Personalized Sentiment Analysis and a Framework with Attention-based Hawkes Process Model

Siwen Guo[1], Sviatlana Höhn[1], Feiyu Xu[2], and Christoph Schommer[1]

[1] ILIAS Research Lab, CSC, University of Luxembourg, Esch-sur-Alzette, Luxembourg
[2] AI Lab, Lenovo, Beijing, China

Abstract. People use different words when expressing their opinions. Sentiment analysis as a way to automatically detect and categorize people's opinions in text, needs to reflect this diversity and individuality. One possible approach to analyze such traits is to take a person's past opinions into consideration. In practice, such a model can suffer from the data sparsity issue, thus it is difficult to develop. In this article, we take texts from social platforms and propose a preliminary model for evaluating the effectiveness of including user information from the past, and offer a solution for the data sparsity. Furthermore, we present a finer-designed, enhanced model that focuses on frequent users and offers to capture the decay of past opinions using various gaps between the creation time of the text. An attention-based Hawkes process on top of a recurrent neural network is applied for this purpose, and the performance of the model is evaluated with Twitter data. With the proposed framework, positive results are shown which opens up new perspectives for future research.

Keywords: Sentiment Analysis · Hawkes Process · Personalized Model · Attention Network · Recurrent Neural Networks.

1 Introduction

Sentiment analysis is defined in Oxford dictionaries[3] as 'the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.'. This definition outlines three types of information that are essential to the study: the text, the target (topic, product, etc.) and the writer. It also reflects the evolvement of this field from document- or sentence-level [23,37] to aspect-level [6,28] which considers various aspects of a target, and later to an advanced level where the text is not the only source for determining sentiments and the diversity among the writers

[3] https://en.oxforddictionaries.com/definition/sentiment_analysis, last seen on April 19, 2018

(or speakers, users depending on the application) is considered as well. However, the writer of a text is not necessarily the person who holds the sentiment. As in [21], the situation is elaborated with an example that a review of product (target) 'Canon G12 camera' by John Smith contains a piece of text *'I simply love it .... my wife thinks it is too heavy for her.'*. The example shows different opinions from two persons published by John Smith who thinks positively towards the target while his wife holds a negative opinion. An accurate research should involve a study that identifies the holder of a sentiment before generating a sentiment score for it. The negligence of this aspect in sentiment analysis is caused by the lack of demand in most applications where opinions are desired regardless of which persons expressing them. Nevertheless, exceptions exist for the task of establishing user groups or for security reasons where locating the holders is as prioritized as extracting opinions. To simplify the task, sentiment holder or opinion holder is mostly used to indicate the person who publishes the text when it comes to analyze individual behaviors through short messages posted on social platforms.

The significance of considering the sentiment holder is based on the observation that people are diverse and they express their sentiments in distinct ways [29]. Such diversity is caused by many factors such as linguistic and cultural background, expertise and experience. While different lexical choices are made by sentiment holders, a model that is tailored by the individual differences should be built accordingly. We name a model that includes individual differences in sentiment analysis *personalized sentiment model*. Note that we distinguish this task from personality modeling [22] where such diversity is also considered in form of linguistic features in discovering users' personality. On social platforms, another phenomenon is that the entity behind a user account is not necessarily one particular individual — it could be a public account run by a person or a group of persons who represent an organization. It is also possible for a person to have more than one account, e.g. a private account and a work account. In our work, we argue that a person may act or express himself/herself differently while using different accounts, but the way of expressing opinions by the person(s) behind one account tends to be consistent.

One critical issue of generating a model for each user individually is the data sparsity. There is an inconsistency in the frequency of posting messages on social platforms per user. For instance, it is reported in 2016 that Twitter has 700 million annually active users, of which 420 million are quarterly active and 317 million are monthly active[4]. The gap between the numbers shows that the amount of messages (also called 'tweets') published per user is normally in the range of a few to a few thousand with roughly 500 million tweets sent per day[5], and the frequency of the postings varies from user to user. In this article, we introduce a framework with neural networks to model individualities

[4] https://www.fool.com/investing/2016/11/06/twitter-has-700-million-yearly-active-users.aspx, last seen on April 19, 2018

[5] http://www.internetlivestats.com/twitter-statistics/#trend, last seen on April 19, 2018

in expressing opinions, which intrinsically offers a solution for the data sparsity in the setting of social networks. This framework is developed based on the first-stage results of PERSEUS [13] which evaluates the effectiveness of including users' historical text for determining the sentiment of the current text. Twitter data was used for the evaluation in the first stage and was used in the improved framework as well. However, different datasets were applied in the experiments that one was manually labeled and the other was automatically labeled and associated with more frequent users.

Major modifications are done after the first stage of PERSEUS. First, each tweet is represented by a sequence that consists of the concepts, the entities, the negation cues and the user identifier. Instead of using user identifier as a separate node, such a combination of features unifies the input structure for neural networks to extract information easier. Second, the embeddings of the tweets are learned directly through a stacked network with sentiment labels, therefore only one learning process is required. Third, an attention model is used after the recurrent layers to enhance the influence of related content from the past. Finally, the output from the attention model is shaped by Hawkes process that is used to capture the decay of information caused by various gaps between the tweets of a user. Hawkes process [15] is a special kind of point process with a 'self-exciting' character, which is widely used for modeling 'arrivals' of events over time. The usage of Hawkes process varies from earthquake modeling [26] to crime prediction [25], and to financial analysis [1]. As an example close to our study, Hawkes process is also used to predict retweets on Twitter for popularity analysis [19,40]. In our work, we argue that the chance that a user's opinion 'arrives' at a specific time point is affected by the time points at which the user expressed past opinions. While the recurrent network is used to find relations between the content of the tweets from the past, the Hawkes process is used to model the decay of such relations with time. Evaluated with a larger number of tweets of frequent users in a period of time, more comprehensive results are given using this framework.

