Looking into the Past: Evaluating the Effect of Time Gaps in a Personalized Sentiment Model

Siwen Guo
ILIAS Research Lab, CSC,
University of Luxembourg
Esch-sur-Alzette, Luxembourg
siwen.guo@uni.lu

Sviatlana Höhn
ILIAS Research Lab, CSC,
University of Luxembourg
Esch-sur-Alzette, Luxembourg
sviatlana.hoehn@uni.lu

Christoph Schommer
ILIAS Research Lab, CSC,
University of Luxembourg
Esch-sur-Alzette, Luxembourg
christoph.schommer@uni.lu

ABSTRACT

This paper concerns personalized sentiment analysis, which aims at improving the prediction of the sentiment expressed in a piece of text by considering individualities. Mostly, this is done by relating to a person's past expressions (or opinions), however the time gaps between the messages are not considered in the existing works. We argue that the opinion at a specific time point is affected more by recent opinions that contain related content than the earlier or unrelated ones, thus a sentiment model ought to include such information in the analysis. By using a recurrent neural network with an attention layer as a basic model, we introduce three cases to integrate time gaps in the model. Evaluated on Twitter data with frequent users, we have found that the performance is improved the most by including the time information in the Hawkes process, and it is also more effective to add the time information in the attention layer than at the input.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; • **Computing methodologies** → **Natural language processing**; **Neural networks**;

KEYWORDS

Sentiment Analysis, Personalized Model, Recurrent Neural Network, Attention Mechanism, Temporal Information.

ACM Reference Format:

Siwen Guo, Sviatlana Höhn, and Christoph Schommer. 2019. Looking into the Past: Evaluating the Effect of Time Gaps in a Personalized Sentiment Model. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus,* 4 pages. https://doi.org/10.1145/3297280.3297572

1 INTRODUCTION

Personalized sentiment analysis is the task to determine the sentiment orientation of a piece of text by considering the individuality of the sentiment holder. A person's individuality can be reflected in the texts in different aspects such as the person's lexical choices, personal interests and preferences. It is shown in a number of researches that modeling individualities has the potential to help recognize and categorize the sentiment holder's sentiment in text [5, 7, 16]. In order to capture such individuality, a person's past opinions are taken into account in the analysis. However, people's opinions may vary from time to time, in which aspect a person's recent opinions can have more impact on the person's current opinion than the earlier ones. In the previous studies, the different time gaps between a person's past opinions were not seen as a feature that might influence the prediction. In this paper, we argue that the time gaps contribute as an important factor in determining the current opinion. We exemplarily take text messages (tweets) from a number of users who have posted at least 20 times before a predefined date on Twitter over 81 days, and time gaps are employed in three distinct ways in the learning process in order to enhance the prediction of the users' sentiments.

Several works have applied recurrent neural networks (RNNs) for personalized sentiment analysis [3–5]. Long short-term memory (LSTM) and attention mechanism were used in Chen et al. [3] and Dou [5] for analyzing product reviews. The former uses user and product information in the attention layers which are applied on top of the LSTM layers to generate sentence- and document level representations. The latter uses LSTM to generate document embeddings and applies multiple computational layers, each consisting of an attention layer and a linear layer, to combine the embeddings. Both approaches have shown positive results that prove the effectiveness of integrating user information in the analysis. Although traditional RNNs are popular for extracting patterns from temporal sequences, they do not have the ability to analyze irregularly emerging events by design. Naturally, such events can be separated into different time intervals, and zero-padding can be applied for empty slots; however this solution is ill-suited when the emerging of events is hardly predictable. In order to model asynchronous events, Neil et al. [13] proposed a phased LSTM, which altered the design of the traditional LSTM by adding a time gate. The time gate is capable of controlling the update of the cell state and the output, and the learning phase is accelerated with this design. However, this approach is not suitable in our task because 1. the publishing of a post is not systematic; 2. in the approach, the time gap is not connected with other information in the phased LSTM cell: The impact of the time gap only matters when the tweet from the past and the current tweet are related. Therefore, alternative approaches that consider precise time points and the connection between the texts are yet to be explored.

