Information Sciences 488 (2019) 158–180

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Deep rolling: A novel emotion prediction model for a multi-participant communication context

Huan Rong[a,b], Tinghuai Ma[a,*], Jie Cao[c], Yuan Tian[d], Abdullah Al-Dhelaan[d], Mznah Al-Rodhaan[d]

[a] School of Computer & Software, Nanjing University of Information Science & Technology, Jiangsu, Nanjing 210-044, China
[b] CICAEET, Jiangsu Engineering Center of Network Monitoring, Nanjing University of information science & Technology, Nanjing 210-044, China
[c] Reading Academy, Nanjing University of information science & Technology, Jiangsu, Nanjing 210-044, China
[d] Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11362, Saudi Arabia

ARTICLE INFO

Article history:
Received 1 November 2018
Revised 9 March 2019
Accepted 12 March 2019
Available online 12 March 2019

Keywords:
Emotion recognition
Time series forecasting
Natural language processing
Deep learning

ABSTRACT

Nowadays, the amount of user-generated contents (UGCs) or texts has surged exponentially. Therefore, recognizing emotions from these texts can bring about lots of advantages. In this paper, we have proposed a novel model named ***Deep Rolling*** to predict emotion for target participant in a multi-participant communication context. First, the proposed method converts a text collection into a set of n-dimension vectors for emotion representation and re-organizes texts into a sequence in time order. Then, Deep Rolling can predict the emotion of target participant corresponding to a future time point. Second, apart from simply taking in texts posted by target participant via LSTM, the proposed method has also incorporated texts posted by other participants at every time step by CNN. In this way, Deep Rolling can predict target participant's emotion by processing emotions from both the target and all the other participants in an ensemble way. Finally, data factorization has also been introduced into Deep Rolling to enhance the overall prediction efficiency. According to experimental results, compared with the state-of-art methods, our proposed model has achieved the best prediction precision on different target participants. At the same time, Deep Rolling has also maintained the prediction efficiency at an acceptable level.

1. Introduction

Nowadays, more and more people are communicating with each other through a diverse range of applications on different topics and the amount of user-generated contents (UGCs) such as reviews, comments or message records has surged exponentially [19,29–31,35]. The information hidden behind these UGCs can be applied to figuring out more proper strategies for service adjustment, potential customer detection and the precise delivery of advertisements [28]. Therefore, in this paper, we have proposed a novel model to predict target participant's emotion based on a large amount of UGCs or texts collected in a multi-participant communication context. In other words, according to specific texts, emotions including happiness, sadness, anxiousness, anger and surprise should be perceived from the current context [48].

* Corresponding author.
E-mail address: thma@nuist.edu.cn (T. Ma).

https://doi.org/10.1016/j.ins.2019.03.023
0020-0255/

Unfortunately, current research associated with emotion prediction has primarily been conducted by classification which means that when a text at one time point has been given, an emotion at a corresponding time point may be obtained via a fine-tuned model [21]. Moreover, due to the characteristics of a multi-participant communication context, the emotion state of surrounding participants can stimulate the emotional changes of a certain participant; in other words, a simple model is not sufficient to predict emotions for one target participant in a multi-participant communication context. Therefore, more effort should be devoted to this area of study.

In this paper, inspired by the state-of-the-art work of text-based emotion prediction and time series forecasting, we have proposed a novel model to predict emotion for a target participant in the multi-participant communication context denoted as Deep Rolling. First, by reorganizing texts posted by a target participant into a sequence according to time order, the proposed model moves forward, similar to time series prediction, with the help of an observation window. In this way, a target's emotion at a future time point can be predicted, even if the text content at that time is still unknown. Second, in terms of the novelty of the proposed model, apart from taking in texts posted by a target participant, Deep Rolling also incorporates texts given by other participants at every time step. In other words, emotion will be simultaneously predicted from both the target and other participants at every time step in an ensemble way so that the emotional stimulation caused by surrounding participants can be captured. The experimental results have shown that our proposed model can outperform classical and state-of-the-art models on different target participants. In addition, the time cost of Deep Rolling has still been controlled at an acceptable level. Moreover, in terms of other participants, first, a model that includes other participants has outperformed the model without other participants. Second, the other participants' emotion can provide positive or negative stimulation to the target participant's emotion prediction. Finally, those who have a high level of interaction with the target participant are more critical to the precision of the target participant's emotion prediction.

The rest of this paper is organized as follows. Section 2 presents the related works associated with text-based emotion prediction and time series prediction which provided the motivation of this paper. The details regarding the proposed model will be described in Section 3, including the paradigm's architecture and problem statement, *Affective Space Mapping* for emotion representation, the emotion processing of both the target and other participants and the data factorization of the proposed model. Finally, experimental analysis will be presented in Section 4, followed by the discussion and conclusion of this paper.

2. Related work

2.1. Text-based emotion prediction

Currently, a large amount of works associated with text-based emotion prediction are mainly conducted by classification, which means that the emotion of new texts or instances can be perceived by a fine-tuning model through a large amount of texts [22]. For example, Alm et al. [2] has utilized supervised machine learning architecture to classify the emotion of sentences in the narrative domain of children's fairy tales. In addition, neural networks that contain memory modules such as Recurrent Neural Networks (RNN) [42], Long Short-Term Memory (LSTM) [36] and Bidirectional Long Short-Term Memory (Bi-LSTM) [21] have been frequently utilized. For example, Zhao et al. [49] have successfully predicted the online news emotion via a Bi-LSTM neural network.

From another aspect, the convolution neural network (CNN) [41] has also been used in text-based emotion prediction. As the convolutional and pooling layer can deeply capture local and global features, CNN has achieved excellent outcomes on extracting emotional concepts from given texts [13,23]. More specifically, in [34], CNN was used to extract emotional features for better emotion representation of utterances. In [32], in order to predict the emotion intensity of Tweets, an ensemble framework containing CNN with word embedding has been proposed. The proposed framework can predict the emotion of a given tweet according to the average predictions of individual methods. In addition, Florian Krebs et al. [24] have combined CNN and LSTM to predict the emotions involved in reactions to Facebook posts. The proposed model first uses emotion-labeled and unlabeled data to fine-tune an SVM [20] model. Then the emotion of posts can be obtained by the fine-tuned SVM based on which CNN and LSTM have been applied collaboratively to predict emotions associated with reactions to posts.

Other works associated with text-based emotion prediction include [4], in which emotion from affective text has been detected by a joint emotion-topic model. This model has added a layer to Latent Dirichlet Allocation (LDA) [6] for emotion modeling based on which a set of topics can be generated. Then emotional terms on these topics will be mined out. In addition, Chih-Hao Chen et al. [9] have proposed a computational method to predict emotion from short chatting messages and a Valence-Arousal emotion space is adopted to quantify and measure different emotions.

2.2. Time series forecasting

As mentioned above, in this paper, we intend to predict emotions for one target participant by organizing texts posted at each time step as a time series. Fortunately, there is a large amount of works associated with time series forecasting applied in different fields. Basically, the statistical models such as Vector Autoregression (VAR) [3], Autoregressive Integrated Moving Average (ARIMA) [38] and Kalman Filters (KF) [8] have been widely utilized in time series forecasting. Typical works include the analysis of a monetary transmission mechanism by VAR [37], the cycle length prediction by ARIMA [33] and the

peak power prediction by an extended KF [14]. Unfortunately, the above models cannot represent non-linear relationships and do not differentiate among the driving input terms. Consequently, various nonlinear auto-regressive exogenous (NARX) [27] models have been proposed. For example, Yan et al. [47] have proposed a substructure vibration NARX approach to predict the damage and a NARX neural net-based model has been proposed in [25] for the real-time prediction of air and gas optimization.

Despite the fact that a substantial effort has been devoted to optimizing the above linear and non-linear models, the obvious drawback is that these linear or non-linear models cannot appropirately capture hidden relationships. Therefore, RNN [42], along with its variants such as like Gated Recurrent Units (GRU) [12], have drawn a great deal of attention due to their capacity to capture unobserved dependency. Representative applications of RNN on NARAX time series forecasting include dynamic rainfall prediction [11], brain connectivity estimation [45] and electricity consumption forecasting [40].

Other state-of-the-art works associated with time series prediction include a novel residual network, R2N2, to capture both linear and non-linear dependency [17]; a dual-stage attention-based RNN model for non-linear time series prediction [39]; a dilated convolution neural network for fast forecasting [7]; a deep denoising autoencoder-based DNN model to predict the morbidity of gas infections [43] and a predictive clustering algorithm [18] to detect features from non-successive positions of the series.

2.3. Motivation

Based on above works, two drawbacks can be observed. First, the emotion prediction of most works is actually implemented by classification. In other words, the emotion state can only be predicted when the text posted by one target participant has already been obtained. It is evidently inappropriate if we want to obtain emotion at a future time point when the text content is still unknown. Second, from the perspective of time series forecasting, the prediction can be conducted by a linear or non-linear time series forecasting model; however, in a multi-participant communication context, it is natural to consider that one participant's emotion can be affected or stimulated by other participants. Such detail has not been considered by current time series forecasting models. Consequently, existing linear or non-linear time series prediction models are ***not*** sufficient.

To overcome the above limitation, in this paper, we propose a novel model called Deep Rolling that is intended to predict the emotion of a target participant in a multi-participant communication context. On the one hand, the proposed model reorganizes all of the texts posted by a target participant as a time series. At each time step, the target participant's emotions will be perceived by a multi-layer LSTM that is applied as a means of time series prediction. In this way, future emotion can be obtained, even if the text content is still unknown. On the other hand, in order to predict the target participant's emotion more precisely, at each time step we have also incorporated a set of texts posted by other participants that are processed by a CNN component. In this way, the emotion perceived at each time step will be simultaneously derived from a target participant via LSTM and other participants by CNN. Consequently, the prediction of Deep Rolling can reflect stimulation from all of the other participants in a multi-participant communication context.

In conclusion, Deep Rolling is considered as a non-linear model to overcome emotion prediction problems involving a time series. In addition, at each time step, the emotion has been predicted via LSTM and CNN in an ensemble way. The details regarding our proposed model, Deep Rolling, will be elaborated upon in Section 3.

3. The proposed prediction model

The brief architecture of Deep Rolling has been demonstrated in Fig. 1. First, texts written by a target participant and all of other participants will be organized into a time series and converted into n-dimension emotion vectors (corresponding to the notation "ASM" in Fig. 1) through ***Affective Space Mapping***. It is worthwhile to mention that, at each time step, all of other participants' texts will also be collected. In addition, in terms of the converted emotion vector or ASM, we have added the constraint that the maximum value can only be occupied by one dimension so that the specific emotion type can be ascertained. Second, at each individual time step, a single emotion vector from the target participant will be fed into a multi-layer LSTM with dropout modules. Third, a set of emotion vectors from all of the other participants that is related to the current time step will be packaged and fed into the CNN component. Finally, an emotion vector that also includes $\boldsymbol{n}$ dimension will be output via a fully connected and softmax layer. Such an emotion vector will be considered as the predicted emotion representation that points to the next time step. Consequently, compared with the input of the next time step, model optimization can be conducted based on "*Loss*".

Moreover, as shown in Fig. 1, we have also set an observation window, w, with moving stride s in Deep Rolling. When moving the observation window towards the end of the entire time series whose length is m, an emotion vector or ASM at time step $m+1$ will be predicted, even if the text content at time step $m+1$ is still unknown. In conclusion, at each time step, the emotion vectors from one target, as well as all of the other participants, are processed and transformed via a multi-layer neural model in a "***Deep***" way, after which the model will move forward according to the value of the window size and window stride, as a way of "***Rolling***" (***Deep Rolling***).

For the rest of this section, in 3.1 we will present the problem statement for Deep Rolling along with a set of necessary notations. In addition, steps to convert every text into an n-dimension vector, or ***Affective Space Mapping***, will be elaborated in 3.2. More importantly, the emotion processing of the target participant by LSTM and all of the other participants via CNN

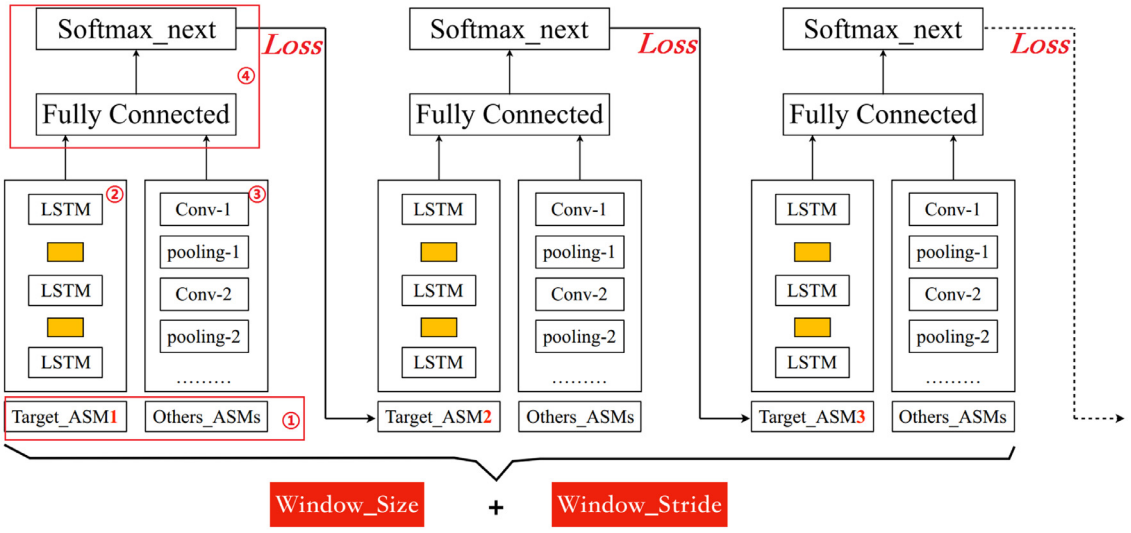


Fig. 1. The architecture of our proposed model (Deep Rolling).

will be analyzed in 3.3 and 3.4 respectively. Finally, the data processing by matrix, or data factorization, of the whole model will be illustrated in 3.5.

3.1. Problem statement

First, given a period of time from *Tstart* to *Tend*, an original text set $D=\{d_1, d_2, ...,d_i|i\in[Tstart,Tend]\}$ has been collected from a multi-participant communication context following a specific time order. d_i represents the collected text given by one participant at time step *i*. Naturally, in order to facilitate the following data processing, it is necessary to map each text, d_i, into an affective space by representing each d_i as an *n*-dimensional emotion vector or ASM. Therefore, we introduce an affective space: $E=\{e_1, e_2, ...,e_n|e_n\in R\}$. In terms of the affective space *E, n* is the number of incorporated emotions and every dimension represents the intensity of the corresponding emotion by numeric value. In this way, each d_i will be projected to an emotion vector, $E\in R^n$, by ***Affective Space Mapping*** whose detailed steps will be elaborated upon in 3.2. Particularly, when mapping each d_i to an emotion vector *E*, we ensure that the maximum value will only be occupied by one dimension; thus, at each time step, *i*, we concentrate on the most obvious emotion reflected by d_i. Finally, the original text set, *D*, will be converted into $D'=\{E_1, E_2, ...,E_i|i\in[Tstart,Tend]\}$, where $E_i\in R^n$.

