fig 5, 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 RCB-LDA 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 accurately detect the public opinion shift in the incident.

We evaluated the model predictive performance with perplexity which is a evaluation standard for measuring the predictive performance of a model. For each incident, We compute the averaged per-word perplexity over the time line of our model, FB-LDA and RCB-LDA. As shown in Fig 6, 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.

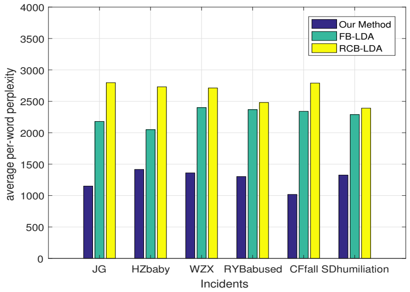


Fig. 6. Averaged per-word predictive perplexity comparison

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. 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.

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, and Arvid Kappas. Sentiment strength detection in short informal text. Journal of the Association for Information Science and Technology 61, 12 (2010), 25442558.
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), 163173.
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 KDD14: Workshop on Data Science for Social Good, 2014.
11. J. Bollen, A. Pepe, and H. Mao. Modeling Public Mood and Emotion : Twitter Sentiment and Socio-Economic Phenomena. In ICWSM11, pages 450-453, 2011.
12. Shulong Tan, Yang Li, Huan Sun, Ziyu Guan. Interpreting the Public Sentiment Variations on Twitter. In Journal14, 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 CICLing16, 2016.
14. Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, Shahram Rahimi. SentiCR: A Customized Sentiment Analysis Tool for Code Review Interactions. In ASE 17, 2017.
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 ICWSM11, 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 LCM11, pp. 3038.
21. C. J. Hutto and E. Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In AAAI14.
22. 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, 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 International Conference on Advances in Social Networks Analysis and Mining.