This article is organized as follows: Section 2 gives discussions of related work; Section 3 introduces the structure of the preliminary personalized sentiment model and the enhanced model, mainly on the design of their input sequences and the description of the recurrent neural network used in the models; in Section 4, we discuss the attention mechanism and Hawkes process, and the possibility to combine them in order to model information decay in personalized sentiment analysis; Section 5 presents the technical setup of our experiments, the datasets used to evaluate the models, and the baselines for the model comparison; evaluation results and findings are reported and discussed in Section 6; we conclude our work in Section 7 and give an outlook on future research.

2 Related Work

Most academic contributions in sentiment analysis focus on population-level approaches [8,30]. Nevertheless, there are a number of studies that consider the

diversity of people and apply such traits in distinct ways to improve the performance. Gong et al. [10] propose an adaptation from a global sentiment model to personalized models assuming that people's opinions are shaped by 'social norms'. By using such a global model, the issue with data sparsity is alleviated while individualities are included by performing a series of linear transformations based on the shared model. Later on, Gong et al. argue that like-minded people tend to form groups and conjointly establish group norms even when there are no interactions between the people in the same group [11]. This argument shifts their study from per-user basis to per-group. The concept of user groups is also explored in another work by Song et al. [32], where user following information is infused in the representation to enhance personalization. Moreover, a modified latent factor model is applied to map users and posts into a shared low-dimensional space while the posts are decomposed into words to handle the data sparsity issue. The consideration of user groups is able to capture individuality to a certain extent and can potentially enrich the sparse data. However, an alternative is discovered in our work that is unconstrained by the user group assumption.

Similarly to our approach, several studies have used neural networks to analyze individualities in sentiment analysis. Targeting product reviews, T. Chen et al. [5] utilize two separate recurrent neural networks to generate user and product representations in order to model the individual differences in assigning rating scores and to obtain the consistencies in receiving rating scores of the same product. A convolutional neural network is used to generate embeddings for the review text. Finally, the representations from the three parties are combined using a traditional machine learning classifier. Another work on product reviews is done by H. Chen et al. [4] who employ a hierarchical network with Long Short-Term Memory (LSTM) on word-level and sentence-level representations. Additionally, an attention mechanism based on user and product information is used on each level after the LSTM layer. By doing that, user preferences and product characteristics are introduced in the network to produce a finer-represented document embeddings. There are similar works that consider individual differences related to sentiment [7,35], but very few have explicitly modeled the evolvement of sentiments of an individual over time. In our work, earlier posted texts are concerned in determining the sentiment of the current text. In addition, we propose a method towards an evaluation of the influence of gaps between texts generated at different time points.

3 Personalized Model with Recurrent Neural Network

In this section, we introduce a basic structure of the *personalized sentiment model*. We explain how it has evolved from the preliminary model, where the effectiveness of considering opinion holders and historical texts is evaluated, to an enhanced version, where the information from the opinion holders and the texts is better represented and learned.

3.1 The Preliminary Model

In [13], we have proposed a personalized model which aims at investigating the effectiveness of including individualities in sentiment analysis. With respect to individualities, the following assumptions were considered:
Assumption I: Different individuals make different lexical choices to express their opinions.
Assumption II: An individual's opinion towards a topic is likely to be consistent within a period of time, and opinions on related topics are potentially influential to each other.
Assumption III: There are connections between an individual's opinion and the public opinion.

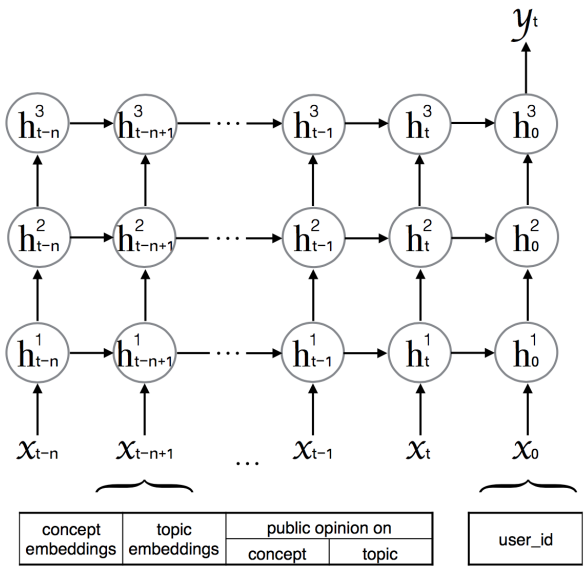

Fig. 1. Personalized sentiment model with a recurrent neural network and two types of neurones at the input layer: The user index (x_0) and the tweet of the user at a specific time point (x_{t*}) [14]. The latter is represented by a concatenation of four components $x_{t*} = [E_{concept}\ E_{topic}\ P_{concept}\ P_{topic}]_*$

To leverage these assumptions, a many-to-one recurrent network with three hidden layers (h^1, h^2 and h^3) is built to preserve and extract related information from historical data (Fig. 1). Each layer contains a number of LSTM cells as defined in [12] without peephole connections. Let $(i_k, f_k, C_k, o_k, h_k)$ denote respectively the input gate, forget gate, cell memory, output gate, and hidden states of the LSTM cell. The update of the cell state and the output of the cell

are then described with the following equations:

$$i_k = \sigma(W_i[x_k, h_{k-1}] + b_i) \tag{1}$$
$$f_k = \sigma(W_f[x_k, h_{k-1}] + b_f) \tag{2}$$
$$C_k = f_k \odot C_{k-1} + i_t \odot \tanh\left(W_C[x_k, h_{k-1}] + b_C\right) \tag{3}$$
$$o_k = \sigma(W_o[x_k, h_{k-1}] + b_o) \tag{4}$$
$$h_k = o_k \odot \tanh\left(C_k\right) \tag{5}$$

where σ denotes the sigmoid activation function. With Equation 1, 2 and 3, the cell k selects new information and discards outdated information to update the cell memory C_k. For the output of the cell, o_k selects information from the current input and the hidden state (Equation 4), and h_k combines the information with the cell state (Equation 5). This memory network is beneficial for understanding implicit or isolated expressions such as *'I have changed my mind about it'*.