Second, since the original text set *D* was constructed in a multi-participant communication context (thus, for $D'=\{E_1, E_2, ...,E_i|i\in[Tstart,Tend]\}$), each E_i belongs to a certain participant. Further, after ***Affective Space Mapping***, an emotion vector-based time series of one target participant will be extracted from *D'*, denoted as $D'_{target}=\{E_{target\_1}, E_{target\_2}, ..., E_{target\_m} \mid 0<m\le i\in[Tstart,Tend]\}$, where *m* represents the length of D'_{target} and *m* is a subscript related to the target participant.

Moreover, when it comes to $D'_{target}=\{E_{target\_1}, E_{target\_2}, ..., E_{target\_m} \mid 0<m\le i\in[Tstart,Tend]\}$, since D'_{target} has remained in time order, the emotion vectors of all of the other participants between two consecutive time points, *m* and $m+1$, will be denoted as $E_{others\_m\_m+1}$. It is worthwhile to mention that, first, it is obvious that the notation E_{target_m} represents one emotion vector of the target participant derived from text *d* at time step *m*. However, $E_{others\_m\_m+1}$ means a set of emotion vectors of all of the other participants between a pair of adjacent target participants at time step *m* and $m+1$. Second, the notation of $E_{others\_m\_m+1}$ has reflected the fact that, in our proposed model, in terms of the emotion of the other participants, only those posted between the adjacent appearance of the target participant will be considered. In this way, the problem that our proposed model must resolve can be expressed as follows.

Problem Statement (Emotion Prediction by Deep Rolling): Given a set of selected emotions that our proposed model focuses on, by mapping every text *d* at time step *i* into an affective space, $E=(e_1, e_2, ...,e_n)\in R^n$, where $e_n\in R$, the original text set constructed in a multi-participant communication context following a time order will be converted into $D'=\{E_1, E_2, ...,E_i|i\in[Tstart,Tend]\}$. From there, a time series associated with the target participant will be extracted as $D'_{target}=\{E_{target\_1}, E_{target\_2}, ..., E_{target\_m} \mid 0<m\le i\in[Tstart,Tend]\}$. Denoting emotions of all the other participants between the adjacent target appearance (E_{target_m}, E_{target_m+1}) as $E_{others\_m\_m+1}$, the emotion vector-based time series D'_{target} can be further converted into a pair-based sequence such as $D'_{target\_pair}=\{(E_{target\_1}, E_{others\_1\_2}), (E_{target\_2}, E_{others\_2\_3}), ..., (E_{target\_m-1}, E_{others\_m-1\_m}), (E_{target\_m}, \text{None})|0<m\le i\in[Tstart,Tend]\}$. More importantly, an observation "window" with the size *w* and stride *s* will also be set and the constraint of $s\le w$ should always be fulfilled in order to guarantee the effectiveness of the window's forward moving. In this way, multiple pieces, $D'_{target\_pair\_w}$, can be extracted from D'_{target_pair} such as $D'_{target\_pair\_w}=\{(E_{target\_t}, E_{others\_t\_t+1}),$

(E_{target_t+1}, $E_{others\_t+1\_t+2}$), ..., ($E_{target\_t+(w-1)}$, $E_{others\_t+(w-1)\_t+w}$) | $0<t<m$, $0<w<m$, $0<t+w\leq m$} based on which the emotion vector $E_{target\_t+(w-1)+1}$, or E_{target_t+w}, at time step $t+w$ will be predicted.

Afterwards, the $D'_{target\_pair\_w}$ will move forward by stepping to the index of *target_t+s* as the beginning of next $D'_{target\_pair\_w}$ piece until reaching the end of D'_{target_pair}. Consequently, following the above rules, the emotion vector E_{target_m+1} at time step $m+1$ will be predicted when given D'_{target_pair}=(E_{target_1}, $E_{others\_1\_2}$), (E_{target_2}, $E_{others\_2\_3}$), ..., (E_{target_m-1}, $E_{others\_m-1\_m}$), (E_{target_m}, None)|$0<m\leq i\in$[*Tstart, Tend*]. The complete process of data transformation as well as the prediction task have been concluded in Algorithm 1.

Algorithm 1 The intact data transformation as well as prediction task of proposed model Deep Rolling.

1: Collect original text set D={$d_1, d_2,...,d_i$ |i∈[*Tstart,Tend*]} from a multi-participant-communication context in time order.
2: Select a set of Emotions=(emotion$_1$, emotion$_2$, ..., emotion$_n$).
3: Conduct ***Affective Space Mapping*** to convert D into D'={$E_1, E_2,...,E_i$ |i∈[*Tstart,Tend*]}
where E_i is an n-dimensional emotion vector=(e_1, e_2, ..., e_n), e_n ∈R corresponding to every d_i.
4: Select a target participant for emotion prediction.
5: Extract time series D'_{target}={E_{target_1},E_{target_2},...,E_{target_m} | $0<m\leq i\in$[*Tstart,Tend*]} from $D'$for prediction
where the length of D'_{target} is m.
6: Insert emotion vectors of all the other participants between a pair of target's adjacent appearance like:
D'_{target_pair}={(E_{target_1},$E_{others\_1\_2}$),(E_{target_2},$E_{others\_2\_3}$),...,(E_{target_m-1},$E_{others\_m-1\_m}$),(E_{target_m},None)}
s.t. $0<m\leq i\in$[*Tstart,Tend*]
where E_{target_m} is one emotion vector of target participant at time step m
and $E_{others\_m-1\_m}$ is a set of emotion vectors of all the other participant between time step *m-1* and m
7: Set the value of window size (w) and window stride (s)
8: ***Training***: model rolling by extracting window-size-pieces 1 to n:
$D'_{target\_pair\_w1}$={(E_{target_t},$E_{others\_t\_t+1}$), (E_{target_t+1},$E_{others\_t+1\_t+2}$),...,($E_{target\_t+(w1-1)}$,$E_{others\_t+(w1-1)\_t+w1}$)}
→ Label: E_{target_t+w1}
Model Rolling by *t: = t+s*.
$D'_{target\_pair\_w2}$={(E_{target_t},$E_{others\_t\_t+1}$), (E_{target_t+1},$E_{others\_t+1\_t+2}$),...,($E_{target\_t+(w2-1)}$,$E_{others\_t+(w2-1)\_t+w2}$)}
→ Label: E_{target_t+w2}
Model Rolling by *t: = t+s*.
... ...
$D'_{target\_pair\_wn}$={(E_{target_t},$E_{others\_t\_t+1}$), (E_{target_t+1},$E_{others\_t+1\_t+2}$),...,($E_{target\_t+(wn-1)}$,$E_{others\_t+(wn-1)\_t+wn}$)}
→ Label: E_{target_t+wn}
s.t. $0<t<m$, $0<w<m$, $0<t+w\leq m$
9: Repeat 08 until reaching the end of D'_{target_pair}, or (E_{target_m}, None).
10: ***Predicting***: Feed the final $D'_{target\_pair\_w}$ piece to the fine-tuned model.
Compute emotion vector E_{target_m+1} at time step m+1.

According to Algorithm 1, it is obvious that Deep Rolling training is conducted on the entire pair-based sequence D'_{target_pair}. Further, the instances for training consist of multiple $D'_{target\_pair\_w(1\ to\ n)}$ pieces extracted from D'_{target_pair} via a moving window forward stride by stride. Also important is the fact that, for every instance, the label for training is an emotion vector aimed at the next time point of the current $D'_{target\_pair\_w}$ piece. Following such a routine, when the model rolling to the end of the whole sequence D'_{target_pair}, the emotion vector aiming at the next time point $m+1$ of D'_{target_pair} will be predicted. In this way, the proposed model, Deep Rolling, can ensure that even if the text posted by the target participant at time point $m+1$ is still unknown, it is also able to predict an emotion vector at that time. And, at every time step, Deep Rolling can consider and capture the effect of other participants on the target participant.

Finally, despite the fact that Algorithm 1 has conducted emotion prediction for only once, it is feasible to attach the new predicted emotion vector E_{target_m+1} to the end of D'_{target_pair} as (E_{target_m+1}, None) and continue moving the window forward so that the emotion vector at $m+2$, $m+3$,.... $m+n$ can also be predicted.

3.2. Affective Space Mapping

The specific steps of ***Affective Space Mapping*** have been listed in Fig. 2. According to Fig. 2, it is evident that the final goal of ***Affective Space Mapping*** is to convert every text d_i into an n-dimension emotion vector, or ***ASM***, via a Bi-LSTM [21] based emotion classifier.

First, after collecting the initial text set D={d_1, d_2, ..., d_i|i ∈ [*Tstart,Tend*]}, all of the texts will be annotated, one by one, in the format of [target_flag, Emotion]. The former value, target_flag (0 or 1), signifies whether the current text at time step i has been posted by the target participant and the latter means that the index (0 to n-1) of a specific emotion corresponds to the one in the selected Emotions = (emotion$_1$, emotion$_2$, ..., emotion$_n$).

After annotation, the emotion index-based series of text set D will be obtained in step 2. Particularly, the index series of the target participant can be extracted according to the value of target_flag (0 or 1). Afterwards, the statistical information

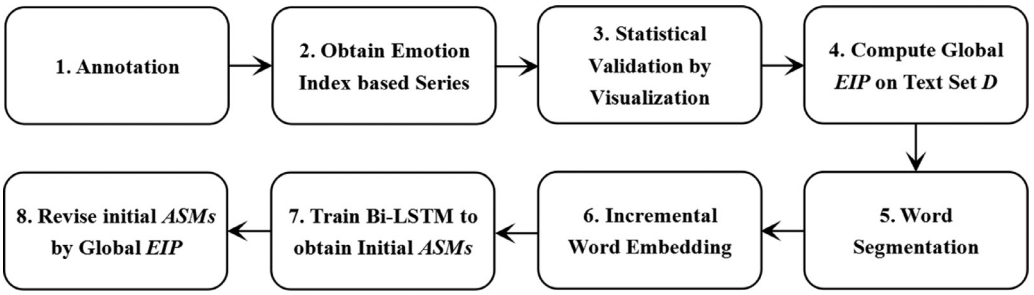


Fig. 2. Affective Space Mapping: Convert text into an n-dimension vector or ***ASM***.

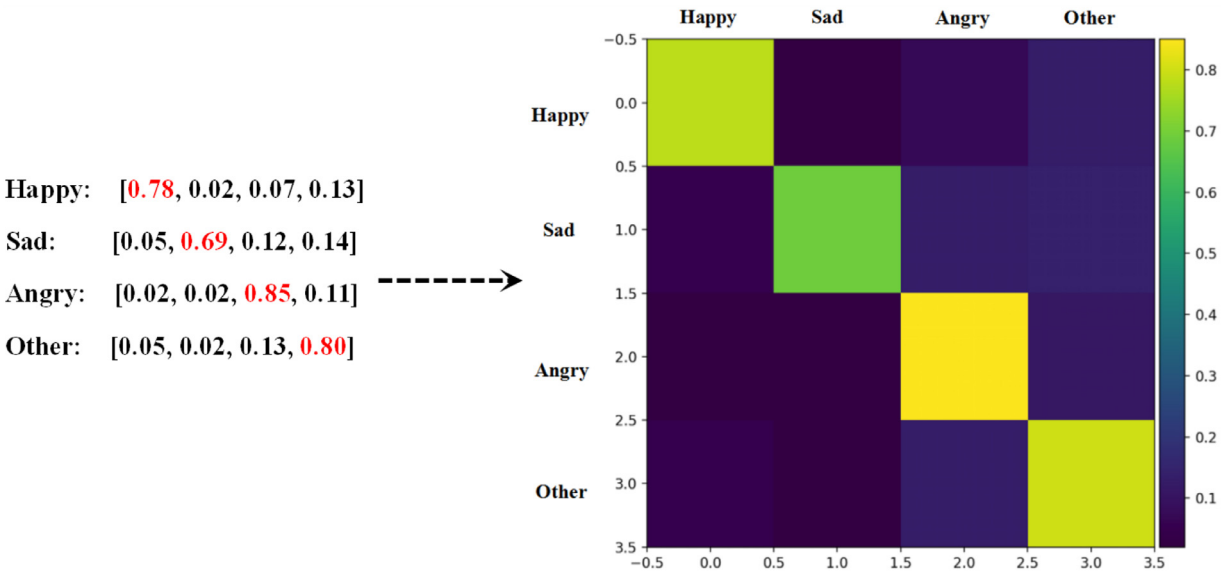


Fig. 3. The color map of global *EIP* on the entire text set *D*.

concerning emotion distribution and the emotion tendency of whole text set *D*, as well as the target participant, will be visualized in step 3 for validation purposes.

Next, in step 4, the global emotion interaction pattern (*EIP*) [50] of the emotion index-based series on text set *D* will be computed as a kind of preparation. An example of *EIP* computation on the whole text *D* has been illustrated in Table 1. As shown in Table 1, by setting up a global interaction dictionary corresponding to selected Emotions=(Happy, Sad, Angry, Other), the *EIP* can be recorded by scanning the whole emotion index-based series of text set *D*. After normalization via *softmax*, the *EIP* will be output as the combination of all of the lists in global interaction dictionary. An example of visualizing computed *EIP* has been presented in Fig. 3 when given Emotions=(Happy, Sad, Angry, Other). From Fig. 3, it is evident that,

Table 1
The specific steps to compute global emotion interaction pattern on the emotion index based series.

Input: Selected emotions = (Happy, Sad, Angry, Other) and the emotion index based series **Output**: The global *EIP* on text set *D*
01. Set up a global interaction dictionary which is initialized as: dict={'happy':[0,0,0,0], 'sad':[0,0,0,0], 'angry':[0,0,0,0], 'other':[0,0,0,0]} **02**. Set up an observation window as size 2. **03**. Scan the whole emotion index based series by above observation window in order. **04**. Update the global interaction dictionary according to the encountered pair like: ('sad', 'angry') → dict={'happy':[0,0,0,0], 'sad':[0,0,**1**,0], 'angry':[0,**1**,0,0], 'other':[0,0,0,0]} **05**. Move window by 1 stride and repeat step 03 to 05 until the end of series like: ('angry', 'sad') → dict={'happy':[0,0,0,0], 'sad':[0,0,**2**,0], 'angry':[0,**2**,0,0], 'other':[0,0,0,0]} ('happy', 'other') → dict={'happy':[0,0,0,**1**], 'sad':[0,0,**2**,0], 'angry':[0,**2**,0,0], 'other':[**1**,0,0,0]} **06**. Conduct *softmax* on all the lists in dict for normalization. **07**. Return the normalized matrix consisting of all the lists in dict as final global *EIP*

Table 2
Extract training instances for target participant from D'_{target} with a given window size and stride.

Input: $D'=\{E_1, E_2, ...,E_i \mid i \in [Tstart,Tend]\}$ $D'_{target}=\{E_{target\_1}, E_{target\_2}, ..., E_{target\_m} \mid 0 < m \le i \in [Tstart, Tend]\}$ window size w and stride s
Output: Instances for training Deep Rolling
$D'_{target\_w1}=\{E_{target\_t}, E_{target\_t+1}, ..., E_{target\_t+(w1-1)}\} \rightarrow E_{target\_t+w1}$ Model Rolling $t: = t+s$
$D'_{target\_w2}=\{E_{target\_t}, E_{target\_t+1}, ..., E_{target\_t+(w2-1)}\} \rightarrow E_{target\_t+w2}$ Model Rolling $t: = t+s$
$D'_{target\_w3}=\{E_{target\_t}, E_{target\_t+1}, ..., E_{target\_t+(w3-1)}\} \rightarrow E_{target\_t+w3}$ Model Rolling $t: = t+s$
... ...
$D'_{target\_wn}=\{E_{target\_t}, E_{target\_t+1}, ..., E_{target\_t+(wn-1)}\} \rightarrow E_{target\_t+wn}$

with the operation of *softmax* as normalization, the maximum value of an emotion vector is occupied by only one dimension. In other words, every list in the above global interaction dictionary can be considered as an emotion vector of one type of emotion.

