PERSEUS: A Personalization Framework for Sentiment Categorization with Recurrent Neural Network

Siwen Guo[1], Sviatlana Höhn[2], Feiyu Xu[3] and Christoph Schommer[1]
[1]*ILIAS Research Lab, CSC, University of Luxembourg, Esch-sur-Alzette, Luxembourg*
[2]*AI Minds, Vianden, Luxembourg*
[3]*Lenovo, Beijing, China*

Keywords: Sentiment Categorization, Opinion Mining, Personalized Memories, Neural Networks.

Abstract: This paper introduces the personalization framework PERSEUS in order to investigate the impact of individuality in sentiment categorization by looking into the past. The existence of diversity between individuals and certain consistency in each individual is the cornerstone of the framework. We focus on relations between documents for user-sensitive predictions. Individual's lexical choices act as indicators for individuality, thus we use a concept-based system which utilizes neural networks to embed concepts and associated topics in text. Furthermore, a recurrent neural network is used to memorize the history of user's opinions, to discover user-topic dependence, and to detect implicit relations between users. PERSEUS also offers a solution for data sparsity. At the first stage, we show the benefit of inquiring a user-specified system. Improvements in performance experimented on a combined Twitter dataset are shown over generalized models. PERSEUS can be used in addition to such generalized systems to enhance the understanding of user's opinions.

1 INTRODUCTION

Sentiment analysis is the task to recognize subjectiveness in text and determine a polarity for a given subject (Nakov et al., 2016). Most existing methods treat different sentiment holders as the same, and generate a sentiment score for each document (Wiebe et al., 2001), sentence (Meena and Prabhakar, 2007), or aspect of an entity (Cheng and Xu, 2008; Pontiki et al., 2014). However, people are diverse while consistent to a degree. They have various lexical choices in expressing sentiments which is caused by many factors, such as preference organization, linguistic and cultural background, expertise and experience. At the same time, some consistencies can be observed in an individual's opinion towards a topic, as well as in the relations of possessing an opinion between an individual and the public. With this background, the objective of this research is investigating the effectiveness of considering individuality in sentiment analysis.

To examine the influence of such traits in sentiment categorization, we propose PERSEUS – a personalization framework that considers the diversity and individual consistency under the following assumptions which are deduced from existing studies and observations (Reiter and Sripada, 2002; Janis and Field, 1956; Nowak et al., 1990):

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.

PERSEUS applies the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), which is one of the recurrent neural network (RNN) architectures, to leverage these assumptions. The potential to fulfill this goal is based on LSTM's ability of carrying valuable information over time regulated by a set of structured gates. LSTM is widely used in natural language processing (Sundermeyer et al., 2012; Sutskever et al., 2014). However, it is mostly used to analyze relations between words inside documents or sentences (Teng et al., 2016; Wang et al., 2016). In contrast, our proposition implies that the learning process needs to take cross-document relations into account.

Although the evaluation of the framework is done with Twitter data, we expect PERSEUS to be data-independent and adaptable to other corpora of similar

Guo, S., Höhn, S., Xu, F. and Schommer, C.
PERSEUS: A Personalization Framework for Sentiment Categorization with Recurrent Neural Network.
DOI: 10.5220/0006584100940102
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 94-102
ISBN: 978-989-758-275-2

user-oriented structure. In this work, a document corresponds to a tweet at a specific time point. We use the term *intra-document relation* to describe the semantic dependencies within a document, and *cross-document relation* to describe the dependencies between documents of the same user. To deal with the issue of data sparsity that comes with user-specific data, we take inspiration from Johnson et al. (2016) where an additional token is added to the input sequence to indicate required target language for multilingual neural machine translation. We add an individual neuron with the user index to the input so that the individuality of a certain user can be captured, while at the same time, the relations between users can be learned automatically. To the best of our knowledge, PERSEUS is the first personalization framework that aims at discovering long term dependencies between user behavior and public behavior associated with topics.