Each input sequence consists of two parts: one is the user identifier x_0 of the current tweet, and the other is the representation of the current tweet and a number of past tweets.

User Identifier The use of the user identifier is inspired by [17] who add a language index in the input sequence to enable zero-shot translation in a multilingual neural machine translation system. By adding this identifier in the input, our proposed network is able to learn user-related information and to compare between users. More importantly, the data sparsity issue is resolved since only one model is required.

Tweet Representation Each tweet is represented by a concatenation of four components $x_{t*} = [E_{concept}\ E_{topic}\ P_{concept}\ P_{topic}]_*$, where $E_{concept}$ is the concept embedding of the tweet $t*$, E_{topic} is the topic embedding of the tweet $t*$, $P_{concept}$ is the public opinion on the concepts, and P_{topic} is the public opinion on the topic. Here, the concepts are taken from SenticNet[6] [2] and contain conceptual and affective information of the text. Topics are provided in the used corpus. The embeddings for concepts and topics are learned using a fully connected shallow network similar to Word2Vec [24]. Concept embeddings are the weights at the output layer trained with the target concept at the output layer and its context concepts at the input layer. Topic embeddings are trained by setting the target topic at the output layer and its associated concepts at the input layer. Such embeddings are generated based on the co-occurrences of terms, so that terms with greater similarity are located closer in the vector space. Furthermore, public opinions are Sentic values extracted from the SenticNet, and the values are static.

In Fig. 1, the input sequence X is a matrix of $[x_{t-n}, x_{t-n+1}, ..., x_{t-1}, x_t, x_0]$ where x_t is the current tweet, x_{t-*} are the tweets published before it by the same user x_0, and n is the number of past tweets considered. Zero-padding is

[6] http://sentic.net/, last seen on April 19, 2018

performed before the earliest tweet for users with less than $n + 1$ tweets. The output y_t is the sentiment orientation of the current tweet. Both x_* and y_t are vectors and n is a constant. For training and testing, the tweets are first sorted by the user index, and then by the creation time of the tweets.

The preliminary model is a simplified network that is used for evaluating the effectiveness of introducing the mentioned assumptions in determining sentiment. Although experiments have shown positive results (Section 6.1), there are several aspects that can be modified to improve the performance. First, the input of the network takes two different types of information – the user index and the tweet representation – at different nodes, and the network has to react with the same set of parameters. This setting makes the network harder to train. Second, the representation of a tweet is not sufficient to include necessary information in the text. Negation cues, as signal terms, can invert the polarity of sentiment, hence they should be added in the representation [16]. Moreover, the single topic given for each tweet can be unilateral since multiple entities are mentioned in some cases. Furthermore, the influence of past opinions can be affected by time, i.e. the gap between the tweets of a user can be a reflector of the importance of the past opinions.

3.2 Stacked Neural Network

Stacked networks are popular for tasks that require representations from different levels [4,38]. Here, we consider a tweet-level representation and a user-level representation, and merge the embedding networks in the preliminary model with the recurrent network so that the representations of the tweets are learned automatically through the network by the sentiment label y_t (Fig. 2).

In the input sequence of the stacked network, each tweet x_* is represented by a set of concepts, entities, negation cues and the user identifier. The concepts are from the same knowledge base as the preliminary model, whereas entities are extracted from the text instead of using the single topic so that the relation between the concept and the target can be more flexible. Additionally, explicit negations are included in the input based on a pre-defined list of rules. As a better alternative, the user identifier is placed in the tweet representation instead of occupying an individual node at the input layer of the recurrent network to obtain a consistency in the inputs. There are also a number of tweets with no explicit concepts, entities or negation cues mentioned in the text, and such tweets are represented by the appeared components. In an extreme case, it is also possible that a tweet is simply represented by the user identifier, and historical tweets will play an important role in predicting the sentiment of the current tweet. Public opinions are redundant, because the opinions of majorities can be learned automatically given enough training samples from a sufficient number of users. Meanwhile, the tendency of whether a person's opinions align with the public can be learned directly. Since the representation is concept-based, the order of words appeared in the text does not play a role in the representation. As a result, a single embedding layer is applied to map the terms into a dense, low-dimensional space.