With the acquaintance of global *EIP* on entire text set D, the real steps for converting every text d_i in $D=\{d_1, d_2, ...,d_i \mid i \in [Tstart,Tend]\}$ into a n-dimensional emotion vector E will be conducted. First, in step 5 as shown in Fig. 2, word segmentation will be performed on text set D and all the distinct words except punctuation will be collected and assigned with a word_id based on which two dictionaries for the mapping between a word and its corresponding id will be constructed. In this way, every text d_i in D will be represented by a word-id sequence.

Second, in step 6, we adopted the skip-2gram [16] model to perform word embedding on text set D. As a result, every distinct word collected above can be represented by a numeric vector with 256 dimensions. In this way, we have pre-trained embeddings for words in text set D. According to Zhao et al. [49], the pre-trained word embeddings are beneficial to the classification precision of Bi-LSTM compared with that derived from the co-trained ones.

Next, in step 7, the n-dimensional emotion vector of every text d_i will be computed by a multi-layer Bi-LSTM-based emotion classifier [50]. The classifier has taken every text d_i as an embedding sequence and will output an n-dimensional vector as "logits". By fine-tuning the above emotion classifier via cross entropy against the annotated emotion index, the final n-dimensional logits will be computed and then normalized through *softmax*. In this way, the initial emotion vectors can be obtained.

Finally, in step 8, validation of the emotion vectors computed by the above Bi-LSTM classifier will be conducted. By checking whether the index of the maximum value of every computed vector is identical to that provided in the annotation step, we substitute an incorrect emotion vector with the one contained in the global *EIP* as shown in Fig. 3 according to the annotated emotion index. In this way, after Affective Space Mapping, the initial text set, $D=\{d_1, d_2, ...,d_i \mid i \in [Tstart,Tend]\}$ will be converted into $D'=\{E_1, E_2, ...,E_i \mid i \in [Tstart,Tend]\}$, where E_i is an n-dimensional emotion vector as $(e_1, e_2, ...,e_n)$, $e_n \in R$. Moreover, the time series $D'_{target}=\{E_{target\_1}, E_{target\_2}, ..., E_{target\_m} \mid 0 < m \le i \in [Tstart,Tend]\}$ will be extracted from D' if a target participant was selected.

3.3. Emotion processing for target participant

When given the emotion vector-based sequences $D'=\{E_1, E_2, ...,E_i \mid i \in [Tstart,Tend]\}$ and $D'_{target}=\{E_{target\_1}, E_{target\_2}, ..., E_{target\_m} \mid 0 < m \le i \in [Tstart, Tend]\}$, the next step is to train the proposed model, Deep Rolling, and predict the emotions for a target participant. By selecting a proper "observation window" with size w and stride s, multiple window-sized pieces will be extracted from D'_{target} for training (as shown in Table 2). Those window-sized pieces will be considered as the instances containing the emotion of the target participant whose label is the true emotion vector pointing to the next time step of the current observation window. Moreover, a multi-layer LSTM with dropout modules will be adopted to process the extracted window-sized pieces for the target participant (as shown in Fig. 4).

In Fig. 4, after extracting one window-sized piece $D'_{target\_w(1\ to\ n)}$ of a target participant whose shape is $[w,n]$, where n is the dimension of an emotion vector determined by selected Emotions, it will be fed into a multi-layer LSTM whose final output points to the emotion vector at time step $t+w$. Furthermore, the shape of one window-sized piece $D'_{target\_w(1\ to\ n)}$ input into LSTM can be enlarged as $[1, w, n]$. Meanwhile, the output of LSTM for one window-sized piece $D'_{target\_w(1\ to\ n)}$ can also be kept in the shape of $[1, w, n]$ by setting the number of neurons in LSTM's output layer also as n. The output can be considered as a set of states collected from every LSTM cell within an observation window. In this way, following the window-sized piece-based processing (as shown in Fig. 4), we have actually set the target participant's emotion processing at a window level, or one extraction of a window-sized piece from D'_{target}.

Unfortunately, as it has been mentioned above, in a multi-participant communication context, the emotion of a target participant can be affected by other surrounding participants. As a result, a simple multi-layer LSTM (as illustrated in Fig. 4)

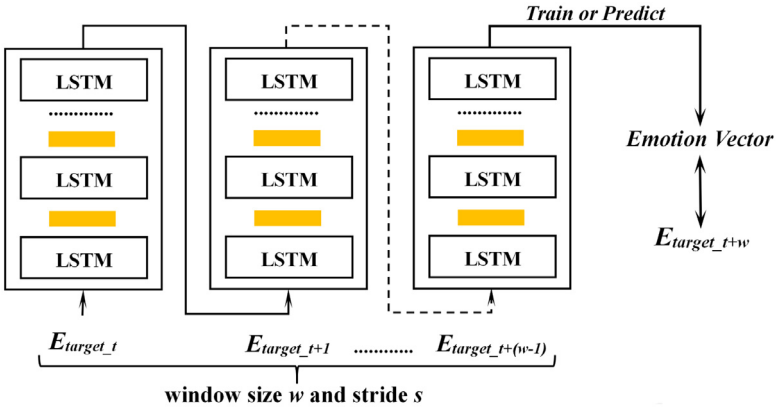


Fig. 4. Use the multi-layer LSTM with dropout to process the emotion of a target participant.

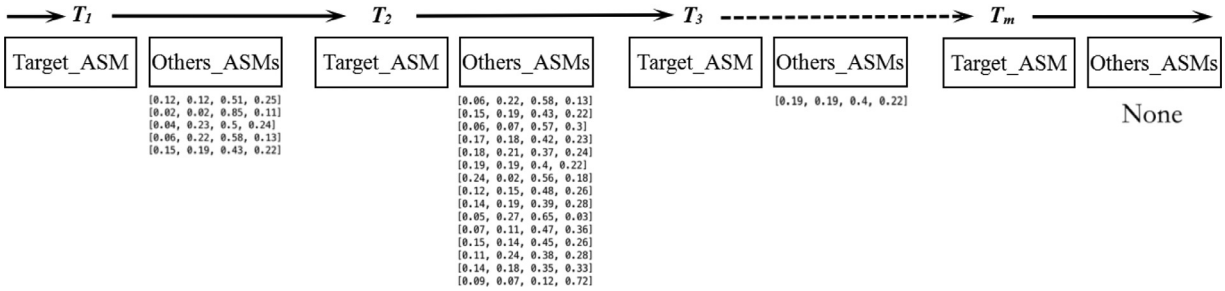


Fig. 5. The abstraction of the appearance of texts from both the target participants and others.

is not sufficient. In other words, in terms of our proposed model, Deep Rolling, it is still necessary to incorporate emotions from all of the other participants at every time step.

3.4. Emotion processing for other participants

When given D'_{target}={E_{target_1},E_{target_2},...,E_{target_m} | $0 < m \le i \in [Tstart,Tend]$}, the emotion vectors of all the other participants can be inserted and organized as D'_{target_pair}={(E_{target_1},$E_{others\_1\_2}$), (E_{target_2},$E_{others\_2\_3}$),...,(E_{target_m-1},$E_{others\_m-1\_m}$), (E_{target_m},*None*)}. Fig. 5 has presented an abstraction of D'_{target_pair} and a direct problem is that between an adjacent pair of target participants, the amount of emotion vectors or ASMs from all of the other participants is not fixed or may even not exist. Consequently, it is necessary to adopt a flexible architecture that can deal with all of the cases of emotion vectors from other participants.

To solve the above problem, in this paper, we use CNN to process the emotion of other participants inspired by its application on the classification of images with multiple channels such as RGB [44]. Specifically, all of the vectors of other participants between an adjacent pair of target participants, or the input to CNN, will be stored into a "cube" shaped as [*channels, height, width*]. Such a cube has consists of multiple layers (or channels) and each layer consists of several *n*-dimension emotion vectors or ASMs. Since we have implemented Deep Rolling by Tensorflow [1], therefore, the input (organized as above cube) to CNN component is required to be shaped as [*height, width, channels*]. In other words, in the rest of this paper, we simply adopt the notation [*height, width, channels*] to represent a cube that is initially shaped as [*channels, height, width*].

Table 3 concludes the steps to construct a cube to store the emotion vectors of all of the other participants posted between an adjacent pair of target participants. As listed in Table 3, in the first step, when given the whole text set *D'*, a "max_span" will be computed by scanning *D'* so that the maximum number of emotion vectors, or E_i*s* posted by all the other participants, between an adjacent pair of target participants will be obtained. Afterwards, the total amount of numbers or bits derived from the above E_i*s* will be calculated. Finally, as shown in step 3 of Table 3, the number of channels or layers denoted as *uniform_channels* will be computed.

A brief demonstration has been presented in Fig. 6. Suppose a set of Emotions = ($emotion_1$, $emotion_2$, $emotion_3$, $emotion_4$) has been selected and the maximum span between an adjacent pair of target participants, in terms of given D'={E_1, E_2, ...,E_i|$i \in [Tstart,Tend]$}, is 147. Therefore, the upper limit of bits_amount to hold the emotion vectors of all of the other participants is 147*4 = 588 and the corresponding uniform_channels is 588/(4*4)≈37.

Consequently, in order to store a set of emotion vectors $E_{others\_m\_m+1}$ of all the other participants between any adjacent target pair (E_{target_m}, E_{target_m+1}), a cube with 37 layers (or channels) of 4*4 square has been constructed (as shown in Fig. 6).

Table 3
The steps to construct a cube to store other participants' emotion vectors at certain time step.

Input: $D'=\{E_1, E_2, ..., E_i | i \in [Tstart,Tend]\}$
where E_i is an n-dimensional emotion vector=$(e_1, e_2, ..., e_n)$, $e_n \in R$
Output: A cube to store the emotion vectors of all of the other participants at different time steps

01. Compute the maximum span (max_span) in $D'=\{E_1, E_2, ..., E_i | i \in [Tstart,Tend]\}$
where max_span represents the maximum amount of E_is between a pair of target's emotion vectors
02. Compute bits_amount := max_span * n where n is the dimension of an emotion vector
03. Compute uniform_channels := bits_amount / n^2
04. If uniform_channels is not an integer, then:
uniform_channels := uniform_channels + 1
05. Return a cube with shape [*height* = *n*, *width* = *n*, uniform_channels]

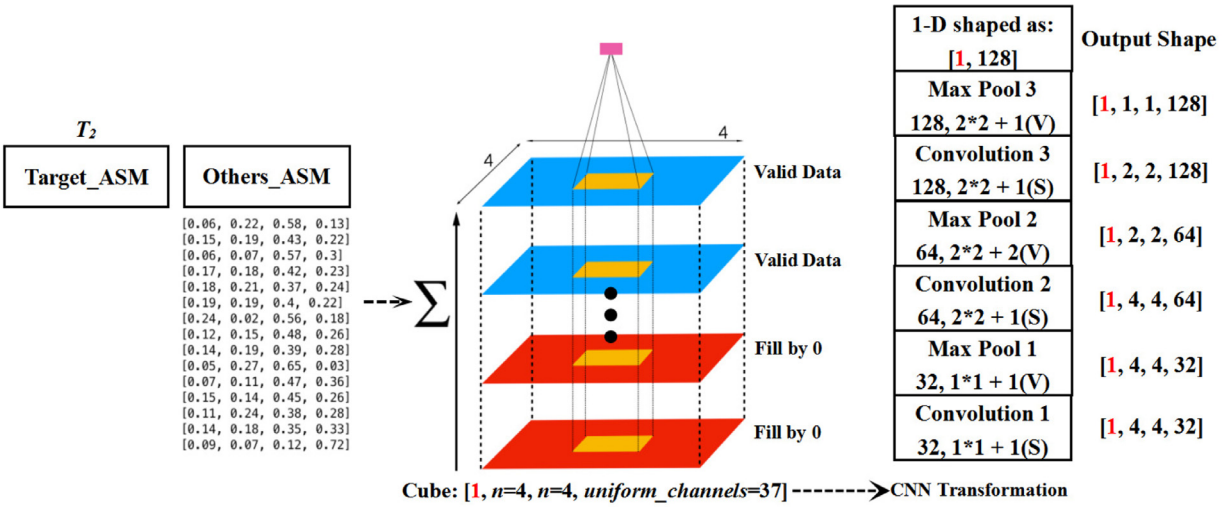


Fig. 6. Use CNN architecture to process the emotion of other participants at time step T_2.

Thus, in each layer, only four valid emotion vectors (denoted as R^4) will be filled in order. What's more, if the cube has extra space, or the emotion vectors from all of the other participants at the current time step are not adequate to fill the whole cube, then multiple "0" will be filled into the remaining positions. In this way, emotion vectors $E_{others\_m\_m+1}$ of all of the other participants between any adjacent target pair (E_{target_m}, E_{target_m+1}) at one time step will be reorganized into a cube shaped as [*height* = *n* = 4, *width* = *n* = 4, *uniform_channels*=37]. Then, the cube will be processed by CNN architecture whose output can be kept as a 1-D series with the proper construction of filters, filters' stride, padding mode in convolution and pooling layer. Furthermore, with the addition of a 4-neuron fully connected layer, the output of CNN can be shaped as [1, *n* = 4]. Such a result can be considered as the fusion of a set of emotion vectors $E_{others\_m\_m+1}$ posted by all of the other participants between time step m and $m+1$.

More importantly, as illustrated in Fig. 6, one set of emotion vectors $E_{others\_m\_m+1}$ posted by all of the other participants can be assembled into a cube with a shape of [*height* = *n*, *width* = *n*, *uniform_channels*]. Obviously, such a cube can be enlarged and shaped as [1, *height* = *n*, *width* = *n*, *uniform_channels*]. In this way, after setting an observation window with size w, multiple w-sized pair-based sequences can be extracted from D'_{target_pair} such as $D'_{target\_pair\_w1}=\{(E_{target\_t}, E_{others\_t\_t+1}), (E_{target\_t+1}, E_{others\_t+1\_t+2}), ..., (E_{target\_t+(w1-1)}, E_{others\_t+(w1-1)\_t+w1})\}$. And, considering all of the $E_{others_m_m+1}$s contained in $D'_{target\_pair\_w1}$, multiple cubes, or $E_{others_m_m+1}$s, can be stacked in the shape of [*w*, *height* = *n*, *width* = *n*, *uniform_channels*]. Following the emotion processing of all of the other participants illustrated in Fig. 6, the output of CNN can be shaped as [1*w, *n*] or [*w*, *n*] when taking into account all of the $E_{others_m_m+1}$s in $D'_{target\_pair\_w1}$ where n is decided by selected Emotions. By such a transformation, the emotion processing of other participants can be set at the window level as well.