This paper is organized as follows: Section 2 contains discussions of related work. In Section 3, we introduce the structure of PERSEUS and approaches used in the framework. Section 4 presents the set up of our experiments and the datasets used for evaluation. Section 5 compares the proposed framework with five baselines and reports evaluation results. Finally, Section 6 concludes our work and gives an outlook of future research.

2 RELATED WORK

While the majority of academic publications do not take individual sentiment holders into account (Saif et al., 2012; Pak and Paroubek, 2010; Pang and Lee, 2005), there are a small number of studies that consider user diversities in sentiment analysis. Some academic publications set similar objectives and include such diversities in the model to improve *intra-document relation* for document-level sentiment classification, but do not involve an explicit study for *cross-document relation* (Chen et al., 2016a; Tang et al., 2015).

Gong et al. (2016) present a framework where a global model captures 'social norms', and personalized models are adapted from the global model via a series of linear transformations. The homogeneity is achieved by applying the global model, while the heterogeneity is achieved by applying the personalized models. However, the correlation between users and topics (e.g. restaurants, products) is not explicitly modelled in this structure.

Chen et al. (2016b) focus on product reviews, and use recurrent neural networks to generate user and product embeddings, which are then incorporated using a traditional machine learning classifier. In addition, temporal relations of reviews are considered. However, the embeddings of users and products are trained in parallel in the sequence modelling, and users are not modelled specifically. In this sense, Chen et al. (2016b) propose an approach which is less user-oriented than PERSEUS.

Song et al. (2015) utilize a modified latent factor model that maps users and posts into a shared latent factor space to analyze individuality. Social network user's following information is also studied to enhance representation of users by assuming that followers and followees may share common interests. Comparing to the existing works, the major difference in PERSEUS is that we consider user's opinions in the past at a cross-document level and associate the opinions with topics, while user-public relations are also included.

3 THE PERSONALIZATION FRAMEWORK

The personalization framework surveys and leverages *cross-document relation* under the assumptions introduced in Section 1.

3.1 Concept Representation

The level of granularity in text representation plays an important role in understanding the text. There are works based on characters (Dos Santos and Gatti, 2014), bag-of-words (Whitelaw et al., 2005), *n*-grams (Bespalov et al., 2011), or concepts (Cambria and Hussain, 2015). As an intra-document representation, we chose to use the concepts from SenticNet[1] which allow capturing implicit meaning of text using web ontologies or semantic networks. The concepts contain conceptual and affective information. For instance, *'It is a nice day to take a walk on the beach'* contains concepts *nice, nice day, take, take walk,* and *walk beach*. At the first stage of PERSEUS, we simplify the text representation to concentrate on the influence of additional user-related information for the cross-document study.

To deal with the sparsity problem in representing words or phrases, embedding methods are usually a good choice. Similar to Word2Vec (Mikolov et al., 2013) which generates word embeddings based on the co-occurrence of the words, we use concepts as the granular base, and place concepts at the input and output of a shallow, fully connected network. Since