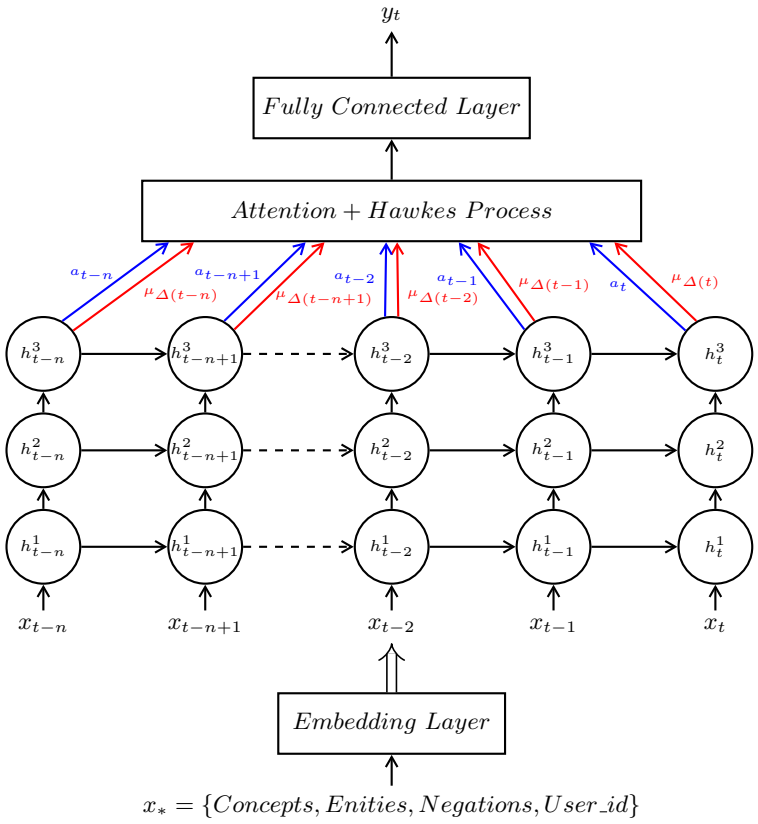

Fig. 2. Stacked personalized sentiment model with a recurrent neural network that is shaped by an attention-based Hawkes process. The input sequence of the network is sets of concepts, entities, negation cues, and the user identifier at different time points

The stacking of networks happens between the generation of the tweet embeddings and the construction of the input sequence for the recurrent neural network. Similarly, a recurrent network with LSTM cells is used in the model. With a consistent formulation of the representation at each input node, the network can be trained efficiently. Again, the input sequences are first sorted by the user identifier and afterwards by the creation time of the text.

4 Attention-based Hawkes Process

As shown in Fig. 2, we employ an attention-based Hawkes process at the output of the recurrent network. This layer models time gaps of different lengths between the publishing dates of the texts. Attention is given to the past tweets that are related to the current tweet content-wise, while the decay of the relation is modeled by the Hawkes process.

4.1 Attention Mechanism

Attention mechanism is widely used in natural language processing [36,38]. In the preliminary model, all the information learned in the network are accumulated at the node that is the closest to the sentiment label (h_0^3), which can be treated as an embedding for all seen tweets in an input sequence. Although LSTM has the ability to preserve information over time, in practice it is still problematic to relate to the node that is far away from the output. LSTM tends to focus more on the nodes that are closer to the output y_t. There are studies that propose to reverse or double the input sequence [39], however attention mechanism can be a better alternative. A traditional attention model is defined as:

$$u_i = \tanh\left(W_t h_i + b_t\right) \tag{6}$$

$$a_i = u_i^T w_s \tag{7}$$

$$\lambda_i = softmax(a_i) h_i \tag{8}$$

$$v = \sum_i \lambda_i \tag{9}$$

where $softmax(x_i) = e^{x_i} / \sum_j e^{x_j}$. λ_i is the attention-shaped output at the specific time t_i with the same dimension as h_i, and v sums up the output at each t_i dimension-wise and contains all the information from different time points of a given sequence. The context vector w_s can be randomly initialized and jointly learned with other weights in the network during the training phase.

4.2 Hawkes Process

Hawkes Process is a one-dimensional *self-exciting* point process. A process is said to be *self-exciting* if an 'arrival' causes the conditional intensity function to increase (Fig. 3) [20]. Hawkes Process models a sequence of arrivals of events

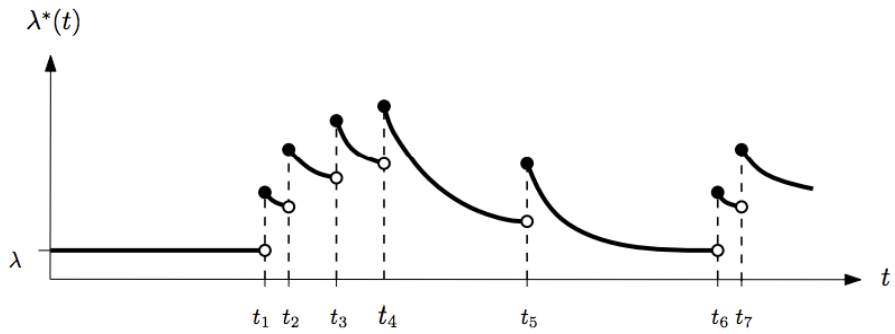

Fig. 3. An example conditional intensity function for a self-exciting process [20]

over time, and each arrival excites the process and increases the possibility of a future arrival in a period of time.

As in [20], the conditional intensity function of Hawkes process is:

$$\lambda^*(t) = \lambda + \sum_{t_i<t} \mu(t-t_i) = \lambda + \sum_{t_i<t} \alpha e^{-\beta(t-t_i)} \tag{10}$$

where λ is the background intensity and should always be a positive value, and $\mu(\cdot)$ is the excitation function. Here, we use exponential decay as the excitation function because it is a common choice for many tasks. The value of α and β are positive constants where α describes how much each arrival lifts the intensity of the system and β describes how fast the influence of an arrival decays.

Hawkes Process in Sentiment Traditional Hawkes process models the influence of the past events on the future event which assumes that these are the same events (or similar in nature). For that reason, the value of α is constant, i.e. each arrival affects the system in the same way. As a heuristic study, we see an opinion as an 'event' that positively influences future opinions, and such influence decreases with time. However, people's opinion may be affected by their past opinions when there are some connections between the targets (topics or entities) of the opinions. In our setting, the preceding opinion can be irrelevant to the current one, in which case the influence from the preceding opinion should not be boosted.