In conclusion, when extracting one w-sized pair-based sequence from D'_{target_pair} such as $D'_{target\_pair\_w1}$, all of the E_{target_m}s in D'_{target_pair} can be assembled as [1,*w*,*n*]. After being processed by LSTM, its output can be shaped as [1, *w*, *n*] or [*w*, *n*] by setting the number of neurons in output layer also as n. Moreover, all of the $E_{others_m_m+1}$s can be assembled into stacked cubes as [*w*, *height* = *n*, *width* = *n*, *uniform_channels*] and processed by CNN with the output shaped as [1*w, *n*] or [*w*, *n*]. In other words, within a given observation window with size w, the output of LSTM and CNN at every time step has constituted a one-to-one match. Moreover, as shown in Fig. 7, the computed emotion vectors from LSTM and CNN can be further merged in an ensemble way by a fully connected layer to obtain a final emotion vector predicted at every time step. Such a one-to-one match can facilitate the emotion processing to the "batch" level or, data factorization, which is beneficial to enhancing the overall prediction efficiency of Deep Rolling.

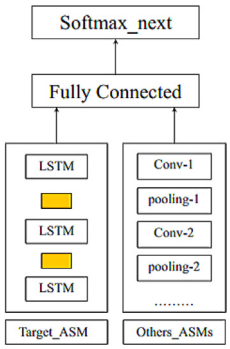


Fig. 7. A single unit of Deep Rolling: Merges emotions of a target and all the other participants at every time step in an ensemble way.

3.5. Data factorization of Deep Rolling

When given a set of selected emotions, a pair-based-sequence D'_{target_pair}, an observation window with size w and a batch of w-sized pair-based sequences, $D'_{target\_pair\_w\ (1\ to\ n)}$ can be extracted from D'_{target_pair}. Here, selected emotions = (emotion$_1$, emotion$_2$, ..., emotion$_n$) $\in R^n$ and D'_{target_pair}={(E_{target_1}, $E_{others\_1\_2}$), (E_{target_2}, $E_{others\_2\_3}$), ..., (E_{target_m-1}, $E_{others\_m-1\_m}$), (E_{target_m}, *None*)}.

In terms of batch-level processing, on the one hand, considering E_{target_m}s in $D'_{target\_pair\_w(1\ to\ n)}$, all the target's emotion vectors can be collected and shaped as [*batch_size, w, n*]. The *batch_size* is the number of extracted w-sized pair-based sequences. On the other hand, when it comes to $E_{others_m-1_m}$s in $D'_{target\_pair\_w(1\ to\ n)}$, all of the vectors of other participants can be assembled in the shape of [*batch_size, w, height = n, width = n, uniform_channels*]. Here *batch_size* is same as the one mentioned above.

More specifically, in terms of the emotion processing for the target participant, all of the emotion vectors will be taken in and output by an LSTM shaped as [*batch_size, w, n*], further reshaped into [*batch_size*w, n*]. Meanwhile, for the emotion processing of other participants, all of the input emotion vectors with a shape of [*batch_size, w, n, n, uniform_channels*] will be reshaped into [*batch_size*w, n, n, uniform_channels*]. Then, as shown in Fig. 6, with the proper construction of filters as well as their strides in the convolution and pooling layer, the output of CNN can be kept as [*batch_size*1*w, n*], or [*batch_size*w, n*] via an n-dimension fully connected layer. In this way, a one-to-one match can be constituted on the entire D'_{target_pair} by two [*batch_size*w, n*] matrix output by LSTM and CNN respectively. Furthermore, by merging the output of both LSTM and CNN via a fully connected layer whose dimension is the same as selected Emotions = (emotion$_1$, emotion$_2$, ..., emotion$_n$) $\in R^n$, the final output of Deep Rolling can be merged and shaped [*batch_size*w, n*].

In addition, as has been mentioned above, the maximum value of every emotion vector has only been occupied by one dimension. Therefore, after placing a softmax operation onto the final output of Deep Rolling, the output of Deep Rolling can be further shrunk to [*batch_size*w*, 1], or [*batch_size, w*] after being reshaped. Such a reshaped result contains the predicted emotion whose index is in the same scale as the annotation step (e.g. $0 \sim n$-1) and every predicted emotion is corresponding to the one of next time step. Finally, when the extraction of a w-sized pair-based sequence moves to the end of D'_{target_pair}={(E_{target_1}, $E_{others\_1\_2}$), (E_{target_2}, $E_{others\_2\_3}$), ..., (E_{target_m-1}, $E_{others\_m-1\_m}$), (E_{target_m}, *None*)}, the emotion of the target participant at time step $m+1$ can be predicted, even if the text content at that time step is still unknown.

Algorithm 2 has concluded all the steps in terms of data factorization of Deep Rolling. As shown in Algorithm 2, after extracting all of the w-sized pair-based sequences, the index-based true labels are organized in the shape of [*batch_size, w*]. Then, all of the emotion vectors concerning the target and other participants will be input into LSTM as well as CNN. Their outputs can be kept in the shape of [*batch_size*w, n*]$_{LSTM}$ and [*batch_size*w, n*]$_{CNN}$ respectively. Afterwards, by merging the output of LSTM and CNN as a one-to-one match via a fully connected layer, the final output of Deep Rolling can be shaped as [*batch_size*w, n*]$_{MERGE}$. Next, a softmax layer has been placed onto [*batch_size*w, n*]$_{MERGE}$ whose output can be further reshaped as [*batch_size, w*]$_{PREDICTION}$. Finally, based on the prediction result ([*batch_size, w*]$_{PREDICTION}$) and true labels ([*batch_size, w*]$_{TRUE}$), Deep Rolling can be fine-tuned via a mean-squared error.

For a more explicit demonstration, an example of tensor flow in terms of data factorization of Deep Rolling is presented in Fig. 8. In such an example, a set of Emotions = (emotion$_1$, emotion$_2$, ..., emotion$_4$) $\in R^4$ has been selected and the window size w is also 4 with stride s = 3. As a result, 135 (batch_size = 135) 4-size pair-based sequences have been extracted from D'_{target_pair}={(E_{target_1}, $E_{others\_1\_2}$), (E_{target_2}, $E_{others\_2\_3}$), ..., (E_{target_m-1}, $E_{others\_m-1\_m}$), (E_{target_m}, None)}. What's more, the max_span computed from the given D' is 639 based on which the uniform_channels is equal to 160.

In this way, in terms of emotion processing of the target participant by LSTM, 135 4-sized pieces with the shape of [1, w = 4, n = 4] have been stacked together as [batch_size = 135, w=4, n = 4]. After being processed by LSTM, the output can also be kept in the shape of [batch_size = 135, w = 4, n = 4] or [540 = batch_size*w, n = 4] after being reshaped. Furthermore, when it comes to the emotion processing of all of the other participants by CNN, after extracting 135 4-sized pair-based sequences from D'_{target_pair}, all of the instances have been organized as [batch_size = 135, w = 4, n = 4,

Algorithm 2 The data factorization for training Deep Rolling.

Input:

window size w;

window stride s;

Selected Emotions=(emotion$_1$,emotion$_2$,..,emotion$_n$) $\in R^n$;

D'=$\{E_1, E_2,\ldots,E_i \mid i\in[Tstart,Tend]\}$;

D'_{target_pair}=$\{(E_{target\_1},E_{others\_1\_2}),..,(E_{target\_m-1},E_{others\_m-1\_m}), (E_{target\_m},None) \mid 0<m\leq i\in[Tstart, Tend]\}$

Output: A fine-tuned Deep Rolling for emotion prediction of target participant

Compute max_span and *uniform_channels* on D'=$\{E_1, E_2,\ldots,E_i \mid i\in[Tstart,Tend]\}$

Extract all the w-size-pair-based sequences from D'_{target_pair} by moving window forward by stride s like:

$D'_{target\_pair\_w1}$=$\{(E_{target\_t},E_{others\_t\_t+1}),(E_{target\_t+1},E_{others\_t+1\_t+2}),..,(E_{target\_t+(w1-1)},E_{others\_t+(w1-1)\_t+w1})\}\rightarrow$ True Labels:[0,1,2... 1]$\in R^w$

$D'_{target\_pair\_w2}$=$\{(E_{target\_t},E_{others\_t\_t+1}),(E_{target\_t+1},E_{others\_t+1\_t+2}),..,(E_{target\_t+(w2-1)},E_{others\_t+(w2-1)\_t+w2})\}\rightarrow$ True Labels:[2,1,3... 2]$\in R^w$

$D'_{target\_pair\_w3}$=$\{(E_{target\_t},E_{others\_t\_t+1}),(E_{target\_t+1},E_{others\_t+1\_t+2}),..,(E_{target\_t+(w3-1)},E_{others\_t+(w3-1)\_t+w3})\}\rightarrow$ True Labels:[1,3,2... 1]$\in R^w$

... .

$D'_{target\_pair\_wn}$=$\{(E_{target\_t},E_{others\_t\_t+1}),(E_{target\_t+1},E_{others\_t+1\_t+2}),..,(E_{target\_t+(wn-1)},E_{others\_t+(wn-1)\_t+wn})\}\rightarrow$ True Labels:[3,0,3... 1]$\in R^w$

Reshape true labels as $[batch\_size, w]_{TRUE}$ where *batch_size* is the number of extracted w-size-pair-based sequences

Collect all the emotion vectors of target participant E_{target_m}s in $D'_{target\_pair\_w\,(1\,to\,n)}$ shaped as [*batch_size*, *w*, *n*]

Input matrix in 14 into LSTM and output the result as [*batch_size*, *w*, *n*]

Reshape matrix in 15 as $[batch\_size^*w, n]_{LSTM}$

Collect all the emotion vectors of other participants $E_{others_m_m+1}$s in $D'_{target\_pair\_w\,(1\,to\,n)}$ as [*batch_size*, *w*, *n*, *n*, *uniform_channels*]

Reshape matrix in 17 as [*batch_size***w*, *n*, *n*, *uniform_channels*]

Input matrix in 18 into CNN and output the result as $[batch\_size^*w, n]_{CNN}$

Merge $[batch\_size^*w, n]_{LSTM}$ and [batch_size**w*, *n*]$_{CNN}$ as $[batch\_size^*w, n]_{MERGE}$ by *n*-dimensional fully connected layer

Conduct softmax on $[batch\_size^*w, n]_{MERGE}$ and output the result as $[batch\_size^*w, 1]_{PREDICTION}$

Reshape matrix in 21 as $[batch\_size, w]_{PREDICTION}$

Conduct model optimization by *LOSS* where $LOSS=\frac{1}{batch\_size*w}\| \|^*|_{PREDICTION} - |^*|_{True} \|_2^2$, here $|^*|$ means [*batch_size*, *w*]

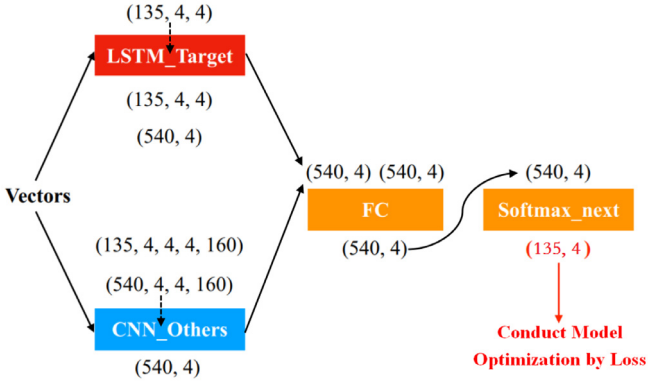


Fig. 8. An example of data factorization for training on Deep Rolling.

n = 4, uniform_channels=160] based on which, via proper data transformation, 540 (batch_size = 135*w = 4) cubes with a shape of [n = 4, n = 4, uniform_channels = 160] have been stacked together as [540 = batch_size*w, n = 4, n = 4, uniform_channels = 160]. Meanwhile, the result of CNN is also shaped as [540 = batch_size*w, n = 4]. Finally, outputs shaped as [540 = batch_size*w, n = 4] from both LSTM and CNN have been merged via a 4-neuron fully connected layer whose output is normalized by softmax and shaped from [540 = batch_size*w, 1] to [batch_size = 135, w = 4].

Finally, the data factorization processing as shown in Algorithm 2 can be transferred to the prediction phase of a fine-tuned Deep Rolling. As shown in Fig. 9, by training extracted w-sized pair-based sequences, parameters of Deep Rolling will be fine-tuned and the emotion of target participant at time step $m+1$ will be predicted. Then, a new predicted pair (E_{target_m+1},None) of target participant will be concatenated to the end of D'_{target_pair} as $\{(E_{target\_1}, E_{others\_1\_2}), (E_{target\_2}, E_{others\_2\_3}), ..., (E_{target\_m-1}, E_{others\_m-1\_m}), (E_{target\_m}, None), (E_{target\_m+1}, None)\}$ based on which another set of w-sized pair-

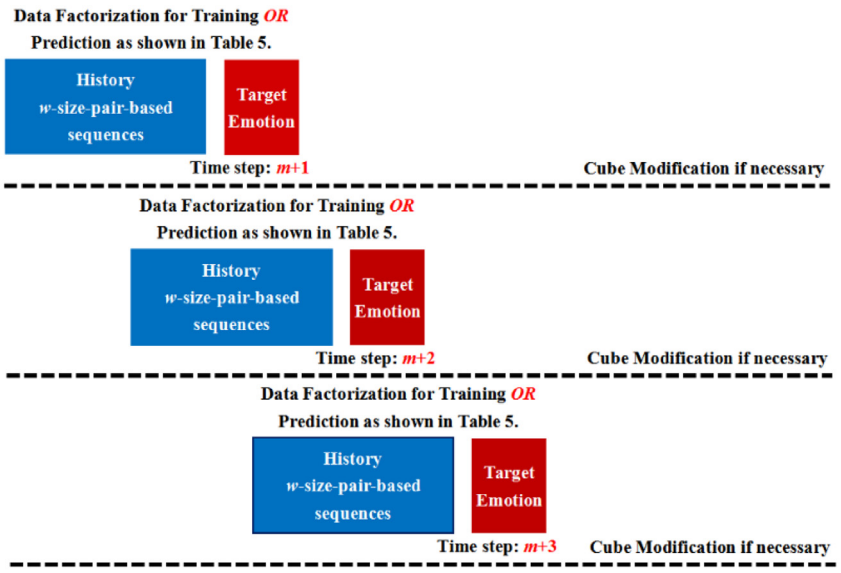


Fig. 9. The data factorization for a prediction in terms of Deep Rolling.

Table 4
The selected datasets for conducting experiments on Deep Rolling.

No.	Name	Area	Nomination	Prediction target	Duration	#Total lines	#Target lines
D1	Forrest **Gump**	U.S.	Oscar 67	Forrest Gump	142 min	2054	905
D2	**Shawshank** Redemption	U.S.	Oscar 67	Ellis Boyd Redding	142 min	1729	590
D3	**Scent** a of Woman	U.S.	Oscar 65	Lieutenant Frank	157 min	2226	1165
D4	I Am Not Madame **Bovary**	China	HK Film Awards 2017	Li Xue Lian	138 min	2323	505

based sequences will be extracted and assembled in the way of data factorization as shown in Algorithm 2. Worthwhile to mention is that, in a real multi-participant communication context, if the newly experienced emotion at time point $m+i$ has changed the value of max_span, or the current cube is not big enough to store other participants' emotion vectors, then the modification of cube size is indispensable to be conducted. Consequently, Deep Rolling is needed to be fine-tuned again according to the current extraction of w-sized pair-based sequences to update parameters of CNN component. Following above routine, by concatenating predicted emotions, more emotions at future time point can be predicted.