[1] http://sentic.net/

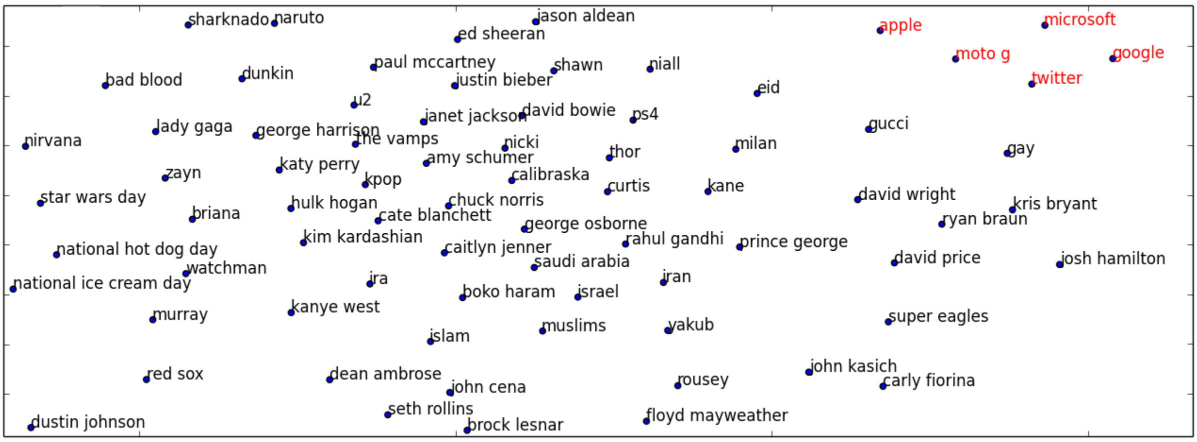

Figure 1: A fragment of a t-SNE projection of the topic embeddings trained on the combined corpora (Section 4.2). Topics with greater similarity (e.g. terms highlighted with red color) are located closer to each other.

posts from social networks are usually short messages with small numbers of concepts and the order of the concepts contains no extra information, a target concept is fed to the output layer and its context in a post placed at the input layer as one training sample. Furthermore, the weights between the hidden layer and the output layer are taken as the embeddings of the concepts. The learned embeddings have the trait that similar concepts are located closer to each other in a high dimensional space. Another way of creating representation space can be found in academic publications on sentiment analysis, which is to group words by their sentiment orientation such as AffectiveSpace (Cambria et al., 2015) and SSWE (Tang et al., 2014). However, an objective representation is much more desired considering the difference between the perspectives of an individual and the public. Therefore, we use the embeddings based on semantic relations instead of sentiment relations.

3.2 Topic Representation

Given the relationship between opinions and topics introduced in Assumption II, we create embeddings for appearing topics. Similar to the concept representation described in Section 3.1, we construct a shallow network with topic as target and presenting concepts as context to find embeddings for topics. The network is built under the assumption that the more a concept and a topic appeared together, the more descriptive the concept is towards the topic. Alternatively, the networks for learning embeddings of concepts and topics can be merged for simplicity. Figure 1 illustrates a fragment of a t-SNE projection of the topic embeddings. Related topics e.g. 'google', 'microsoft', 'twitter', 'apple', and 'moto g' are located close to each other (upper right corner).

3.3 Structure of Input Sequence

As shown in Figure 2, the input sequence of the recurrent neural network consists of two parts. The first part is the identifier of the user who published the tweet. Instead of building a model for each user, a user index x_0 is added at the end of the input sequence and is encoded as a one-hot vector. This enables the network to learn user related information and to compare different users. For users with only one tweet, we give them an identical index because there is no historical relations that can be learned for such users. In this way, these users are considered as one user that acts aligned with the public with fluctuations. This solution also saves the space for storing the user index for these users. In the situation that PERSEUS is used upon another sentiment model, these users can be excluded until there are at least two tweets from the same user. For users with more than one tweet, their sentiments towards different topics are learned individually. This part is required to examine the effect of using Assumption I.

The second part of the input sequence corresponds to the current and the past tweets of a user, and each tweet contains four components: Concept embeddings of the tweet $E_{concept}$, topic embeddings of the tweet E_{topic}, public opinion on the concepts $P_{concept}$, and public opinion on the topic P_{topic}. In the current version of PERSEUS, public opinions are pre-defined and extracted from an external source as described in 4.1. Concept and topic embeddings are used to introduce Assumption II to the network. Assumption III is included by applying the components of public opinions $P_{concept}$ and P_{topic}.