4.3 Hawkes Process with Attention

In order to apply Hawkes process on opinions, we shape the effect of past opinions with attention mechanism. The exponential decay factor is added at each time point based on the attention output, so that the historical text which contains more relevant content affects the current opinion more intensively than those

with less relevant content. The following two equations describe the shaped output with the attention mechanism and the Hawkes process:

$$v'(t) = v + \varepsilon \sum_{i:\Delta t_i > 0} \lambda'_i e^{-\beta \Delta t_i} \tag{11}$$

$$= \sum_{i:\Delta t_i \geq 0} (\lambda_i + \varepsilon \lambda'_i e^{-\beta \Delta t_i}) \tag{12}$$

where

$$\lambda'_i = \begin{cases} \lambda_i & \text{if } \lambda_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

In Equation 11, the first element v is the background intensity which acts as a base factor and describes the content in the text throughout the time. ε represents a decay impact factor and balances the importance of adding the Hawkes process in the output, and a value of zero indicates that the information decay does not play any part in the decision making. Theoretically, there should be no upper bound for the value of ε, however it is illogical to take a value that is much greater than 1 ($\varepsilon \not\gg 1$). $\Delta t_i = t - t_i$ is the time difference between the current time t and the time t_i. β is the decay rate for the time difference, and the value of β varies from task to task. Note that ε and β are constants, and their values must be chosen priorly. Here, α in Equation 10 is replaced by λ'_i so that the effect of an arrival is not constant anymore. λ'_i is a rectifier which takes $max(0, \lambda_i)$. With the rectifier, the effect remains non-negative and only relevant events (targets) are considered. λ_i is calculated according to Equation 8. Given Equation 9, Equation 12 is deduced so that at each time point in the past, the attention output is boosted by the process factor when it is a positive value. The final output $v'(t)$ is the sum of the modified attention outputs over time for the current tweet created at time t. Additionally, a fully connected layer is added after applying the Hawkes process in order to regularize the output for the network to train.

5 Implementation

In this section, we present the implementation of both the preliminary model and the enhanced model. The former is referred to as 'P-model' (**P**reliminary model) throughout this text and the latter as 'AHP model' (**A**ttention-based **H**awkes **P**rocess model). The models are evaluated using different datasets which are labeled in distinct ways.

5.1 Technical Setup

The concepts used as a part of the text representation are taken from SenticNet and are 50,000 in total. The implementation of both models is conducted using

Keras[7] with the Tensorflow[8] back-end. In the P-model, topics are embedded with 32-dimensional vectors and concepts are embedded with 128-dimensional vectors. In contrast, all the terms are embedded with 128-dimensional vectors in the embedding layer of the AHP model. The first two layers in the recurrent network are equipped with 64 LSTM cells each while the last layer has 32 cells. Moreover, dropout is used for both models to prevent overfitting [33].

5.2 Datasets

There are a number of datasets for Twitter sentiment analysis, and they differ mainly by the annotation technique. The labeling of sentiment can be performed in two ways: manually and automatically. Manual labeling is mostly done by the 'Wisdom of Crowd' [34], which is time consuming and only applicable for a small amount of data. For the text from social platforms, an automatic labeling mechanism is possible by categorizing the emoticons appeared in the text. The emoticons are employed in a distant supervision approach. As discussed in [9], such labeling can be quite noisy but effective. Furthermore, these two methods can be seen as annotations from different standpoints [31], which is a separate research direction that may be focus of a separate study. For a personalized sentiment analysis, manually labeled data do not contain sufficient user information to explore the individualities. For that reason, we use manually labeled data for the P-model in our experiments so that an universal comparison can be made to evaluate the effectiveness of including user data, while automatically labeled data are used for the AHP model so that a more comprehensive evaluation on the personalization can be provided.

Table 1. Statistics of the datasets used for the P-model and the AHP model

Model	Dataset	Polarity				#	User Frequency	
		Pos.	Neg.	Neu.	Total	Topic/Entity	#Tweets p.User	#User
P-	Sanders	424	474	2,008	2,906	4	> 5	51
	SemEval	6,758	1,858	8,330	16,946	100	$>= 2$	971
AHP	Sentiment140	79,009	42,991	—	122,000	311	$>= 20$	2,369

The statistics of the datasets used for the models is shown in Table 1. Sanders Twitter Sentiment Corpus[9] and the development set of SemEval-2017 Task 4-C Corpus[10] are manually labeled datasets and are used for the evaluation of the P-model. For the SemEval corpus, germane labels are merged into three classes (positive, negative and neutral) in order to combine the corpus with the Sanders corpus. These two datasets are combined since there are no sufficient

[7] https://keras.io/, last seen on April 19, 2018
[8] https://www.tensorflow.org/, last seen on April 19, 2018
[9] http://www.sananalytics.com/lab/twitter-sentiment/, last seen on April 19, 2018
[10] http://alt.qcri.org/semeval2017/task4/, last seen on April 19, 2018

frequent users (only 51 users have tweeted more than 5 times). It is also to show the independency of the topic-concept relation between different datasets. Sentiment140[11] is automatically labeled with two classes (positive and negative) and is used in the AHP model. Originally, Sentiment140 contains 1,600,000 training tweets, however we extract tweets published by users who have tweeted at least 20 times before a pre-defined date so that only frequent users are considered in this model. The extracted subset contains 122,000 tweets in total. As explained in Section 3.2, entities are used instead of topics which results in 15,305 entities extracted from the text.

Furthermore, each dataset is split into a training set, a validation set and a test set for training and evaluation. The original test sets from the mentioned corpora are not suitable for our experiments because for the P-model, the topic-opinion relations are to be examined while the provided test set contains only unseen topics; for the AHP model, the provided test set contains only unseen users which is unable to verify the user preferences learned by the network.

5.3 Baselines

Sanders + SemEval To evaluate the effectiveness of introducing user information in sentiment analysis, we use the original, manually labeled datasets and compare the P-model with five baselines. The first one is the Sentic values which are sentiment scores between -1 (extreme negativity) and 1 (extreme positivity) from SenticNet. For each tweet, the Sentic values are combined, and afterwards the result and the number of concepts appeared in the tweet are fed to a shallow fully connected network for training.

To compare the P-model with the traditional machine learning technique, we choose to apply the Support Vector Machine (SVM) which is a prominent method in this field. Two SVM classifiers are built that one is trained with the concepts and the topic in the text (named Generalized SVM) while the other one is trained with the same features of the Generalized SVM together with the user index and public opinions (named Personalized SVM). Implemented with scikit-learn [27], the radial basis function kernel and the parameters $C = 0.01$ and $\gamma = 1/N\_features$ are set by 10-fold cross-validation.