4. Experiments

In this section, we conduct a series of experiments on Deep Rolling. Firstly, in Section 4.3, by selecting a target participant with the largest amount of posted texts, we compare Deep Rolling with other classical and state-of-the-art text-based emotion prediction and time series prediction methods. For the first experiment, we add emotions from all of the other participants at every time step. The performance will be evaluated by accuracy and F1-Score for precision. Meanwhile, time cost (ms) will be used to evaluate efficiency to obtain the above two metrics. Second, in Section 4.4, we have compared Deep Rolling itself with and without inputting the emotions of other participants. Specifically, with the same target participant, we gradually add emotions vectors of ***only*** 0 (without any other participants), 1, 2, 3, 4, 5, ..., m, ***ALL*** (with all of the other participants) other participants into the CNN component of Deep Rolling. Afterwards, the effect of other participants on the target participant's emotion prediction will be analyzed. Third, in Section 4.5, we have additionally selected another four target participants; in every corresponding scenario, all of the remaining participants will be considered as other participants and input to CNN component. Then evaluation metrics (accuracy, F1-Score and time cost (ms)) will be computed on different target participants. All of the experiments elaborated in this section are conducted in the environment of MacBook Pro-(Retina, 13 inch, Late 2013) with a CPU of 2.4 GHz Intel Core i5 and an internal memory of 8GB 1600 MHz DDR3.

4.1. Data preparation

In order to reveal a multi-participant communication context, in this paper, we have collected subtitles of 4 movies in time order. The specific information of 4 collected datasets has been listed in Table 4. The next step as shown in Fig 2 is annotation. Particularly, for D1 to D3, we annotate every line as [target_flag, Emotion] in 6 types of emotions, while as shown in Table 4, since the amount of texts posted by target participant in D4 is limited, therefore only 4 types of emotions

Table 5
The emotion distribution after annotation.

No.	Name	Type	Happy(%)	Sad(%)	Angry(%)	Anxious(%)	Surprise(%)	Other(%)
D1	Gump.	ALL	40.70	17.96	10.81	18.16	1.36	11.00
D2	Shawshank.		42.34	12.78	21.75	12.67	0.29	10.18
D3	Scent.		31.18	12.8	28.26	18.28	1.08	8.40
D4	Bovary.		13.78	6.03	43.65	*None*	*None*	36.55
No.	Name	Type	Happy(%)	Sad(%)	Angry(%)	Anxious(%)	Surprise(%)	Other(%)
D1	Gump.	TARGET	47.29	22.54	0.88	17.35	0.55	11.38
D2	Shawshank.		58.31	19.83	5.93	7.97	0.51	7.46
D3	Scent.		35.45	18.03	29.18	4.03	0.34	12.96
D4	Bovary.		9.11	7.33	74.06	*None*	*None*	9.50

Table 6
The global emotion interaction pattern (*EIP*) on every dataset (D1 = Gump, D2 = Shawshank, D3 = Scent, D4 = Bovary).

Datasets	***Emotions***	Happy	Sad	Angry	Anxious	Surprise	Others
	Happy	***0.876***	0.030	0.014	0.056	0.007	0.017
	Sad	0.068	***0.843***	0.030	0.033	0.010	0.016
D1	Angry	0.052	0.050	***0.725***	0.119	0.016	0.038
	Anxious	0.126	0.043	0.071	***0.718***	0.012	0.030
	Surprise	0.214	0.100	0.025	0.161	***0.393***	0.107
	Others	0.064	0.027	0.038	0.048	0.013	***0.810***
	Happy	***0.871***	0.02	0.037	0.037	0.001	0.034
	Sad	0.065	***0.814***	0.075	0.025	0.005	0.016
D2	Angry	0.072	0.044	***0.750***	0.0098	0.004	0.032
	Anxious	0.123	0.025	0.169	***0.607***	0.001	0.075
	Surprise	0.167	0.167	0.150	0.100	***0.333***	0.083
	Others	0.143	0.020	0.069	0.094	0.003	***0.671***
	Happy	***0.822***	0.038	0.040	0.080	0.008	0.012
	Sad	0.091	***0.719***	0.072	0.088	0.005	0.025
D3	Angry	0.045	0.033	***0.770***	0.117	0.008	0.027
	Anxious	0.136	0.061	0.181	***0.558***	0.006	0.058
	Surprise	0.229	0.063	0.208	0.104	***0.292***	0.104
	Others	0.043	0.037	0.091	0.126	0.013	***0.690***
	Happy	***0.78***	0.02	0.07	*None*	*None*	0.13
	Sad	0.05	***0.69***	0.12	*None*	*None*	0.14
D4	Angry	0.02	0.02	***0.85***	*None*	*None*	0.11
	Anxious	*None*	*None*	*None*	*None*	*None*	*None*
	Surprise	*None*	*None*	*None*	*None*	*None*	*None*
	Others	0.05	0.02	0.13	*None*	*None*	***0.8***

have been considered. The emotion distribution of above four datasets has been presented in Table 5. In Table 5, type "*ALL*" represents the emotion distribution on the whole text set while type "TARGET" means emotion distribution on texts only concerning target participant. Moreover, as shown in Table 5, the ratio of "Anxious" and "Surprise" in D4 is None. The annotated emotion index based series of above datasets has been stored in order to provide true labels when training Deep Rolling.

Afterwards, following the steps listed in Fig. 2, the global emotion interaction pattern (EIP) will be computed on 4 datasets respectively. As listed in Table 6, it is obvious that the maximum value of an emotion vector is occupied by only one dimension, corresponding to a certain type of emotion.

Then, all the texts in the above 4 datasets will be segmented for word embedding. In this paper, we utilize Jieba,[1] a famous tool for natural language processing, to conduct word segmentation. In addition, Tensorflow [64], along with the Skip-2gram [62] model has been adopted to obtain the word embedding. As a result, every text in the above 4 datasets will be converted into a 256-dimension word vector sequence. Furthermore, the Bi-LSTM [20] based emotion classifier will be trained to obtain the initial emotion vectors and those which are incorrectly classified will be substituted with an emotion vector in global EIP (as shown in Table 6). In this way, the above 4 datasets will be converted into emotion vector-based sequences, denoted as D1', D2', D3' and D4', where $D'=\{E_1, E_2, ...,E_i|i \in [Tstart,Tend]\}$. More specifically, for D1' to D3', every emotion vector E_i has 6 dimensions and only 4 dimensions in terms of E_i in D4'.

[1] https://pypi.org/project/jieba/.

Table 7
The brief architecture of Deep Rolling for different datasets implemented by Tensorflow 1.4.

*D*1 to *D*3 on Deep Rolling				*D*4 on Deep Rolling			
CNN architecture		LSTM architecture		CNN architecture		LSTM architecture	
Conv_1	12, 1*1+1 (S)	FC	24 neurons	Conv_1	32, 1*1+1(S)	FC	64 neurons
Pooling_1	12, 1*1+1 (V)	Layer_1	24 neurons, *relu*	Pooling_1	32, 1*1+1(V)	Layer_1	64 neurons, *relu*
Conv_2	24, 2*2+1 (S)	Dropout	90% output	Conv_2	64, 2*2+1(S)	Dropout	90% output
Pooling_2	24, 2*2+2 (V)	Layer_2	12 neurons, *relu*	Pooling_2	64, 2*2+2(V)	Layer_2	8 neurons, *relu*
Conv_3	30, 2*2+1 (S)	Dropout	90% output	Conv_3	128, 2*2+1(S)	Dropout	90% output
Pooling_3	30, 3*3+1 (V)	Layer_3	6 neurons, *relu*	Pooling_3	128, 2*2+1(V)	Layer_3	4 neurons, *relu*
FC_1	24 neurons	Dropout	90% output	FC_1	64 neurons	Dropout	90% output
FC_2	12 neurons			FC_2	32 neurons		
FC_3	6 neurons			FC_3	4 neurons		
FC: 6 neurons *softmax*				FC: 4 neurons *softmax*			

4.2. Experimental setting

First, we have implemented the proposed model (Deep Rolling) via Tensorflow with the configuration illustrated in Table 7. For the convolution layer, "12, 1*1 + 1 (S)" represents 12 filters with size 1*1 moving by stride 1, in the same (S) or valid (V) padding mode. The keeping probability of output has been set as 90% for dropout between each LSTM layer. Moreover, the Adagrad optimizer [64] has been adopted to fine-tune the proposed model. Further, the most optimal "epoch" values are: 250 on D1 and D2, 400 on D3 and D4. In addition, the learning rate has been selected by a grid search with the result of 0.03 on D1 and D3, 0.06 on D2, 0.001 on D4. Finally, with the structure shown in Table 7, Deep Rolling will be implemented via data factorization (as illustrated in Algorithm 2).

Second, for comparison, we have adopted several classical regression models for time series prediction including Decision Tree Regression (DT_R) [46], Linear Regression (LR) [15], SVM Regression (SVM_R) [10], KNN Regression (KNN_R) [26] and Random Forest Regression (RF_R) [5]. Specifically, the depth of the decision tree in DT_R has been set as "*auto*"; the penalty parameter $\boldsymbol{C}$ and the error ε in SVM_R have been set as 1 and 0.1 respectively, with '*rbf*' kernel; the number of neighbors of KNN_R has been set as 5 on D1, D2 and D3, 3 on D4; 10 trees have been incorporated in RF_R. All of the above models have been implemented by the Scikit-learn[2] framework. Such a framework has optimized all of the classical regression models automatically with the above parameter setting.

Moreover, several state-of-the-art methods have also been adopted for comparison. Specifically, DDAE [43] has been constructed by a three-hidden-layer DNN with "relu" activation and pre-trained by a two-layer auto-encoder in an unsupervised way. REE [24] has also contained CNN and multi-layer LSTM components to predict emotion reactions to Facebook posts in an ensemble way. DA_RNN [39] consists of an encoder and a decoder with a dual-stage attention mechanism for time series processing and two context vectors for decoding have been computed. R2N2 [17] has firstly processed data via a multi-layer LSTM based on which the residual error has been computed and "tanh" activation function has been used for neurons in each LSTM layer. Finally, CNN_Dilation [7] has experienced 3 convolution layers with a dilation rate of 1, 2 and 4, respectively, and its receptive field is set as 8. All of the above state-of-the-art models are implemented by Tensorflow [1].

More specifically, in terms of DDAE, the number of neurons in the hidden and output layers of DNN has been set as 384-192-96-6 for D1, D2 and D3; 256-128-64-4 for D4. The learning rate is 0.9 on D1, D2 and D3; 0.3 on D4 with 300 epoches. What's more, REE has a two-layer LSTM and the number of neurons in each layer is the same as that of emotion types in different datasets. In addition, the CNN component of REE has been equipped with 2 convolution and max pooling layers. The width of filters has been set as 3 on D1, D2 and D3; 2 on D4 with filters' height setting as 6 on D1, D2 and D3; 4 on D4. The learning rate has been set as 0.01 on D1 and D2, 0.3 on D3, 0.001 on D4. Moreover, DA_RNN has set the length of the encoder according to input and the length of the decoder's output is the same as the number of emotions types in different datasets. Meanwhile, the Bahdanau attention [50] mechanism has been used. Next, R2N2 has firstly computed the estimation of emotion pointing to the next time step via KF [8] with a lag of 2. Based on the first estimation, then residual error has been obtained by two-layer LSTM. The number of neurons is the same as that of emotion types in different datasets. In addition, the learning rate has been set as 0.07 on D1 and D4, 0.09 on D2 and D3 with 65 epoches. Last but not the least, CNN_Dilation has been implemented by taking in a long 1-D series, which means that the input emotion vectors have been catenated as a long sequence by "row" axis. Finally, the Adagrad optimizer [1] has also been used for the optimization of DDAE, REE, DA_RNN, R2N2 and CNN_Dilation.

Other important details include: (1) All of the classical methods only accept emotion index-based sequence. In this way, true label, or emotion index, can be provided by the annotation step. Meanwhile, instances for training can also be extracted as index-based sequences from every text's emotion vector according to the maximum value in corresponding dimension. (2) All of the state-of-the-art methods take an emotion vector-based series. The emotion vectors are provided by the output

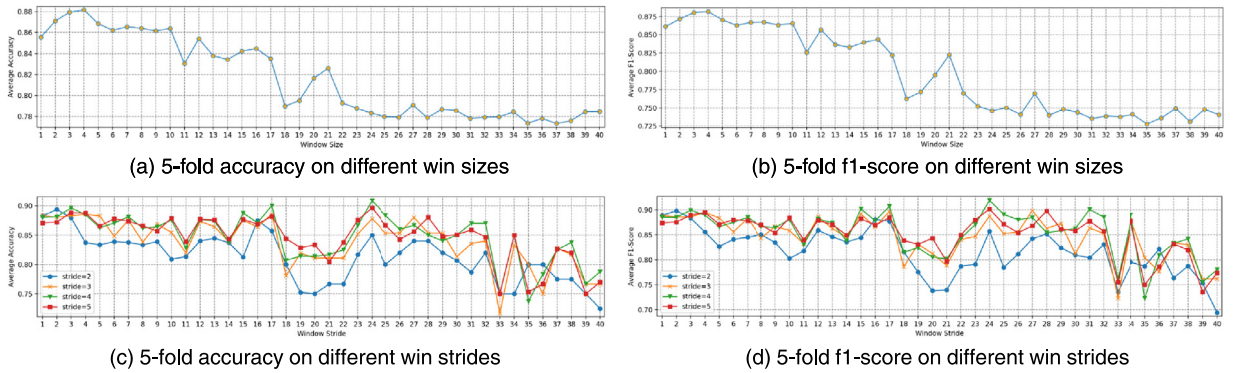
[2] https://scikit-learn.org/stable/.

Table 8
The accuracy and f1-score by the most optimal window size and stride of all the adopted methods.

Dataset	Methods	win_size	win_stride	Accuracy	F1-Score	Time cost (ms)
D1	DT_R	28	22	0.824107	0.818886	1.009473
	LR	28	22	0.824107	0.819968	0.919124
	SVM_R	44	28	0.824194	0.822127	3.728711
	KNN_R	37	1	0.783154	0.778958	588.375977
	RF_R	22	20	0.830976	0.826075	7.022135
	DDAE	58	41	0.834483	0.829076	0.004772
	REE	25	5	0.793122	0.764111	111.231311
	DA_RNN	23	1	0.780848	0.774153	12915.61279
	R2N2	12	9	**0.836690**	**0.832319**	25.42421
	CNN_Dilation	2	2	0.835897	0.823660	141.690461
	Deep Rolling	**2**	**1**	**0.837778**	**0.834199**	42.626693
	Improvement (%)			+0.13%	+0.23	
D2	DT_R	8	5	0.819792	0.818350	0.815788
	LR	10	8	0.810967	0.805660	0.843864
	SVM_R	8	5	0.831250	0.830478	1.840088
	KNN_R	37	25	0.831351	0.816665	2.749528
	RF_R	14	10	0.799603	0.793002	8.32679
	DDAE	33	19	0.751064	0.676932	0.006216
	REE	38	33	0.780627	0.787936	39.904229
	DA_RNN	9	1	0.709684	0.700783	1902.978992
	R2N2	7	3	**0.837596**	**0.830690**	23.530976
	CNN_Dilation	4	2	0.762712	0.719408	232.465386
	Deep Rolling	**3**	**3**	**0.837778**	**0.836197**	40.372754
	Improvement (%)			+0.02	+0.66	
D3	DT_R	3	2	0.821652	0.814601	1.028092
	LR	3	2	0.811736	0.805402	1.422624
	SVM_R	5	2	0.851034	**0.849540**	7.776709
	KNN_R	31	1	0.832954	0.817952	555.3486
	RF_R	11	6	0.855789	0.847655	8.34821
	DDAE	3	2	0.770115	0.714166	0.004413
	REE	31	31	0.835665	0.811109	39.66678
	DA_RNN	13	2	0.685516	0.678652	3421.798897
	R2N2	54	38	0.856667	0.845873	19.577201
	CNN_Dilation	4	3	**0.858083**	0.838417	129.559072
	Deep Rolling	**3**	**2**	**0.860510**	**0.850635**	48.475358
	Improvement (%)			+0.28	+0.13	
D4	DT_R	34	26	0.833824	0.830318	0.859391
	LR	54	34	0.848148	0.845341	1.541341
	SVM_R	5	1	0.886531	0.886028	5.491634
	KNN_R	43	23	**0.891860**	0.881179	2.63431
	RF_R	52	34	0.844017	0.835784	7.828369
	DDAE	53	34	0.866187	0.855451	0.004211
	REE	26	2	0.809082	0.741626	132.577562
	DA_RNN	6	2	0.871528	0.864735	595.604229
	R2N2	5	5	0.887500	**0.889545**	13.721418
	CNN_Dilation	29	24	0.850000	0.888095	725.085449
	Deep Rolling	**4**	**3**	**0.893088**	**0.896511**	42.917757
	Improvement (%)			+0.14	+0.78	

of *Affective Space Mapping*. (3) An observation window with size w and stride s has been introduced into all the methods for comparison [5,7,10,15,17,24,26,39,43,46]. (4) To our best knowledge, no method concerning emotion prediction or time series forecasting has incorporated the emotions of other participants. Therefore, only the proposed model, Deep Rolling, contains emotion vectors of both the target and other participants. (5) All of the following experiments are conducted in 5-fold validation with multiple times and the results presented in Tables 8, 10, 12 and 13, are the average ones. Moreover, in terms of evaluation metrics, accuracy and F1-Score have been used for emotion prediction precision. Meanwhile, time cost in millisecond (ms) has been adopted for emotion prediction efficiency. The time cost represents the duration to obtain accuracy and F1-Score without considering the time for assembling data.