In practice, the required dimensions of the two parts can be of different lengths. To keep a consistent length for each input node, either more than one node

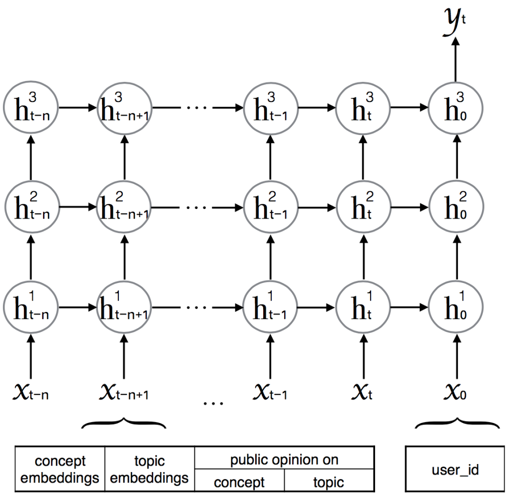

Figure 2: Personalized recurrent neural network with two types of neurones at the input layer: the user index (x_0) and the tweet of the user at a specific time point (x_{t*}) (Guo and Schommer, 2017). The latter is represented by a concatenation of four components. A detailed explanation can be found in the text.

is allocated to the user index or padding is performed for the second part (the tweets). The latter is used in our experiments to enhance the impact of tweets.

3.4 Personalized Recurrent Network

The personalized recurrent neural network is the central of the PERSEUS architecture. It accomplishes the goal of capturing individuality and understanding the user's perspectives. As shown in Figure 2, it has a many-to-one structure, and is composed of three hidden layers (h^1, h^2, and h^3). Each of the hidden layers contains a number of long short-term memory cells which help to preserve and extract valuable information from temporal / sequential data.

To utilize this network, the tweets are first sorted by the user index, and then the creation time of the tweets. Thus, the input sequence X is a matrix of $[x_{t-n}, x_{t-n+1}, \ldots, x_{t-1}, x_t, x_0]$ 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. Note that in current version of PERSEUS, the different gaps between two successive posts x_{t-*} and x_{t-*-1} are not explicitly considered. For the user with more than one but less than $n+1$ tweets, a number of vectors with zeros are padded before the earliest tweet of the user. The output y_t is the sentiment orientation of the current tweet, which can be positive, negative, or neutral. Both x_* and y_t are vectors, and n is a constant number. The LSTM cell used in this architecture follows Graves et al. (2013), however without using peephole connections. As reported in Greff et al. (2016), there is no significant difference in the performance using the peephole connections or other tested modifications.

Let $(i_k, f_k, C_k, o_k, h_k)$ denote respectively the input gate, forget fate, cell memory, output gate, and hidden states of the LSTM cell. The update of the cell state is then described with the following equations:

$$i_k = \sigma(W_i[x_k, h_{k-1}] + b_i) \quad (1)$$

$$f_k = \sigma(W_f[x_k, h_{k-1}] + b_f) \quad (2)$$

$$C_k = f_k \odot C_{k-1} + i_t \odot \tanh(W_C[x_k, h_{k-1}] + b_C) \quad (3)$$

where σ denotes the sigmoid activation function, $k = 0 \rightarrow x_0 = user\_id$ for the input node at the end of the sequence, $k = t$ for the previous input node indicating the current tweet, and $k = t - *$ for other input nodes corresponding to the historical tweets. With the help of the gates i_k and f_k, the cell k selects new information and discards outdated information to update the cell memory C_k. For the output of the cell,

$$o_k = \sigma(W_o[x_k, h_{k-1}] + b_o) \quad (4)$$

$$h_k = o_k \odot \tanh(C_k) \quad (5)$$

where o_k selects information from the current input and the hidden state, and h_k combines the information with the cell state. Moreover, $x_* = [E_{concept}\ E_{topic}\ P_{concept}\ P_{topic}]_*$ is set for $* \neq 0$ as introduced in Section 3.3. Such concatenation of components has been shown effective by Ghosh et al. (2016).

With this design, the network is able to recognize a user index from the input sequence so that the drifting distance between user opinions and public opinions can be learned by accessing information from the past. This approach offers a better alternative for implicit or isolated expressions. For instance, the tweet *'This totally changes my mind about Apple products.'* contains unclear sentiment orientation that the expressed sentiment can only be determined by knowing the past opinion of the user about *'Apple products'*. For the tweets with no concepts extracted, the network is able to make predictions by comparing the topic of the tweet with other tweets that associated with the same topic. Similarly, for tweets with new topics, the presenting concepts are considered.