Convolutional neural network (CNN) as another widely used neural network structure, has been shown to provide competitive results on several sentence classification tasks compared to various approaches. We use a network similar to the simple CNN proposed by Kim [18] with concepts as inputs instead of words that are used in the original work. Such a network highlights the relations between adjacent elements, however it may be difficult to explore since the order of concepts does not necessarily convey useful information.

The last one is a generalized recurrent neural network (named Generalized RNN) which uses the same network with the P-model but without the user index and the public opinions. As a result, $x_{t*} = [E_{concept}\ E_{topic}]_*$ is set at each input node, and the input sequence is ordered by the creation time of the tweets.

[11] http://help.sentiment140.com/for-students/, last seen on April 19, 2018

Sentiment140 With the larger, automatically labeled dataset, we are able to train the AHP model with frequent users. We compare the performance of the following models:

1. A model with the output layer applied directly after the recurrent network (named Basic model);
2. A model with the output layer applied after adding the attention layer to the recurrent network but without Hawkes process (named Attention model);
3. The AHP model as in Fig. 2.

The same input sequences are given to the AHP model and its substitutions, thus the effect of applying the attention and Hawkes process layer can be evaluated.

6 Results and Discussion

In this section, we report the evaluation results of the P-model compared with five baselines, and discover the effects of the key factors for personalization. The performance of the AHP model will be discussed as well.

6.1 Evaluation of the P-model

We compare the P-model with the baselines introduced in Section 5.3. Accuracy is used as the primary evaluation metric, and macro-averaged recall is used to demonstrate the balance of the prediction over classes which is more intuitive than displaying the recall value for each class. Note that we do not compare the performance of the P-model with the reported results of SemEval because different test data are used for the evaluation, as explained in Section 5.2. The comparison is shown in Table 2 as in our earlier publication [13].

Table 2. Comparison of the performance between the P-model and the chosen baselines

Dataset	Model	Accuracy	Avg. Recall
Sanders + SemEval	Sentic	0.3769	0.4408
	Generalized SVM	0.6113	0.5847
	Personalized SVM	0.6147	0.5862
	CNN	0.5481	0.5360
	Generalized RNN	0.6382	0.6587
	P-Model	**0.6569**	**0.6859**

In the same way as for the P-model, concept-based representation is used for the baseline models. The Sentic scores reflect an interpretation of the concepts from a general point of view; they are used as public opinions in the P-model. Without integrating any additional information, it performs the worst implying that no implicit knowledge is captured from the text. Reasonable results are

achieved by the generalized and personalized SVM models. The personalized model offers a slightly better performance given additional user-related features, however the improvement is not significant enough to serve the purpose of modeling individuality. The CNN model, which follows the work by Kim [18], makes use of the dependencies between contiguous terms. Such dependencies are rather vague in the concepts, because the words are already shuffled while extracting concepts from the text. Thus, the performance of the CNN model is comparatively worse than the SVM- and RNN-based models.

The generalized RNN captures the trend in public opinions by comparing the concepts and the associated topic from different time points in the past. This model outperforms the personalized SVM, which reveals the significance of considering the dependencies between tweets. By adding the user-related information in the P-model, the performance is further improved with $p < 0.05$ for the t-test. The improvement indicates that individuality is a crucial factor in analyzing sentiment and is able to positively influence the prediction.

6.2 Key Factors for Personalization

To explore the influential factors in the personalized sentiment model, we conduct experiments from three different angles. More evidence is shown for the effectiveness of including user diversity in the model.

Topic-Opinion Relation To evaluate the effect of considering topic-opinion relations, we exclude the topic-related components from the input sequence and set $x_{t*} = [E_{concept}\ P_{concept}]_*$ for the input nodes before the user identifier. The resulting model gives an accuracy of 0.5536 and an average recall of 0.5429, which is significantly worse than the P-model (Table 2). The gap between the results reveals the advantage of associating sentiment with topics and adding the components E_{topic} and P_{topic} in the system.

User Frequency As shown in Table 1, there are 714 users who have tweeted twice and 51 users who have tweeted more than 5 times. In fact, most users in the combined dataset have only one tweet. While targeting users with different frequencies, the P-model achieves an accuracy of 0.6282 for the users who have tweeted twice, and the accuracy rises to 0.7425 for the users who have more than 5 tweets. As expected, the model is able to provide a better prediction for more frequent users.

Length of the History An experiment is conducted with different numbers of past tweets added in the input sequence. Results are shown in Table 3 as reported in our previous work [13]. The performance is poor while considering one past tweet, because the possibility of having meaningful relations between two consecutive tweets is relatively low. By taking more past tweets into account, the performance of the model keeps improving. Note that when associating 10

Table 3. Performance of the P-model considering different numbers of past tweets in the input sequence

Number of Past Tweets	Accuracy	Avg. Recall
1	0.5680	0.5481
5	0.6216	0.6346
10	0.6305	0.6671
15	0.6461	0.6688
20	0.6569	0.6859

past tweets, the performance is competitive with the generalized RNN model, which is associated with 20 past tweets as reported in Table 2. This shows the importance of integrating user information and historical text in sentiment analysis.

6.3 Evaluation of the AHP model

Based on the results from the last two sections, we enhance the model and verify its performance by experimenting with a larger dataset and comparing with its substitutions. For the AHP model, accuracy is used as the primary evaluation metric as well, and F1-score is calculated for each class to demonstrate the balance of the prediction since only two classes are concerned.