In order to select an optimal value for window size w and stride s, grid search has been adopted for all of the methods mentioned above. Taking Deep Rolling on D4 as an example, as shown in Fig. 10, first, stride s has been set to 1 with a set of w ranging from 1 to 40 and 5-fold validation on D4 has been conducted by multiple times. As a result, optimal window sizes such as [2,3,4,5] have been selected by accuracy and F1-Score, as shown in Fig. 10 (a) and (b). Second, as shown in Fig. 10 (c) and (d), based on the window size [2,3,4,5], different strides in the range of 1 to 40 have been compared (still via 5-fold validation by multiple times). In this way, candidate combinations such as w 2-s [1,2,3], w 3-s [3,4,5,17], w 4-s



(a) 5-fold accuracy on different win sizes

(b) 5-fold f1-score on different win sizes

(c) 5-fold accuracy on different win strides

(d) 5-fold f1-score on different win strides

Fig. 10. Conduct grid search on window size and stride for Deep Rolling with Dataset 4: (a) and (b) 5-fold accuracy and f1-score on different window sizes; (c) and (d) 5-fold accuracy and f1-score on different window strides.

[2,3,4,17,24] and w 5-s [3,4,24] have been selected. Finally, an extra round of 5-fold validation by multiple times has been conducted on the above combinations. Consequently, a most optimal pair of window w and stride s can be selected.

4.3. Precision and efficiency comparison for emotion prediction on different methods

In this section, we analyze the experimental result of comparing Deep Rolling with other classical and state-of-the-art methods on the precision and efficiency of target participant's emotion prediction. Moreover, in terms of the target participant, we use the role listed in Table 4 and the amount of their lines are the largest in the corresponding dataset. Meanwhile, for this experiment, we input emotions of all of the other participants into the CNN component of Deep Rolling at every time step. Finally, the experiment has been conducted by 5-fold validation with multiple times and the most optimal combination of window size w and stride s has been selected following the approach illustrated in Fig. 10. The average accuracy and F1-Score, as well as time cost (ms) have been listed in Table 8. The time cost represents the duration to obtain accuracy and F1-Score, which are computed together without considering the time for assembling data.

Second, as shown in Table 8, compared with all of the classical and state-of-the-art methods, Deep Rolling has achieved the best accuracy and F1-score on all datasets (which has been highlighted in **bold** font). Moreover, the highest accuracy and F1-Score of compared methods has been denoted in underlined **font**. And, the improvement of accuracy and F1-Score between Deep Rolling (denoted in **bold**) and the best compared method (denoted in **font**) has been computed on every dataset. It can be clearly observed in Table 8 that, Deep Rolling has indeed improved the prediction precision compared with classical and state-of-the-art methods.

However, from the other aspect, it can be discovered that, on D2 and D4, enhancement of accuracy is much lower than that of the F1-Score, while the accuracy and F1-Score enhancement on D1 and D3 is more balanced. The reason why such phenomenon has occurred is due to the effect of other participants. More specifically, considering the amount of target and total lines presented in Table 4, it can be discovered that, on D2 and D4, the amount of texts posted by the target participant is much more limited than that of the total lines. But, on D1 and D3, there are more target lines. As illustrated in Fig. 5, other participants' emotion vectors will only be considered when they are located between a pair of adjacent target participants. In this way, for D1 and D3, when the target participant has posted much more texts on the entire movie, then the emotion vectors of other participants, located between a pair of adjacent target participants, are more related to the current communication context. But for D2 and D4, since the target participant has posted fewer lines on the entire movie, as a result, a pair of adjacent target participants can be separated by a longer distance. Consequently, more emotion vectors of other participants will be collected between a pair of adjacent target participants. In this way, after fusing the emotion vectors of other participants by the CNN component in Deep Rolling, the output is weakly linked to current communication context. Consequently, compared with D1 and D3, the improvement of accuracy and F1-Score on D2 and D4 is less balanced.

Third, when it comes to prediction efficiency as shown in Table 8, due to the data factorization presented in Algorithm 2, even though Deep Rolling has incorporated the extra emotion vectors of all of the other participants at every time step, the time cost has still been controlled at an acceptable level. Moreover, KNN_R, REE, DA_RNN and CNN_Dilation have the worst efficiency compared to Deep Rolling. Reasonable explanations are that, first, KNN_R does not have separated phases for training and prediction. Therefore, its time cost has been directly linked with data quality. Then, since the CNN component of REE has taken every single text as a sequence of vectors or a large, two-dimensional matrix in terms of the whole text set, the efficiency has deteriorated on the first and third movie. In addition, DA_RNN has experienced two-stage attention on the sequence-to-sequence structure; as a result, the input is difficult to be factorized, especially for computing two context vectors. Finally, CNN_Dilation has imported the dilation rate and organized the whole input as a long text sequence followed by head-to-end scanning. All of the sequences are processed one by one instead of in batch. Therefore, its duration for predictions is much longer than that of Deep Rolling.

Table 9
A subset of all of the other participants in four movies.

D1		D2		D3		D4	
Prediction target:		Prediction target:		Prediction target:		Prediction target:	
Forrest Gump (905) - 1		Ellis Boyd 'Red' (590) - 2		Lieutenant Frank (1165) - 7		Li Xue Lian (505) - 3	
Other participant subset							
ID	Name (Amount)	ID	Name (Amount)	ID	Name (Amount)	ID	Name (Amount)
11	*Some "John Doe"s* (439)	1	Andy Dufresne (352)	5	Charlie Simms (428)	12	*Some "John Doe"s (439)*
3	Lieutenant Dan (247)	10	*Some "John Doe"s* (205)	3	Mr. Trask (135)	4	Wang Gong Dao (243)
2	Jenny Curran (192)	4	Warden Norton (174)	2	George Willis (109)	22	Mayor Ma (231)
5	Bubba Blue (83)	7	Brooks Hatlen (85)	1	Harry Havemeyer (65)	21	Zheng Zhong (221)
4	Mrs. Gump (74)	3	Heywood (81)	12	*Some "John Doe"s* (64)	18	Zhao Da Tou (194)
13	Drill Sergeant (17)	5	Captain Hadley (77)	9	Randy (62)	1	Aside (110)
7	Young Forrest (15)	6	Tommy (68)	6	Mrs. Rossi (57)	20	Leading Cadre (97)
12	Principal of 7. (15)	9	Solicitor (48)	13	Freddie - sell Ferrari (28)	6	Zhao Yi (85)
9	Elvis Presley (14)	8	Bogs Diamond (26)	11	Donna (25)		
		12	Floyd (14)				

Finally, from the perspective of Deep Rolling itself, it can be observed from Table 8 that, on the one hand, the value of the selected optimal window size w of Deep Rolling is not higher than that of the compared methods, which means that the input cost for the predictions in Deep Rolling is still maintained at an acceptable level. On the other hand, the selected optimal window stride s of Deep Rolling is not lower than half of the corresponding window size, specifically, $1/2 = 50\%$, $3/3 = 100\%$, $2/3 \approx 66.7\%$ and $3/4 = 75\%$ on four datasets respectively.

In conclusion, by comparing Deep Rolling with classical and state-of-the-art methods on the precision and efficiency of predicting emotions for a target participant, it can be observed that Deep Rolling has achieved the best accuracy and F1-score on all datasets and, due to data factorization, still maintained the time cost at an acceptable level. However, there are still two questions needed to be answered: (1) How do the emotion vectors of other participants affect the target participant's emotion prediction? (2) Can Deep Rolling still achieve precision improvements when selecting different target participants? Such two questions will be analyzed in Section 4.4 and 4.5 respectively.

*4.4. Deep rolling analysis **with** and **without** other participants*

In this section, we particularly analyze Deep Rolling with and without inputting emotion vectors from the other participants. First, as shown in Table 9, we have additionally selected a subset of other participant candidates while maintaining the previous target participant. Moreover, there are many supporting roles in every movie and the number of their lines is very small. Therefore, we have combined some supporting roles, denoted as *Some "John Doe"s* in Table 9, processing as a single role.

Second, for Deep Rolling, keeping feeding emotion vector of the target participant to the LSTM component at every time step, we add the emotion vectors of ***only*** 0, 1, 2, 3, 4, 5,..., *m, ALL* other participants to the CNN component in Deep Rolling. Here, case 0 represents the fact that no emotion vector of other participants has been added. In this situation, Deep Rolling has degenerated into a simple LSTM model. In addition, case *ALL* means feeding emotion vectors of all the other participants in the entire movie, as the experiment shown in Table 8.

More importantly, for case 2,3,4...,*m* it means that we have ***only*** added the emotion vectors of 2,3,4...,*m* other participants to Deep Rolling. Taking case 2 on D1 as an example, at the beginning, we have the whole ID set established as $\boldsymbol{C}$={11,3,2,5,4,13,7,12,9}, as shown in Table 9. Then, all of the subsets with ***only*** two elements will be extracted from $\boldsymbol{C}$ such as {11,3}, {11,2}... {3,2}, {3,5}... {7,12}, {7,9}, etc. Apparently, a large number of subsets will be extracted. Next, for every two-element subset, the emotion vectors of corresponding other participants will be added to Deep Rolling. Moreover, in order to reveal the precision tendency when adding the emotion vectors of different amounts of other participants, also considering the page limit due to a large amount of extracted subsets, we ***only*** keep a two-element subset–which has helped Deep Rolling to achieve the highest performance in comparison to its counterparts. In this way, we can further observe Deep Rolling with different amounts of other participants at the corresponding best level for case 0, 1, 2, 3, 4, 5,.., *m, ALL* respectively. Following this routine, case 3 means a best three-element subset of other participants, which has helped Deep Rolling to achieve the highest performance among its counterparts.

Based on the above principle, the specific performance when feeding different amounts of other participants into Deep Rolling is presented in Table 10. In Table 10, we re-use $w = 2$, $s = 1$ on D1; $w = 3$, $s = 3$ on D2; $w = 3$, $s = 2$ on D3 and $w = 4$, $s = 3$ on D4. And, we have also reused the performance for case *ALL*, as presented in Table 8. What's more, in Table 10, we denote the highest performance with **bold** font on every dataset. Meanwhile, the performance of Deep Rolling without inputting any other participants, or case 0, has been underlined as **font**. Based on these two values, the improvement of Deep Rolling, with and without other participants, has been computed. More specifically, Fig. 11 has visualized corresponding performance as presented in Table 10.

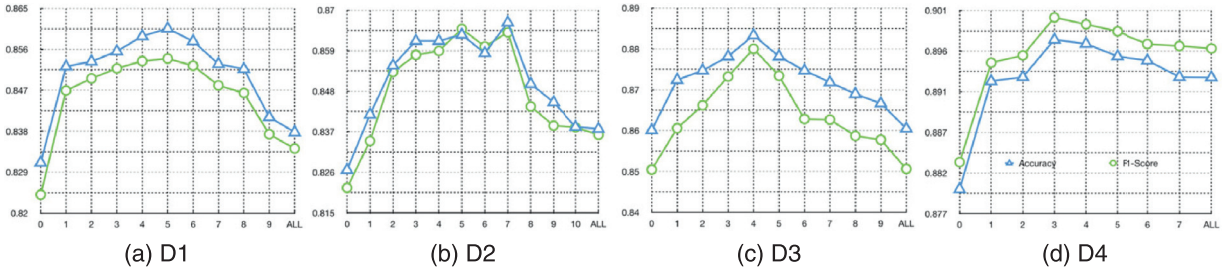


Fig. 11. The visualized tendency of precision metrics corresponding to Table 10.

Table 10
Deep Rolling analysis: The prediction precision of Deep Rolling with or without other participants.