Based on a basic personalized sentiment model that utilizes RNNs with an attention layer, we compare the performance for:

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3297572

Case I: Adding the encoded time gaps in the input sequence.
Case II: Using the encoded time gaps to reshape the context vector at the attention layer.
Case III: Shaping the output of the attention layer with Hawkes process [9].

2 PERSONALIZED SENTIMENT MODEL

In this section, we introduce the basic personalized sentiment model (Figure 1(a)), which is designed to accomplish the goal of including user information in the system.

2.1 Input Sequence

In Figure 1(a), each input sequence X_{tweet} consists of the current tweet and a number of past tweets from the same user. Assume x_{t_i} is the tweet at time t_i, then a matrix of $(n+1)$ tweets $X_{t_0} = [x_{t_{-n}}, ..., x_{t_{-2}}, x_{t_{-1}}, x_{t_0}]$ where x_{t_0} is the current tweet, and $x_{t_{-n}}, ..., x_{t_{-2}}, x_{t_{-1}}$ are the n past tweets published by the same user and ordered by the publishing time. Each tweet x_{t_i} is represented by a vector indicating the concepts, topics, and negation cues that are appeared in the text at time t_i, together with the user identifier. The components used in the representation are as in [8]: Concepts are extracted according to an external source and contain conceptual and affective information of the text [2]; Main topics are extracted from the text because the relation between the opinion and the target of the opinion has shown to be beneficial to the personalized model; Negations are included in the representation for their ability to invert the orientation of the sentiment. A lexicon of negation cues that follows Reitan et al. [15] is taken for this purpose. Personalized models generally suffer from the issue of data sparsity caused by different frequencies of users posting messages on the social platforms. Similar to the notion employed by Johnson et al. [10] for multilingual translation, we handle this problem by implementing a personalized sentiment model [8] which adds a user identifier in the representation in order to include the user information in the system. In this way, the individuality can be analyzed by the model and the comparison between users is possible.

2.2 The Basic Model

As shown in Figure 1(a), the basic personalized model first applies a tweet encoder, which is a fully connected layer, to embed each input tweet. Then, the embeddings of tweets from different time points are fed to the RNNs which contain three RNN layers with LSTM cells. Attention mechanism is employed on top of the RNNs, which is defined as follows:

$$u_i = \tanh(W_t h_i + b_t) \tag{1}$$

$$a_i = u_i^{\top} w_s \tag{2}$$

$$\lambda_i = softmax(a_i) h_i \tag{3}$$

$$v = \sum_i \lambda_i \tag{4}$$

where $softmax(x_i) = e^{x_i} / \sum_j e^{x_j}$. h_i is the i-th output of the RNNs, u_i is a hidden representation of h_i, and w_s is a 'context vector' that is randomly initialized in this model and jointly learned with other weights during the training phase. v is the output of the attention layer that contains all the information of the tweets from different time points. Finally, the output y_t of the model in Figure 1(a) is the sentiment orientation of the current tweet at time t_0.

3 MODEL EXTENSION WITH TIME GAPS

The basic model introduced in the last section is extended in three ways in order to test experimentally the influence of analyzing time gaps on personalized sentiment modeling. We use the same basic structure and tweet representation for each extended model. The following cases are designed to include the time information in the model such that the time gap interacts with the content of the tweet, and they jointly influence the final output and thus the prediction. The structure of these models is illustrated in Figure 1.

3.1 Case I: Integrating Time in the Input

The model in Figure 1(b) uses a time encoder — a fully connected network, to embed different time gaps. A time gap Δt_i at time t_i is the time difference between the past tweet x_{t_i} and the current tweet x_{t_0}. The time gap is calculated as the number of hours between the past and the current tweet as a float positive value. Afterwards, each encoded time gap is concatenated with the encoded tweet representation according to the timestamp. The concatenated sequence is fed to the RNNs so that the network is able to learn the connections between the content of the tweets and their publishing time. The time embeddings act as an auxiliary input in comparison to the basic model in Figure 1(a).