Table 4. Comparison of the performance between the AHP model and its substitutions

Dataset	Model	Accuracy	Pos. F1	Neg. F1
Sentiment140	Basic Model	0.7287	0.7344	0.7223
	Attention Model	0.7491	0.7464	0.7516
	AHP Model	**0.7596**	**0.7534**	**0.7652**

In Table 4, we can see improvements after adding the attention layer and after shaping the attention outputs with Hawkes process. The Attention model compensates the loss of focus of the Basic model with distant nodes, and the AHP model tightens or loosens the relations of the nodes according to the gaps between them. To implement the AHP model, the values of ε and β in Equation 11 must be set beforehand. We take $\varepsilon = 0.7$ and $\beta = 0.01$ which give the best performance in the experiment, and the corresponding results are shown in Table 4.

6.4 Key Factors for Information Decay

Decay Impact Factor $\boldsymbol{\varepsilon}$ and Decay Rate $\boldsymbol{\beta}$ The values for ε and β are found experimentally by grid search given an empirical range. The results are illustrated in Fig 4. Because ε and β interact with each other on the rectified attention output, it is difficult to find a significant trend between these two values

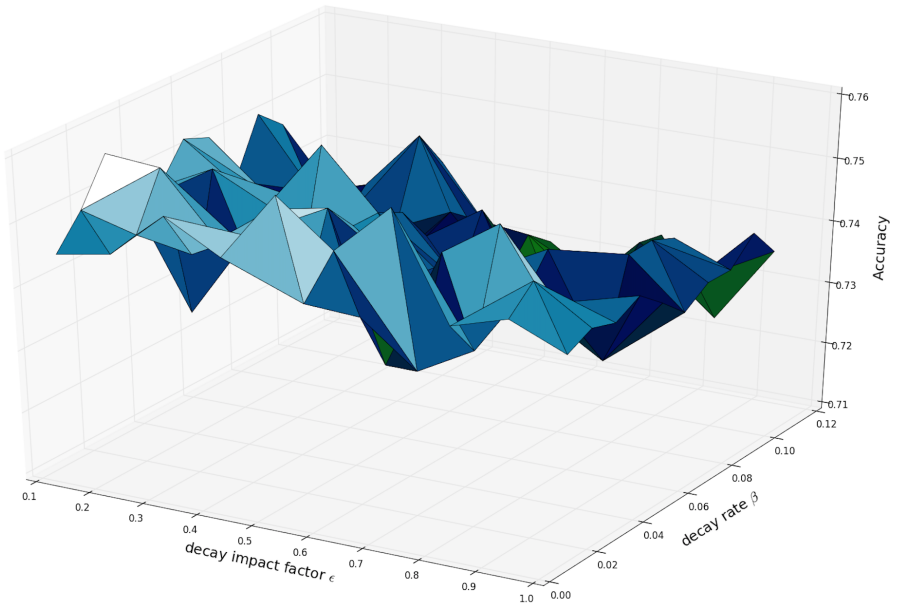

Fig. 4. A grid search on the parameters ε and β for the attention-based Hawkes process

and the accuracy of the prediction. However, there is a slight tendency to offer better results with a smaller β. Comparing to the best results in Table 4 which are given by $\varepsilon = 0.7$ and $\beta = 0.01$, the worst result in the set range is 0.7270 (accuracy) given by $\varepsilon = 0.9$ and $\beta = 0.1$ which is worse than the accuracy of the Basic model. Such a performance shows the importance of setting suitable parameters in the excitation function.

Excitation Function The advantages of using exponential kernels in Hawkes process descend from a Markov property as explained in [1], which motivates the use of this excitation function in our model. The property can be extended to the case where the value of β is non-constant, which is useful in assigning different decay rates for different users or for different levels of intensity (attention outputs). The effect of using these variants or applying other excitation functions for Hawkes process is to be discovered.

7 Conclusion and Future Work

In this article, we focused on developing a *personalized sentiment model* that is able to capture users' individualities in expressing sentiment on social platforms. To evaluate the effectiveness of including user information in sentiment analysis, we built a preliminary model based on three assumptions and conducted a series of experiments with Twitter data. The assumptions reflect the individuality

from different aspects and the model is designed accordingly. We use concepts appeared in the text to represent people's lexical choices; we add the topic in the text representation to include the topic-opinion relations; public opinions are used in the input in order to find connections between individual and public opinions. A simple recurrent neural network is built for this task which is able to relate the information of the current tweet with historical tweets. The issue of data sparsity is handled by adding a user identifier in each input sequence. The preliminary model is evaluated with a combined, manually labeled Twitter dataset, and the effectiveness of introducing user data in the model is verified by comparing to five baseline models. Moreover, the key factors of a personalized sentiment model are discovered. We believe that the topic-opinion relation, the user frequency and the number of the historical tweets considered in the network are the major factors that influence the performance of the model.

Given the positive results of the preliminary model, we proposed an enhanced model that focuses on frequent users. The enhanced model is a stacked network that takes concepts, entities, negation cues and user identifier to represent each tweet and applies an embedding layer to generate inputs for the recurrent network. Furthermore, attention mechanism is used on the output of the recurrent network which helps the network to concentrate on related and distant tweets. To consider the different gaps between the tweets, we introduce a novel approach that is to shape the attention output with Hawkes process. By using this approach, the attention on the related tweets is boosted and the effect fades by a certain decay rate on the distance between these tweets and the current tweet. Thus, a decay of information with time can be modeled in the network. This model is tested on a larger dataset with users who have tweeted at least 20 times before a pre-defined timestamp, and improvements are shown after adding the attention layer with Hawkes process.

The results from these two models bring us significant meanings of applying a personalized sentiment model. We have learned that the individualities have substantial influence on sentiment analysis and can be easily captured by models like the ones we have proposed in this article. Moreover, traditional recurrent neural networks neglect the effect of various gaps between the nodes which can be an important factor in many tasks. As we have shown, the Hawkes process can be combined with recurrent networks to compensate such lack of information, and the effect of using different variants of Hawkes process has yet to explore.