Dataset	Other participant combination (Outperform counterparts)	#Other participants	#Post	Accuracy	F1-score	Time cost (ms)
D1	*None (Do not feed any other participants)*	0	0	**0.831111**	**0.824071**	41.950570
	11	1	439	0.852222	0.846947	41.931771
	11, 7	2	454	0.853333	0.849594	42.528776
	11, 2, 12	3	646	0.855556	0.851809	43.954883
	11, 3, 7, 9	4	715	0.858889	0.85342	42.031283
	11, 2, 13, 12, 9	**5**	**677**	**0.860556**	**0.853968**	41.385498
	11, 5, 13, 7, 12, 9	6	583	0.857778	0.852389	44.296257
	11, 3, 2, 5, 4, 7, 12	7	1,065	0.852778	0.848071	41.811751
	11, 2, 5, 4, 13, 7, 12, 9	8	849	0.851667	0.846394	41.796077
	11, 3, 2, 5, 4, 13, 7, 12, 9	9	1,096	0.841111	0.837335	41.767611
	ALL (Feed all the other participants)	16	1,149	0.837778	0.834199	42.626693
	Improvement between case 5 and 0 (%)			+3.54	+3.63	
D2	*None (Do not feed any other participants)*	0	0	**0.826667**	**0.821798**	39.195671
	7	1	85	0.841667	0.834527	45.696859
	4, 5	2	251	0.855000	0.853244	39.171875
	1, 7, 5	3	514	0.861667	0.857873	38.175863
	1, 10, 4, 12	4	745	0.861667	0.858954	38.843669
	1, 10, 7, 5, 9	**5**	**767**	0.863333	**0.864943**	43.019417
	1, 10, 5, 6, 9, 8	6	776	0.858333	0.860127	42.483073
	1, 4, 5, 6, 9, 8, 12	**7**	**759**	**0.866667**	0.863965	44.126090
	1, 10, 7, 3, 6, 9, 8, 12	8	876	0.850000	0.843888	40.251839
	10, 4, 7, 3, 5, 6, 9, 8, 12	9	778	0.845000	0.838656	38.685221
	1, 10, 4, 7, 3, 5, 6, 9, 8, 12	10	1,130	0.838333	0.838196	38.145247
	ALL (Feed all the other participants)	11	1,729	0.837778	0.836197	40.372754
	Improvement between case 7, 5 and 0 (%)			+4.84	+5.25	
D3	*None (Do not feed any other participants)*	0	0	**0.860111**	**0.850466**	43.084766
	9	1	62	0.872414	0.860587	44.561572
	9, 11	2	87	0.874713	0.866201	42.757617
	9, 13, 11	3	115	0.878161	0.873234	42.013770
	2, 12, 6, 11	**4**	**255**	**0.883333**	**0.880011**	41.936230
	2, 1, 9, 6, 13	5	321	0.878161	0.873433	42.481836
	3, 2, 12, 9, 13, 11	6	423	0.874713	0.862834	41.857520
	3, 2, 1, 12, 9, 13, 11	7	488	0.871839	0.862678	42.136426
	5, 3, 1, 12, 9, 6, 13, 11	8	864	0.868966	0.858712	42.397217
	5, 3, 2, 1, 12, 9, 6, 13, 11	9	973	0.866667	0.857774	41.632666
	ALL (Feed all the other participants)	16	2,226	0.860510	0.850635	48.475358
	Improvement between case 4 and 0 (%)			+2.70	+3.47	
D4	*None (Do not feed any other participants)*	0	0	**0.879902**	**0.883093**	43.384027
	22	1	231	0.892647	0.894828	43.209410
	22, 1	2	341	0.893137	0.895694	47.341061
	22, 21, 20	**3**	**549**	**0.897549**	**0.900176**	43.077183
	22, 21, 12, 1	4	809	0.897059	0.899366	44.761753
	22, 21, 20, 12, 1	5	906	0.895588	0.898533	46.774769
	22, 21, 20, 18, 12, 1	6	1,100	0.895098	0.897032	43.485165
	22, 21, 18, 12, 6, 4, 1	7	1,331	0.893137	0.896806	42.849779
	ALL (Feed all the other participants)	21	1,818	0.893088	0.896511	42.917757
	Improvement between case 3 and 0 (%)			+2.01	+1.93	

Table 11
Different target participants analysis on D1 to D4 (Target name (post amount) and Corresponding ID).

	Target 1 - ID	Target 2 - ID	Target 3 - ID	Target 4 - ID	Target 5 - ID
D1	Forrest Gump (905) - 1	Lieutenant Dan (247) - 3	Jenny Curran (192) - 2	Bubba Blue (83) - 5	Mrs. Gump (74) - 4
D2	Ellis Boyd 'Red' (590) - 2	Andy Dufresne (352) - 1	Warden Norton (174) - 4	Brooks Hatlen (85) - 7	Heywood (81) - 3
D3	Lieutenant Frank (1165) - 7	Charlie Simms (428) - 5	Mr. Trask (135) - 3	George Willis (109) - 2	Harry Havemeyer (65) - 1
D4	Li Xue Lian (505) - 3	Wang Gong Dao (243) - 4	Mayor Ma (231) - 22	Zheng Zhong (221) - 21	Zhao Da Tou (194) - 18

Apparently, it can be observed from Table 10 that no matter how many other participants have been added–for example, case 1, 2, 3, 4, 5,., *m, ALL*–Deep Rolling can always achieve a better performance than case *None* or case 0. In other words, by adding the emotion vectors of other participants to Deep Rolling, the performance on the target's emotion prediction has indeed been enhanced in comparison to a model without considering the emotion vectors of other participants.

However, it can also be observed in Fig. 11 that different combinations of other participants have different effects on the target participant's emotion prediction. According to Fig. 11 (a), (c) and (d), the accuracy and F1-score have gradually increased to a peak, followed by a decrease to the case "*ALL*" when adding additional other participants to Deep Rolling. A similar tendency has been observed in Fig. 11 (b) that, with the addition of more other participants, the accuracy and F1-score have gradually increased to a peak while a fluctuation has occurred at the top. After that point, the performance of Deep Rolling demonstrated an evident drop.

Furthermore, in terms of the other participant combinations that achieve the highest performance denoted as **bold** font in Table 10, such as {11, 2, 13, 12, 9} on D1, it can be discovered that the most effective other participants with whom to best predict the target participant's emotions include those interact the most often with the target participant. For example, "Jenny Curran (2)" with "Forrest Gump (1)" as lovers on D1, "Andy Dufresne (1)" with "Ellis Boyd Red (2)" as two main characters on D2, "Donna (11)" with "Lieutenant Frank (7)" for famous dancing "Por Una Cabeza" on D3 and "Mayor Ma (22)" with "Li Xue Lian (3)" who has helped Li to resolve her dilemma on D4.

Finally, as shown in Table 10, the time costs of Deep Rolling with different combinations of other participants are close to each other. Such a phenomenon can be attributed to data factorization. When predicting emotion, the emotion vectors of other participants have been organized into a "*cube*". When more and more w-sized pair-based sequences have been extracted, all of the cubes will be stacked together shaped as [batch_size*w, height = n, width = n, uniform_channels], which can be promptly processed by CNN component in Deep Rolling. Consequently, in terms of Deep Rolling, the overall time cost or prediction efficiency will not be directly affected by the addition of more other participants.

In conclusion, Deep Rolling with other participants has outperformed the one without including other participants. In addition, according to Table 10 and Fig. 11, different combinations of other participants can cause a positive or negative stimulation to target participant's emotion prediction. Moreover, for other participants, those who have more interaction with the target participant are more critical to target participant's emotion prediction.

4.5. Deep rolling analysis on different target participants

We finally perform an experiment on Deep Rolling by predicting emotions for different target participants but, similar to the experiment in Table 8, we have input the emotion vectors of all of the other participants at every time step. In this way, we can further compare Deep Rolling with classical and state-of-the-art methods on different target participants. For the above purpose, first, as shown in Table 11, we have selected another four target participants. The first target participant is the one that is used in the experiment illustrated in Table 8. Therefore, the performance of the first target participant in Table 8 has been reused. Second, for the other four target participants, the amount of posted text has gradually decreased. More importantly, we still select the most optimal window size (w) and window stride (s) by grid search, as shown in Fig. 10, for the added target participants. In addition, the experiment has also been conducted by 5-fold validation with multiple times. The specific average accuracy and F1-score, along with the corresponding time cost, have been presented in Tables 12 and 13 respectively. In terms of time cost, in this experiment, the accuracy and F1-score are computed together and the time to assemble the data has not been considered.

In Table 12, we denote the highest performance in **bold** font and the best accuracy or F1-score of the compared methods in **font**. It is clear that, when predicting emotions for different target participants, Deep Rolling can still achieve the best overall accuracy and F1-score compared with adopted classical and state-of-the-art methods. Moreover, in Table 12, we also compute the improvement between Deep Rolling and the best compared method. Compared with the improvement in Table 8, the degree of enhancement in Table 12 can reach a higher level. For example, in terms of accuracy improvement, Deep Rolling has achieved an enhancement of 11.36% on D2, 24.03% and 11.26% on D3, 8.34% and 10.19% on D4. Furthermore, when it comes to F1-score improvement, Deep Rolling has achieved enhancement of 6.01% and 7.45% on D1, 13.40% and 11.98% on D2, 19.57% and 9.00% on D3, as well as 12.42% on D4.

Moreover, the time cost to obtain the above accuracy and F1-score has been presented in Table 13. It is evident that, the time cost of Deep Rolling on different target participants has still been maintained at an acceptable level compared with other adopted methods. More importantly, combined with Table 8, it can be discovered that, the time cost for Deep Rolling on given datasets has retained at a certain level. The reasonable explanation for this phenomenon is that, according

Table 12
Deep Rolling analysis: The prediction accuracy and F1-Score of different target participants on four datasets.

Datasets	Methods	Accuracy collection for target 1–5					F1-score collection for target 1–5				
		Target1	Target2	Target3	Target4	Target5	Target1	Target2	Target3	Target4	Target5
D1	DT_R	0.824107	0.718182	0.554545	0.881818	0.700000	0.818886	0.719971	0.479120	0.875776	0.681896
	LR	0.824107	0.712188	0.554545	0.881818	0.700000	0.819968	0.710449	0.479120	0.875776	0.681896
	SVM_R	0.824194	**0.792857**	0.696396	0.785714	0.500000	0.822127	0.765398	0.675068	0.791429	0.444444
	KNN_R	0.783154	0.782371	0.533333	0.685870	0.570667	0.778958	0.761294	0.501982	0.683570	0.568184
	RF_R	0.830976	0.740000	0.560000	0.880000	0.703333	0.826075	0.732232	0.482640	0.874869	0.682684
	DDAE	0.834483	0.771930	0.765263	0.737500	0.547693	0.829076	**0.770670**	**0.751459**	0.679570	0.500000
	REE	0.793122	0.641611	0.560460	0.696296	0.569565	0.764111	0.602840	0.540000	0.629677	0.530330
	DA_RNN	0.780848	0.640803	**0.780870**	0.647826	0.708696	0.774153	0.612750	0.750324	0.633156	**0.697091**
	R2N2	**0.836690**	0.633333	0.580000	0.613983	0.580000	**0.832319**	0.631111	0.531667	0.613333	0.538239
	CNN_Dilation	0.835897	0.680000	0.647368	**0.887500**	**0.714286**	0.823660	0.661773	0.613627	**0.886310**	0.683746
	Deep Rolling	**0.837778**	**0.822222**	**0.800000**	**0.902734**	**0.755385**	**0.834199**	**0.816951**	**0.784072**	**0.895939**	**0.749033**
	Improvement (%)	+0.13	+3.70	+2.45	+1.72	+5.75	+0.23	+6.01	+4.34	+1.09	+7.45
D2	DT_R	0.819792	0.796000	0.650667	0.762963	0.560000	0.818350	0.772803	0.643610	0.745725	0.585188
	LR	0.810967	**0.821875**	0.668750	0.612500	0.537500	0.805660	**0.822514**	0.669127	0.581715	0.553819
	SVM_R	0.831250	0.810000	0.632000	**0.793333**	0.573333	0.830478	0.808791	0.639236	**0.793843**	0.576077
	KNN_R	0.831351	0.786667	0.656000	0.696000	0.472000	0.816665	0.763519	0.647535	0.678400	0.438016
	RF_R	0.799603	0.818889	0.640000	0.566667	0.580000	0.793002	0.810336	0.640487	0.544124	0.575744
	DDAE	0.751064	0.765306	0.650000	0.676471	**0.702500**	0.676932	0.718992	0.623112	0.613884	**0.660989**
	REE	0.780627	0.671111	0.676190	0.629412	0.625000	0.787936	0.624954	0.648953	0.590727	0.518789
	DA_RNN	0.709684	0.620219	0.687111	0.569444	0.649206	0.700783	0.608183	0.666401	0.562718	0.639304
	R2N2	**0.837596**	0.622424	**0.733333**	0.730000	0.606667	**0.830690**	0.598522	**0.716468**	0.720397	0.608889
	CNN_Dilation	0.762712	0.582353	0.681250	0.712500	0.642857	0.719408	0.514058	0.667339	0.682500	0.585697
	Deep Rolling	**0.837778**	**0.832000**	**0.816667**	**0.828889**	**0.744444**	**0.836197**	**0.825633**	**0.812491**	**0.823297**	**0.740159**
	Improvement (%)	+0.02	+1.23	+11.36	+4.48	+5.97	+0.66	+0.38	+13.40	+3.71	+11.98
D3	DT_R	0.821652	0.637931	0.910000	0.712500	0.733333	0.814601	0.593896	0.879047	0.693143	0.705218
	LR	0.811736	0.637931	0.910000	0.712500	**0.740000**	0.805402	0.593042	0.891241	0.693143	**0.728303**
	SVM_R	0.851034	0.644444	0.885714	0.700000	0.725000	**0.849540**	0.629602	0.837699	0.679707	0.696667
	KNN_R	0.832954	0.600000	0.619327	0.600000	0.585849	0.817952	0.620000	0.592082	0.600000	0.523509
	RF_R	0.855789	0.629630	0.903704	0.603704	0.650000	0.847655	0.596383	0.873947	0.592612	0.643280
	DDAE	0.770115	0.517460	0.430769	**0.813333**	0.681481	0.714166	0.486525	0.406084	**0.801186**	0.657901
	REE	0.835665	0.555686	**0.933333**	0.714286	0.723077	0.811109	0.447521	0.902613	0.639822	0.619631
	DA_RNN	0.685516	0.628482	0.660577	0.743077	0.666667	0.678652	0.613043	0.638349	0.683754	0.602558
	R2N2	0.856667	**0.646667**	0.932381	0.795556	0.713333	0.845873	**0.657778**	**0.910147**	0.780625	0.634386
	CNN_Dilation	**0.858083**	0.564286	0.755556	0.671429	0.675000	0.838417	0.513735	0.736185	0.608740	0.620833
	Deep Rolling	**0.860510**	**0.802063**	**0.948462**	**0.865556**	**0.823333**	**0.850635**	**0.786535**	**0.920591**	**0.851273**	**0.793847**
	Improvement (%)	+0.28	+24.03	+1.62	+6.42	+11.26	+0.13	+19.57	+1.15	+6.25	+9.00
D4	DT_R	0.833824	0.621795	**0.834615**	0.702222	0.606154	0.830318	0.628602	**0.804559**	0.674908	**0.620513**
	LR	0.848148	0.517647	0.741176	0.594118	0.429412	0.845341	0.526579	0.751068	0.574187	0.482350
	SVM_R	0.886531	0.581818	0.776017	0.456992	0.433333	0.886028	0.487395	0.776017	0.456992	0.464646
	KNN_R	**0.891860**	0.608696	0.755385	**0.741026**	0.486957	0.881179	0.605546	0.755385	**0.721041**	0.548196
	RF_R	0.844017	0.611765	0.780392	0.664700	**0.623529**	0.835784	**0.632446**	0.780392	0.666601	0.580381
	DDAE	0.866187	**0.637500**	0.675862	0.518519	0.476190	0.855451	0.605431	0.577791	0.451885	0.401136
	REE	0.809082	0.509402	0.748252	0.542821	0.513187	0.741626	0.411369	0.655745	0.501614	0.490241
	DA_RNN	0.871528	0.568687	0.500000	0.573333	0.547059	0.864735	0.534717	0.485772	0.575306	0.554396
	R2N2	0.887500	0.553333	0.787407	0.609630	0.603846	**0.889545**	0.540305	0.738438	0.593048	0.616805
	CNN_Dilation	0.850000	0.562500	0.700000	0.500000	0.450000	0.888095	0.485165	0.650000	0.466667	0.416667
	Deep Rolling	**0.893088**	**0.690645**	**0.852222**	**0.816522**	**0.673333**	**0.896511**	**0.649870**	**0.823510**	**0.810613**	**0.633052**
	Improvement (%)	+0.14	+8.34	+2.11	+10.19	+7.99	+0.78	+2.76	+2.36	+12.42	+2.02

to Algorithm 2, the computation bottleneck of Deep Rolling is to fuse the stacked "cubes", derived from emotion vectors of other participants, into a 1-D series via the CNN component. And the time spent for LSTM and other merging operations has not occupied a large portion.