3.2 Case II: Integrating Time in the Attention

In Figure 1(c), the time gaps are encoded in the same way as in Case I, however the output influences the attention layer directly that the Equation (3) is replaced by the following equation:

$$\lambda_i' = softmax(a_i E(\Delta t_i)) h_i \tag{5}$$

where $E(\Delta t_i)$ is the i-th encoded time gap. In this way, the impact of the attention value is decreased when the time gap is large (a decreasing function can be learned by the time encoder). Again, v sums up λ_i' over time step i.

3.3 Case III: Integrating Time with Hawkes Process

Hawkes process is a one-dimensional point process that can be used for modeling the arrivals of events over time. It has a *self-exciting* property that an occurrence of an event excites the process and boosts the probability of a future arrival of the event in a period of time [12]. Such a property is advantageous for tasks such as earthquake modeling [14] and retweet prediction [11]. In this task, we argue that an opinion expressed at time t_i can be seen as an 'event' that its appearance positively affects the opinion at (future) time t_0, and the effect decays according to the time gap Δt_i. In order to consider such effect for the opinions with the same or similar targets (topics), we revise the traditional Hawkes process, and replace Equation (4) in the attention layer by the following equation:

$$v' = \sum_{i:\Delta t_i \geqslant 0} (\lambda_i + \varepsilon \lambda_i' e^{-\beta \Delta t_i}) \tag{6}$$

where $\lambda_i' = max(\lambda_i, 0)$. With λ_i' taking non-negative values, only the effect of the past tweets that have related content to the current

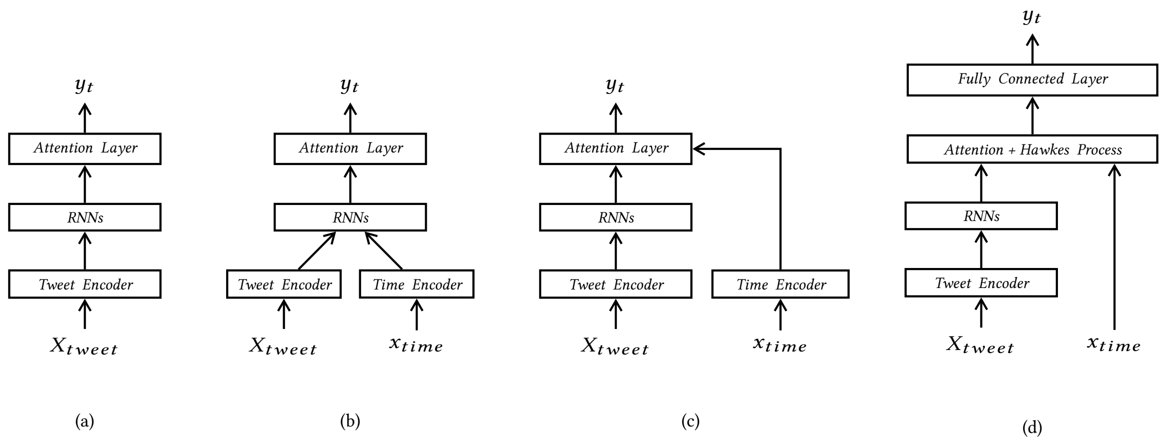

Figure 1: The basic personalized sentiment model (a), Case I (b), Case II (c), and Case III (d). X_{tweet} corresponds to a number of tweets at the past and the current time points, x_{time} is the time gaps between each past tweet and the current tweet, and y_t is the sentiment label at the current time.

Table 1: Model evaluation while using all the user data, or the users with frequency not less than 50 or 100.