Another distinction of long short-term memory is that it does not suffer with vanishing or exploding gradient problem like simple recurrent network does. This works due to the implementation of an identity function which indicates if the forget gate is open or not and makes the gradient remain constant over each time step. This trait of gated networks enables the model to learn long-term dependencies of concepts and topics over time.

4 IMPLEMENTATION

This section presents the implementation of PERSEUS, the datasets used for the experiments, and the baselines chosen for the model comparison.

4.1 Technical Setup

PERSEUS is trained for sentiment categorization task with three classes: Positive, negative, and neutral. The implementation is conducted using Keras[2] with Tensorflow[3] back-end. The embeddings networks contain 32 nodes at the hidden layer for topics, and 128 nodes for concepts. Topics are given by the datasets introduced below, and are 104 in total. Concepts are taken from SenticNet knowledge base (Cambria et al., 2016). From 50,000 SenticNet cocnepts in total, 10,045 occur in the datasets chosen for this research. Public opinions on concepts are set according to Sentic values from SenticNet which are sentiment scores between -1 (extreme negativity) and +1 (extreme positivity) investigated in terms of four affective states (pleasantness, attention, sensitivity, and aptitude). They reflect a common understanding of the associated terms. Public opinions on topics are based on Sentic values as well and calculated by averaging over posts with the same topic. The recurrent network includes three LSTM layers that the first two layers contain 64 cells each, while the third one contains 32 cells. Dropout is applied on inputs and weights during the training phase to prevent overfitting (Srivastava et al., 2014). The model integrates at most 20 past tweets. The experiments are executed 5 times to avoid inconsistency of the neural networks caused by randomly initialized weights, and average results are shown in Section 5.

4.2 Datasets

Table 1: Statistics of the SemEval and Sanders datasets. The datasets are labeled with three classes.

Polarity	SemEval	Sanders
Positive	6758	424
Negative	1858	474
Neutral	8330	2008
Total	16946	2906

PERSEUS is evaluated on Twitter datasets with 19852 tweets in total. We combine Sanders Twitter Sentiment Corpus[4] with the development set of SemEval-2017 Task 4-C Corpus[5]. The SemEval corpus is comparatively more objective than the Sanders corpus, because the annotation of SemEval is done by crowd-sourcing while for Sanders, the classification is done by one person. Furthermore, germane labels are merged to three classes for the SemEval corpus which is associated with a five-point scale. The reasons to combine these two corpora are: 1. They are both human-labeled data. 2. The independence between a corpus and the use of concepts can be verified: The SemEval corpus contains 100 topics, while the Sanders corpus contains only four topics that are 'apple', 'google', 'microsoft', and 'twitter'. As shown in Figure 1, 'moto g' is located very close to these four topics because they are more correlated than others, although it is from the other corpus. The statistics of each dataset is in Table 1.

For training, we use a subset of the combined dataset while the rest (30%) is reserved for testing. The training set is further separated for development and evaluation (30%). We do not use the test set provided by SemEval, because it contains only new topics which are not suitable for examining topic dependencies learned by the network. PERSEUS is able to deal with tweets with unseen topics, but the relations between the unseen topics and learned topics will be lost and the system becomes topic-independent.

Twitter data contains highly informal text such as word stretches *'loooooovee'*, neologisms *'zomg'*, and symbol omissions *'isnt'*, which makes preprocessing very difficult. After the preprocessing which consists of text normalization and lemmatization, only a small number of tweets can not be represented by concepts. The highest number of concepts per tweet is 37. Since SemEval contains more tweets than Sanders, we can find a lager number of tweets per user in SemEval. In average, tweet messages in SemEval contain more extracted concepts (mode: 9) than Sanders (mode: 5), thus SemEval can be better represented. For Sanders, such concept representation may not be sufficient for describing the information contained in the text.