It is also important to note that, in Table 12, the prediction precision on Target 2 to Target 5 is overall better than that of Target 1 on D1 to D4. The reasonable explanation for such phenomenon is that, according to Tables 9 and 11, Target 1 has the largest number of lines. At the same time, Target 1 is the main character on D1 to D4. However, Target 2 to Target 5 are supporting roles with fewer lines. Consequently, for Target 1, more instances or (E_{target_t}, $E_{others\_t\_t+1}$) pairs have been extracted from D1 to D4. Furthermore, since Target 1 is the main character, the appearance of its lines has spread all over the corresponding dataset. In other words, the emotion of Target 1 has changed as time passed, stimulated by the emotion of surrounding other participants at different stages. Therefore, in terms of Target 1, Deep Rolling has been assigned with more fine-grained tasks including correctly reflecting the emotion changes of the prediction target at different stages as well as recognizing emotional stimulation caused by other participants in different time periods.

Table 13
Deep Rolling analysis: The time cost for obtaining above accuracy and F1-Score (Accuracy and F1-Score were computed together).

Datasets	Methods	Time cost collection for target 1–5 (ms)				
		Target1	Target2	Target3	Target4	Target5
D1	DT_R	1.009473	0.644303	0.739632	0.631852	0.616357
	LR	0.919124	0.683724	0.775814	0.628564	0.651481
	SVM_R	3.728711	1.111702	0.976009	0.626432	0.626351
	KNN_R	588.375977	1.444157	0.898210	0.938346	0.951872
	RF_R	7.022135	6.681413	6.801107	6.700146	6.660954
	DDAE	0.004772	0.004089	0.004554	0.004934	0.004697
	REE	111.231311	33.425395	29.053132	20.659526	18.858592
	DA_RNN	12915.612790	2466.418505	1557.725286	122.577	123.087764
	R2N2	25.424210	14.620018	14.934158	14.881929	14.646753
	CNN_Dilation	141.690461	167.05606	168.146849	187.528276	157.926607
	Deep Rolling	42.626693	44.17513	41.745736	44.848291	40.423014
D2	DT_R	0.815788	1.465365	0.626383	0.636458	0.670394
	LR	0.843864	1.158415	0.622673	0.623405	0.723877
	SVM_R	1.840088	1.505778	0.703467	0.624674	0.664453
	KNN_R	2.749528	1.672347	1.479980	1.583691	1.527376
	RF_R	8.326790	6.903743	6.565055	6.583854	6.790316
	DDAE	0.006216	0.004576	0.004554	0.004062	0.004246
	REE	39.904229	25.094477	22.875770	17.95586	20.137358
	DA_RNN	1902.978992	1067.855406	442.579794	145.680881	127.745390
	R2N2	23.530976	13.788637	14.454095	15.304359	13.542986
	CNN_Dilation	232.465386	183.532691	190.375042	204.976153	187.437534
	Deep Rolling	40.372754	40.409505	40.854492	43.972249	42.499935
D3	DT_R	1.028092	0.667187	0.643962	0.623356	0.594824
	LR	1.422624	0.697705	0.605208	0.602441	0.600488
	SVM_R	7.776709	0.885417	0.582373	0.625667	0.590234
	KNN_R	555.348600	1.396566	1.110189	0.833691	0.755111
	RF_R	8.348210	6.838818	8.453988	6.757845	6.433854
	DDAE	0.004413	0.004587	0.005459	0.004069	0.004053
	REE	39.666780	26.482821	21.462297	19.461759	17.756208
	DA_RNN	3421.798897	1159.156895	267.493033	170.755315	35.946671
	R2N2	19.577201	17.586772	15.320190	13.699389	14.713589
	CNN_Dilation	129.559072	198.922348	182.728839	216.951323	222.526693
	Deep Rolling	48.475358	41.893783	39.719141	41.203385	40.756706
D4	DT_R	0.859391	0.738737	1.144189	0.690120	0.698910
	LR	1.541341	0.683154	0.702751	0.687158	0.688688
	SVM_R	5.491634	0.692318	0.931055	0.569792	0.598340
	KNN_R	2.634310	1.503760	1.490153	1.491715	1.671354
	RF_R	7.828369	6.833382	6.727637	6.568490	6.674154
	DDAE	0.004211	0.004070	0.004066	0.004350	0.004483
	REE	132.577562	50.820875	48.864301	42.347431	34.499820
	DA_RNN	595.604229	217.431879	214.084435	193.142335	167.466354
	R2N2	13.721418	21.291463	14.929136	29.360310	13.407373
	CNN_Dilation	725.085449	763.240576	867.066097	718.684387	737.897134
	Deep Rolling	42.917757	40.849870	41.972363	40.190658	39.476758

However, when it comes to Target 2 to Target 5, since they are supporting roles, they have a limited number of lines. As a result, fewer instances or (E_{target_t}, $E_{others\_t\_t+1}$) pairs have been extracted. What's more, different from Target 1, the lines of Target 2 to Target 5 have mainly appeared in a period of time instead of the whole movie. In other words, their emotion has not evidently changed and the emotion of other participants is more strongly correlated with the current communication context. Consequently, in terms of Target 2 to Target 5, Deep Rolling can more easily recognize the emotion of fewer instances with the help of emotional stimulation caused by other participants, finally leading to higher accuracy and F1-score as shown in Table 12.

In conclusion, in this paper, we have proposed a novel emotion prediction model, Deep Rolling, for target participant in a multi-participant communication context. According to the above experiments, compared with classical and state-of-the-art methods, our proposed model has achieved the best prediction precision on different target participants. In addition, in terms of other participants, first, a model with other participants has outperformed the one without considering other participants. Second, the emotion of other participants can provide positive or negative stimulation to the target participant's emotion prediction. Finally, for other participants, those who have high interaction with target participant are more critical to the precision of target participant's emotion prediction.

5. Conclusion and discussion

In this paper, we have proposed a novel emotion prediction model, Deep Rolling, to predict the emotions of a target participant in a multi-participant communication context. Basically, the proposed model has processed the text collection as an n-dimension emotion vector based time series. And, in terms of innovation, the proposed model has incorporated the emotion from both the target and all of the other participants at every time step. Finally, the data factorization has also been introduced into the training and prediction phase of Deep Rolling. In conclusion, as shown in various experiments, Deep Rolling can achieve the best prediction precision and acceptable efficiency in comparison with classical and state-of-the-art methods. In addition, a model with other participants can outperform the one without considering other participants.

In terms of future work, on the one hand, since the emotion of other participants can only be considered as a form of stimulation, therefore, in order to predict the target participant's emotion more precisely, we will devote efforts to explore more reasonable ways to merge the emotion of a target and other participants instead of simply utilizing a fully connected layer. On the other hand, in this paper, we only differentiate the emotional stimulation caused by other participants as positive or negative. Consequently, we will also ascertain how the emotion from other participants can affect the emotion of target participant in a quantitative way. Moreover, we will seek an approach with which to embed Deep Rolling into an automatic conversation system to generate more proper responses with predicted emotions based on the current communication context. In this way, human-computer interaction can be enhanced.

Acknowledgments

This work was supported in part by the National Science Foundation of China (no. 61572259, no.U1736105). The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-264. This work was also supported by the Chinese Scholarship Council (CSC, Beijing, China).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.03.023.

References

[1] M. Abadi, Tensorflow: learning functions at scale, ACM Sigplan Notices 51 (9) (2016). 1–1.
[2] C.O. Alm, D. Roth, R. Sproat, Emotions from text: machine learning for text-based emotion prediction, in: Proceedings of the conference on human language technology and empirical methods in natural language processing, Association for Computational Linguistics, 2005, pp. 579–586.
[3] M. Balcilar, K. Thompson, R. Gupta, et al., Testing the asymmetric effects of financial conditions in south africa: a nonlinear vector autoregression approach, J. Int. Financ. Markets Inst. Money 43 (2016) 30–43.
[4] S. Bao, S. Xu, L. Zhang, et al., Mining social emotions from affective text, IEEE Trans. Knowl. Data Eng. 24 (9) (2012) 1658–1670.
[5] r. Biau G, L. Devroye, On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification, J. Multivar. Anal. 101 (10) (2010) 2499–2518.
[6] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. (2003). 993–102. 3(Jan).
[7] A. Borovykh, S. Bohte, C.W. Oosterlee, Conditional time series forecasting with convolutional neural networks, 2017. arXiv:1703.04691 [stat.ML].
[8] B. Chen, X. Liu, H. Zhao, et al., Maximum correntropy Kalman filter , Automatica 76 (2017) 70–77.
[9] C.H. Chen, W.P. Lee, J.Y. Huang, Tracking and recognizing emotions in short text messages from online chatting services, Inf. Process. Manag. 54 (6) (2018) 1325–1344.
[10] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Netw. 17 (1) (2004) 113–126.
[11] Y.M. Chiang, F.J. Chang, Integrating hydrometeorological information for rainfall-runoff modelling by artificial neural networks, Hydrol. Process. 23 (11) (2010) 1650–1659.
[12] K. Cho, B. Van Merrienboer, C. Gulcehre, et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, Comput. Sci. (2014).
[13] N. Colneric, J. Demsar, Emotion recognition on twitter: comparative study and training a unison model, IEEE Trans. Affect Comput. (1949). PP(99):1–1.
[14] G. Dong, Z. Chen, J. Wei, A method for peak power prediction of series-connected lithium-ion battery pack using extended kalman filter, Int. J. Rob. Res. 6 (2) (2017) 134–139.
[15] M.B. Ferraro, R. Coppi, G.G. Rodriguez, et al., A linear regression model for imprecise response, Int. J. Approx. Reason. 51 (7) (2010) 759–770.
[16] H. Gamez-Adorno, J.P. Posadas-Duron, G. Sidorov, et al., Document embeddings learned on various types of n-grams for cross-topic authorship attribution, Computing2018(5) 1–16.
[17] H. Goel, I. Melnyk, A. Banerjee, R2n2: Residual recurrent neural networks for multivariate time series forecasting, 2017. arXiv:1709.03159 [cs.LG]1
[18] V.A. Gromov, E.A. Borisenko, Predictive clustering on non-successive observations for multi-step ahead chaotic time series prediction, Neural Computing & Applications 26 (8) (2015) 1827–1838.
[19] R. He, N. Xiong, L.T. Yang, et al., Using multi-modal semantic association rules to fuse keywords and visual features automatically for web image retrieval , Inf. Fusion 12 (3) (2011) 223–230.
[20] X. Huang, L. Shi, J.A.K. Suykens, Support vector machine classifier with pinball loss, IEEE Trans. Pattern Anal. Mach. Intell. 36 (5) (2014) 984–997.
[21] Y. Huang, Y. Jiang, T. Hasan, et al., A topic biLSTM model for sentiment classification, in: Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence, ACM, 2018, pp. 143–147.
[22] P. Kamble, S. Bhasme, S. Waghale, et al., Emotion determination based on opinion mining, Int. J. Eng. Sci. (2017) 4741.
[23] Z. Kozareva, B. Navarro, et al., UA-ZBSA: a headline emotion classification through web information, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, Association for Computational Linguistics, 2007, pp. 334–337.
[24] F. Krebs, B. Lubascher, T. Moers, et al., Social emotion mining techniques for facebook posts reaction prediction, 2017. arXiv:1712.03249 [cs.AI].
[25] W.J. Lee, J. Na, K. Kim, et al., NARX modeling for real-time optimization of air and gas compression systems in chemical processes, Comput. Chem. Eng. 115 (2018) 262–274.
[26] S. Li, et al., Random KNN classification and regression, Phys. Lett. B 196 (4) (2013) 543–546.
[27] T.N. Lin, C.L. Giles, B.G. Horne, et al., A delay damage model selection algorithm for NARX neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2719–2730.

[28] B. Liu, Sentiment analysis: mining opinions, sentiments, and emotions, Comput. Linguist. 42 (3) (2016) 1–4.
[29] Y. Lv, T. Ma, M. Tang, et al., An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (2016) 9–22.
[30] T. Ma, H. Rong, H. Chang, Y. Tian, Abdullah Al-Dhelaan, Mznah Al-Rodhaan. detect structural-connected communities based on BSCHEF in c-DBLP, Concurr. Computat. 28 (2) (2016) 311–330.
[31] T. Ma, Y. Wang, M. Tang, J. Cao, Y. Tian, Abdullah al-dhelaan, mznah al-rodhaan, LED: a fast overlapping communities detection algorithm based on structural clustering, Neurocomputing 207 (2016) 488–500.
[32] S. Madisetty, M.S. Desarkar, An ensemble based method for predicting emotion intensity of tweets, Min. Intell. Knowl. Explor. (2017) 359–370.
[33] B. Moghimi, A. Safikhani, C. Kamga, et al., Cycle-length prediction in actuated traffic-signal control using ARIMA model, J. Comput. Civil Eng. 32 (2) (2018).
[34] S. Mundra, A. Sen, M. Sinha, et al., Fine-grained emotion detection in contact center chat utterances, Adv. Knowl. Discov. Data Min. (2017) 337–349.
[35] M.T. Nguyen, D.V. Tran, L.M. Nguyen, Social context summarization using user-generated content and third-party sources, Knowl. Based Syst. 144 (2018) 51–64.
[36] S. Pal, S. Ghosh, A. Nag, Sentiment analysis in the light of LSTM recurrent neural networks, Int. J. Synth. Emot. (IJSE) 9 (1) (2018) 33–39.
[37] A. Poon, The transmission mechanism of malaysian monetary policy: a time-varying vector autoregression approach, Empir. Econ. (29) (2017) 1–28.
[38] M. Qin, Z. Li, Z. Du, Red tide time series forecasting by combining ARIMA and deep belief network, Knowl. Based Syst. (125) (2017) 39–52.
[39] Y. Qin, D. Song, H. Chen, et al., A dual-stage attention-based recurrent neural network for time series prediction, 2017, 2627–2633. arXiv:1704.02971 [cs.LG].
[40] A. Rahman, V. Srikumar, A.D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, Appl. Energy 212 (2018) 372–385.
[41] S. Ren, K. He, R. Girshick, et al., Object detection networks on convolutional feature maps, IEEE Trans. Pattern Anal. Mach. Intell. 39 (7) (2017) 1476–1481.
[42] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors., Nature 323 (6088) (1986) 399–421.
[43] Q. Song, Y.J. Zheng, Y. Xue, et al., An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination , Neurocomputing 226 (C) (2017) 16–22.
[44] L. Wang, S. Guo, W. Huang, et al., Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs, IEEE Trans. Image Process. (2017). PP(99):1–1.
[45] Y. Wang, K. Lin, Y. Qi, et al., Estimating brain connectivity with varying length time lags using recurrent neural network, IEEE Trans. Biomed. Eng. (2018). PP(99):1–1.
[46] M. Xu, P. Watanachaturaporn, P.K. Varshney, et al., Decision tree regression for soft classification of remote sensing data, Remote Sens. Environ. 97 (3) (2005) 322–336.
[47] L. Yan, A. Elgamal, G.W. Cottrell, Substructure vibration NARX neural network approach for statistical damage inference, J. Eng. Mech. 139 (6) (2013) 737–747.
[48] T. Zhang, W. Zheng, Z. Cui, et al., Spatial-temporal recurrent neural network for emotion recognition, IEEE Trans. Cybern. (2017) 1–9. PP(99).
[49] X. Zhao, C. Wang, Z. Yang, et al., Online news emotion prediction with bidirectional LSTM, in: International Conference on Web-Age Information Management, Springer, Cham, 2016, pp. 238–250.
[50] H. Zhou, M. Huang, T. Zhang, et al., Emotional chatting machine: emotional conversation generation with internal and external memory, 2017. arXiv:1704.01074 [cs.CL].