Model	Pos. F1	Neg. F1	Accuracy	Accuracy user frequency⩾ 50	Accuracy user frequency⩾ 100
RNNs	0.7467	0.7401	0.7435	0.7759	0.8968
Basic Model	0.7500	0.7549	0.7526	0.7852	0.8962
Case I	0.7494	0.7625	0.7562	0.7936	**0.9102**
Case II	0.7510	0.7655	0.7586	0.7880	0.8962
Case III	**0.7553**	**0.7670**	**0.7613**	**0.7995**	0.9022

one is concerned. ε indicates how important such decay effect is to the system, and β indicates the decay rate. The exponential decay function is widely used for the Hawkes process and is employed in this model based on the advantages explained in Bacry et al. [1]. Existing works have used fixed values for ε and β [8], whereas in our implementation, ε and β are empirically initialized and jointly learned with other weights during the training phase. This indicates that the weights needed to be trained at the attention layer become $[W_t, w_s, \varepsilon, \beta]$ instead of $[W_t, w_s]$. In the end, a fully connected layer is applied to regularize the shaped attention output.

4 EXPERIMENTS AND DISCUSSIONS

We compare the performance of the introduced cases with the RNNs and the basic model. We also report the results for users with different frequencies in order to gain more insights to the task.

4.1 Dataset and Technical Setup

We take the Sentiment140[1] corpus which contains 1,600,000 training samples, however only tweets of users who have published at least 20 times before a pre-defined date are used in the experiments, which results in 122,000 tweets in total. In the dataset, 22.4% of the tweets are generated by users with frequency not less than 50 which results in 27304 instances; 6.1% are generated by users with frequency not less than 100 which results in 7449 instances. The tweets are automatically classified as positive or negative using emoticons [6]. In this task, manually labeled data do not contain sufficient frequent users to explore the individuality. During the experiments, the dataset is split into a training set, a validation set, and a test set given two chosen timestamps. Concepts are extracted from SenticNet5[2] which has 100,000 common-sense concepts in total. The implementation is conducted using Keras[3] with Tensorflow[4] back-end. Experiments are executed five times and average results are shown. Detailed settings can be found in the appendix A.

4.2 Model Comparison

Table 1 shows the performance of the five models. Comparing the F1-scores for the positive and negative classes as well as the accuracy, the RNNs model performs the worst which is followed by the basic model. These two models do not associate time information in the model, thereby revealing the significance of including such information in the analysis. The three cases provide competitive results while Case III with Hawkes process is slightly better than the others. In fact, the improvement of Case III over the models without time information passes the t-test (with p<0.05). Case II and III add the time at the attention layer thus affect the output more directly than Case I. In Case II, the attention value is modified by the time using a multiplication, while in Case III the employed decay (or excitation) process results in better performance. However, the time encoder in Case I and II is also capable of forming a decay function,

[1] http://help.sentiment140.com/for-students, last seen on August 24, 2018

[2] http://sentic.net/, last seen on August 24, 2018

[3] https://keras.io/, last seen on August 24, 2018

[4] https://www.tensorflow.org/, last seen on August 24, 2018

and more sophisticated structures to enhance the results are yet to discover. The differences among the three cases are not significant in all categories (e.g., the difference of Pos. F1 between Case I and III is significant), which motivates us for future exploration.

We also compare the performance of the models for users with different publishing frequencies. Results show that the more frequent a user tweets during the test period (19 days), the more accurate the models can predict for the user. Taking the users with frequency not less than 50 or 100 as examples, the improvement of the accuracy with respect to this conclusion is significant for all the models. The accuracies for the users tweeted not less than 100 times are high since the topics of the tweets can be highly related within a short period and the time gaps are comparably low. Case I offers the best result in this range indicating that it is easier for the network to find the relations from the input when associating with highly frequent users.