The improvements of the preliminary model and the enhanced model have opened up new opportunities for future research. To generalize the use of the proposed models, we can test the performance by evaluating with finer-labeled sentiments or emotions. It is also possible to use these models on existing sentiment models that do not concern user information in the prediction in order to enhance the performance. As a heuristic research, the attention mechanism can be combined with the Hawkes process in different ways. For instance, Cao et al. [3] proposed an approach of non-parametric time decay effect, which takes different time intervals and learns discrete variables for the intervals as the decay effect μ. As a result, no pre-defined decay functions are needed for the modeling,

and the effect can be flexible based on different time intervals. Such a technique can also be beneficial for our task, and the decay effect and the attention model can be applied on the output of the recurrent network separately. As an extension on the field of application, the personalized model can be used in an artificial companion that is adapted under a multi-user scenario to improve communication experience by offering user-tailored responses.

References

1. Bacry, E., Mastromatteo, I., Muzy, J.F.: Hawkes processes in finance. Market Microstructure and Liquidity **1**(01), 1550005 (2015)
2. Cambria, E., Poria, S., Bajpai, R., Schuller, B.W.: SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In: COLING. pp. 2666–2677 (2016)
3. Cao, Q., Shen, H., Cen, K., Ouyang, W., Cheng, X.: DeepHawkes: Bridging the gap between prediction and understanding of information cascades. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1149–1158. ACM (2017)
4. Chen, H., Sun, M., Tu, C., Lin, Y., Liu, Z.: Neural sentiment classification with user and product attention. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1650–1659 (2016)
5. Chen, T., Xu, R., He, Y., Xia, Y., Wang, X.: Learning user and product distributed representations using a sequence model for sentiment analysis. IEEE Computational Intelligence Magazine **11**(3), 34–44 (2016)
6. Cheng, X., Xu, F.: Fine-grained opinion topic and polarity identification. In: LREC. pp. 2710–2714 (2008)
7. Dou, Z.Y.: 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. pp. 521–526 (2017)
8. Gilbert, C.H.E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf (2014)
9. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford **1**(12) (2009)
10. Gong, L., Al Boni, M., Wang, H.: 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, pp. 855–865 (2016)
11. Gong, L., Haines, B., Wang, H.: Clustered model adaption for personalized sentiment analysis. In: Proceedings of the 26th International Conference on World Wide Web. pp. 937–946. International World Wide Web Conferences Steering Committee (2017)
12. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. pp. 6645–6649. IEEE (2013)
13. Guo, S., Höhn, S., Xu, F., Schommer, C.: PERSEUS: A personalization framework for sentiment categorization with recurrent neural network. In: International Conference on Agents and Artificial Intelligence, Funchal 16-18 January 2018. p. 9 (2018)

14. Guo, S., Schommer, C.: Embedding of the personalized sentiment engine PERSEUS in an artificial companion. In: International Conference on Companion Technology, Ulm 11-13 September 2017. IEEE (2017)
15. Hawkes, A.G.: Spectra of some self-exciting and mutually exciting point processes. Biometrika **58**(1), 83–90 (1971)
16. Jia, L., Yu, C., Meng, W.: The effect of negation on sentiment analysis and retrieval effectiveness. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 1827–1830. ACM (2009)
17. Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al.: Google's multilingual neural machine translation system: Enabling zero-shot translation. arXiv preprint arXiv:1611.04558 (2016)
18. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
19. Kobayashi, R., Lambiotte, R.: TiDeH: Time-dependent Hawkes process for predicting retweet dynamics. In: ICWSM. pp. 191–200 (2016)
20. Laub, P.J., Taimre, T., Pollett, P.K.: Hawkes processes. arXiv preprint arXiv:1507.02822 (2015)
21. Liu, B.: Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press (2015)
22. Markovikj, D., Gievska, S., Kosinski, M., Stillwell, D.: Mining Facebook data for predictive personality modeling. In: Proceedings of the 7th international AAAI conference on Weblogs and Social Media (ICWSM 2013), Boston, MA, USA. pp. 23–26 (2013)
23. Meena, A., Prabhakar, T.: Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In: European Conference on Information Retrieval. pp. 573–580. Springer (2007)
24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
25. Mohler, G.O., Short, M.B., Brantingham, P.J., Schoenberg, F.P., Tita, G.E.: Self-exciting point process modeling of crime. Journal of the American Statistical Association **106**(493), 100–108 (2011)
26. Ogata, Y.: Space-time point-process models for earthquake occurrences. Annals of the Institute of Statistical Mathematics **50**(2), 379–402 (1998)
27. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**(Oct), 2825–2830 (2011)
28. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. Proceedings of SemEval pp. 27–35 (2014)
29. Reiter, E., Sripada, S.: Human variation and lexical choice. Computational Linguistics **28**(4), 545–553 (2002)
30. Saif, H., He, Y., Fernandez, M., Alani, H.: Contextual semantics for sentiment analysis of Twitter. Information Processing & Management **52**(1), 5–19 (2016)
31. Schommer, C., Kampas, D., Bersan, R.: A prospect on how to find the polarity of a financial news by keeping an objective standpoint. Proceedings ICAART 2013 (2013)
32. Song, K., Feng, S., Gao, W., Wang, D., Yu, G., Wong, K.F.: Personalized sentiment classification based on latent individuality of microblog users. In: IJCAI. pp. 2277–2283 (2015)

33. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
34. Surowiecki, J.: The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business. Economies, Societies and Nations **296** (2004)
35. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 1014–1023 (2015)
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 6000–6010 (2017)
37. Wiebe, J., Wilson, T., Bell, M.: Identifying collocations for recognizing opinions. In: Proceedings of the ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation. pp. 24–31 (2001)
38. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1480–1489 (2016)
39. Zaremba, W., Sutskever, I.: Learning to execute. arXiv preprint arXiv:1410.4615 (2014)
40. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: Seismic: A self-exciting point process model for predicting tweet popularity. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1513–1522. ACM (2015)