4.3 Baselines

We compare the performance of PERSEUS with five baselines. The first one is purely the Sentic values as mentioned in Section 4.1. The values are combined for each tweet, after which the result together with the number of concepts occurred in the tweet are fed to a shallow fully connected network for training. This is done in order to set up a baseline that demonstrates the performance when no implicit connections of any aspect are concerned.

We compare the neural network-based approach with Support Vector Machine (SVM) which is a

[2]https://keras.io/

[3]https://www.tensorflow.org/

[4]http://www.sananalytics.com/lab/twitter-sentiment/

[5]http://alt.qcri.org/semeval2017/task4/

prominent method for sentiment-related tasks. Two SVM models are trained with different inputs using scikit-learn (Pedregosa et al., 2011). One is a generalized model (Generalized SVM) trained with the presenting concepts and the associated topic (no user information attached), and the other is a personalized model (Personalized SVM) trained with the input of the generalized SVM together with user index and public opinions. The radial basis function (RBF) kernel, the value for the parameters $C = 0.01$ and $\gamma = 1/N\_features$ are set by 10-fold cross-validation on the training data.

We also perform an experiment with convolutional neural network (CNN), which is a widely used method in image processing (Krizhevsky et al., 2012; Lawrence et al., 1997), and has been found to provide good performance for natural language processing tasks as well. Kim (2014) uses the convolutional neural network over static and non-static representations for several sentence classification tasks. They show that a simple convolutional neural network is able to offer competitive results compared to other existing approaches. The structure we used as baseline in the experiment is similar to Kim (2014) but with the following modifications. First, each tweet is represented by the concatenation of its N constituent concepts, and then a convolutional network with two convolutional layers is applied on the concept embeddings as explained earlier. This structure highlights the inner relationship between concepts, especially on their adjacent appearances in a message.

Finally, we use a generalized recurrent neural network (Generalized RNN) to compare the performance considering the dependence between the past and the current tweets when no user related information is used. We use the network proposed for PERSEUS without attaching a user index in the input sequence, and $x_{t*} = [E_{concept}\ E_{topic}]_{t*}$ is set at the input nodes. With such a network, $E_{concept}$ and E_{topic} represent the concepts and topics from a general view, thus $P_{concept}$ and P_{topic} are no longer needed. The tweets are then ordered by the creation time. With user information removed, the network mainly learns by comparing the presenting concepts and the associated topic from different time points.

5 DISCUSSION

In this section, we compare PERSEUS with the chosen baselines. We evaluate the effectiveness of the proposed system by constructing experiments from different angles.

5.1 Model Comparisons

Table 2: Comparison of the performance between PERSEUS and the chosen baselines.

Model	Accuracy	Avg. Recall
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
PERSEUS	**0.6569**	**0.6859**

Table 2 shows the accuracy and average recall of PERSEUS compared with the five baselines described in the preceding section. The granular level for all the models are concept-based to enable a consistent intra-document representation. The Sentic model performs the worst for it is a simple network for combining Sentic values. In PERSEUS, the Sentic values act as public opinions that are not representative on their own. They reflect a general understanding of the concepts which is neither user related nor semantically dependent.

The SVM models provide reasonable results for the given dataset. The performance of the generalized SVM is slightly below that of the generalized recurrent network where the difference is mainly caused by the trait of recurrent networks being able to consider dependencies through time. The fact that there is no significant improvement after adding user information in the personalized SVM shows us that the SVM model in its current form is not a suitable candidate for the task of analyzing individuality in sentiment.

In the work of Kim (2014), the convolutional neural network performs mapping by a sliding window over adjacent words which implies that the order of words appeared in a sentence plays a significant role, i.e., contiguous words have greater dependence. However, for concepts such an interaction is not obvious. The concept itself includes implicit connections between words, therefore this network only studies the co-occurrence of the concepts on the intra-document level.

The generalized recurrent network works better than the convolutional network because the connections to the past as well as between topics are studied. This model intents to capture the trends in public opinions – the information of public preference towards a topic at different time is memorized and analyzed. This baseline shows the effect of Assumption I and III in PERSEUS. By adding the user index in PERSEUS, the performance is further improved (t-test with $p < 0.05$), which indicates that considering the diversity in lexical choices and an individual's relation with the public positively influence the prediction.