5 CONCLUSION AND FUTURE WORK

We investigate the impact of time gaps between messages in personalized sentiment analysis. The time information is integrated based on a basic model with RNNs and an attention model in three ways. Evaluated on a Twitter corpus with frequent users, we found that the time indeed positively affects the final prediction, and it is more effective to add the time information in the attention layer that is closer to the output. Employing a Hawkes process with attention mechanism has shown promising results, and the user frequency of publishing messages also has a noticeable impact on the performance. Furthermore, time gaps can be seen as context information, which implies that other sorts of context information may also be beneficial to the personalized model.

The results bring new perspectives in discovering the relation between the creation time and the content of an opinion. Other ways to include time in the model can be proposed, for instance, using variants of Hawkes process with different decay functions. More experiments and comparisons are planned in the future work. Various types of data, *e.g.*, a longer duration of messages, other sources of text and other languages, can also be tested. This work also provides a solution for similar tasks that require modeling time differences between events for temporal sequences.

A DATA STATISTICS & MODEL SETTINGS

Table 2: Statistics of the dataset used in the models.

Dataset	Sentiment140	
Polarity	Positive	79,009
	Negative	42,991
	Total	122,000
# User	2,369 (frequency $\geqslant$ 20)	
# Topic	appears min. 10 times: 761	
Starting Time	2009-04-06 22:19:57	
Ending Time	2009-06-25 10:28:28	
# Training Samples	82,361	
Timestamp 1	2009-06-03 00:00:00	
# Validation Samples	16,437	
Timestamp 2	2009-06-07 00:00:00	
# Test Samples	23,202	

Table 3: Technical setup of the models.

Tweet Encoder	Fully-connected Layer	64 nodes
	Activation Function	tanh
	Encoded Dimension	128
Time Encoder	Fully-connected Layer	20 nodes
	Activation Function	tanh
	Encoded Dimension	20
RNNs	Timesteps n	20
	RNN_1	64 LSTM cells
	RNN_2	64 LSTM cells
	RNN_3	32 LSTM cells
	Recurrent Dropout	0.4
Hawkes Process	Initial ε	0.01
	Initial β	0.001
	Time Unit	hour
	Fully-connected Layer	32 nodes
	Activation Function	tanh

REFERENCES

[1] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity* 1, 01 (2015), 1550005.

[2] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. 2018. SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*.

[3] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1650–1659.

[4] Tao Chen, Ruifeng Xu, Yulan He, Yunqing Xia, and Xuan Wang. 2016. Learning user and product distributed representations using a sequence model for sentiment analysis. *IEEE Computational Intelligence Magazine* 11, 3 (2016), 34–44.

[5] Zi-Yi Dou. 2017. Capturing User and Product Information for Document Level Sentiment Analysis with Deep Memory Network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 521–526.

[6] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1, 12 (2009).

[7] Lin Gong, Mohammad Al Boni, and Hongning Wang. 2016. Modeling social norms evolution for personalized sentiment classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 855–865.

[8] Siwen Guo, Sviatlana Höhn, Feiyu Xu, and Christoph Schommer. 2019. Personalized Sentiment Analysis and a Framework with Attention-based Hawkes Process Model. In *Agents and Artificial Intelligence: ICAART 2018, LNAI*, Vol. 11352. Springer.

[9] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[10] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* (2016).

[11] Ryota Kobayashi and Renaud Lambiotte. 2016. TiDeH: Time-Dependent Hawkes Process for Predicting Retweet Dynamics.. In *ICWSM*. 191–200.

[12] Patrick J Laub, Thomas Taimre, and Philip K Pollett. 2015. Hawkes processes. *arXiv preprint arXiv:1507.02822* (2015).

[13] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*. 3882–3890.

[14] Yosihiko Ogata. 1998. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics* 50, 2 (1998), 379–402.

[15] Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. 2015. Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 99–108.

[16] Kaisong Song, Shi Feng, Wei Gao, Daling Wang, Ge Yu, and Kam-Fai Wong. 2015. Personalized Sentiment Classification Based on Latent Individuality of Microblog Users.. In *IJCAI*. 2277–2283.