5.2 Effect of Associating with Topics

We evaluate the personalized framework on the combined datasets without using the associated topics in order to reflect the effect of topic-opinion relations in Assumption II. The setup of the network is the same as before with one difference: $x_{t*} = [E_{concept}\ P_{concept}]_{t*}$ is set at the input nodes before the user index. The experiment shows an accuracy of 0.5536 and average recall of 0.5429, which is worse than the performance of PERSEUS. This result indicates the benefit of associating sentiment with topics through the components E_{topic} and P_{topic}.

5.3 Effect of Personalization

A general view of the distribution of user frequency in our dataset is shown in Table 3. Majority of the users have only tweeted once, and only 51 users have tweeted more than 5 times. The user with the highest number of tweets has 113 posts.

Table 3: Performance of PERSEUS with users of different numbers of tweets.

# Tweets per User	# User	Accuracy
> 5	51	0.7425
3, 4, 5	206	0.6671
2	714	0.6282

We take different intervals to show in this table because we need a certain number of samples to provide meaningful results, e.g., no user has tweeted exactly 13 times in the experimented data. The results indicate that the more a user tweets, the more accurately PERSEUS is able to predict for the user. Consequently, a high level of personalization requires adequate user data. By treating all the users who tweeted once as one user, the system achieves an accuracy of 0.6461 for this 'one user'. The setting of the input sequence for this group of users is the same with the generalized recurrent network, however they are trained together with other users so that the network is enhanced by comparing between different users. Therefore, we are confident to claim that the overall performance will increase if most users have tweeted more than 5 times, which is very likely in a real world scenario.

5.4 Effect of the Past

We conduct an experiment adding different numbers of past tweets in the input sequence. Given the distribution of user frequency in Table 3, there is no need to consider more than 20 past tweets. Recurrent networks assume that recent events have more impact, therefore more attention is given to close nodes. Nevertheless, Table 4 shows that the network offers better results when relating to a longer history.

Table 4: Performance of PERSEUS considering different numbers of past tweets.

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

When considering only one previous tweet, the performance is very poor because two consecutive tweets may not be related and there is not enough useful information from the previous tweet that can be memorized and extracted for the current tweet. When considering 10 past tweets, the system shows competitive results compared to the generalized recurrent network which considers 20 past tweets (Table 2). The maximum capability of this model can be examined by a larger set of user data in a separate study. These experiments show that a rich set of user data and a network with a sufficient depth of input sequence are the major influential factors of the system.

6 CONCLUSIONS AND FUTURE WORK

We have introduced PERSEUS – a personalized framework for sentiment categorization on user-related data. The framework provides a deeper understanding of user behavior in determining the sentiment orientation. The system takes advantage of a recurrent neural network with long short-term memory to leverage the assumptions as mentioned in Section 1. Evaluated with Twitter text, our experiments have shown the implication of integrating user preferences on lexical choices and topics, the effectiveness of the components used in the system, and a promising future research that PERSEUS can be adapted to offer a better performance.

In the current version, the framework is simplified to concentrate on the *cross-document relation*. Such a simplification is efficient for observing the effectiveness of our system, but does not provide a competitive performance globally since *intra-document relation* is also a very important aspect of sentiment categorization. *Intra-document relation* can be learned more effectively to compensate the information missed by the concept representation. A simple solution is to apply the proposed framework as an additional tool on top of other existing models that concern *intra-document relation* to enhance user understanding. This solution will allow us to compare the system embedded with PERSEUS to state-of-the-art methods so that more profound evaluation can be shown.

There are several aspects where PERSEUS can be extended. First, for the sake of simplicity, we treat public opinions as static in the current version. However, public opinions change over time and a mechanism should be designed to include the evolvement in the framework. Second, an attention model can be combined with the recurrent neural network to enable a more explicit concentration on the information that is related to the current tweet. Such a combination is more beneficial to the task compared to using the recurrent network alone, since recurrent networks tend to emphasize the information that is happened recently. Moreover, the model can be trained with a larger dataset in order to enhance the embeddings for the concepts and topics, to discover the transferability across domains, and to determine an upper bound for influential historical data. Last but not least, observing the performance implemented on automatically labeled dataset may provide clearer indications of user perspective.

For an advanced application, PERSEUS can be adapted following an endorsement of personalization in an artificial companion (Guo and Schommer, 2017). In a multi-user scenario, such an adaptation is realized to improve user experience of communication and interaction by designing user-tailored response.

REFERENCES

Bespalov, D., Bai, B., Qi, Y., and Shokoufandeh, A. (2011). Sentiment classification based on supervised latent n-gram analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 375–382. ACM.

Cambria, E., Fu, J., Bisio, F., and Poria, S. (2015). Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, pages 508–514.

Cambria, E. and Hussain, A. (2015). *Sentic computing: a common-sense-based framework for concept-level sentiment analysis*, volume 1. Springer.

Cambria, E., Poria, S., Bajpai, R., and Schuller, B. W. (2016). Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *COLING*, pages 2666–2677.

Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016a). Neural sentiment classification with user and product attention. In *Proceedings of EMNLP*, pages 1650–1659.

Chen, T., Xu, R., He, Y., Xia, Y., and Wang, X. (2016b). Learning user and product distributed representations using a sequence model for sentiment analysis. *IEEE Computational Intelligence Magazine*, 11(3):34–44.

Cheng, X. and Xu, F. (2008). Fine-grained opinion topic and polarity identification. In *LREC*, pages 2710–2714.

Dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual LSTM (CLSTM) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Gong, L., Al Boni, M., and Wang, H. (2016). Modeling social norms evolution for personalized sentiment classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 855–865.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (ICASSP)*, pages 6645–6649. IEEE.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*.

Guo, S. and Schommer, C. (2017). Embedding of the personalized sentiment engine PERSEUS in an artificial companion. In *International Conference on Companion Technology (to appear)*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Janis, I. L. and Field, P. B. (1956). A behavioral assessment of persuasibility: Consistency of individual differences. *Sociometry*, 19(4):241–259.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.

Meena, A. and Prabhakar, T. (2007). Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *European Conference on Information Retrieval*, pages 573–580. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in Twitter. *Proceedings of SemEval*, pages 1–18.

Nowak, A., Szamrej, J., and Latané, B. (1990). From private attitude to public opinion: A dynamic theory of social impact. *Psychological Review*, 97(3):362.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of SemEval*, pages 27–35.

Reiter, E. and Sripada, S. (2002). Human variation and lexical choice. *Computational Linguistics*, 28(4):545–553.

Saif, H., He, Y., and Alani, H. (2012). Semantic sentiment analysis of Twitter. In *International Semantic Web Conference*, pages 508–524. Springer.

Song, K., Feng, S., Gao, W., Wang, D., Yu, G., and Wong, K.-F. (2015). Personalized sentiment classification based on latent individuality of microblog users. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 2277– 2283.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). LSTM neural networks for language modeling. In *Interspeech*, pages 194–197.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Tang, D., Qin, B., and Liu, T. (2015). Learning semantic representations of users and products for document level sentiment classification. In *ACL (1)*, pages 1014–1023.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for Twitter sentiment classification. In *ACL (1)*, pages 1555–1565.

Teng, Z., Vo, D.-T., and Zhang, Y. (2016). Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1629–1638.

Wang, J., Yu, L.-C., Lai, K. R., and Zhang, X. (2016). Dimensional sentiment analysis using a regional CNN-LSTM model. In *The 54th Annual Meeting of the Association for Computational Linguistics*, volume 225.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631. ACM.

Wiebe, J., Wilson, T., and Bell, M. (2001). Identifying collocations for recognizing opinions. In *Proceedings of the ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, pages 24–31.