Mathematical Biosciences and Engineering

MBE, 19(3): 2671–2699.
DOI: 10.3934/mbe.2022122
Received: 19 October 2021
Revised: 01 January 2021
Accepted: 04 January 2021
Published: 10 January 2022

http://www.aimspress.com/journal/MBE

Research article

TEP2MP: A text-emotion prediction model oriented to multi-participant text-conversation scenario with hybrid attention enhancement

Huan Rong[1], Tinghuai Ma[2,*], Xinyu Cao[2,*], Xin Yu[2] and Gongchi Chen[3]

[1] School of Artificial Intelligence (School of Future Technology), Nanjing University of Information Science and Technology, Nanjing 210044, China
[2] School of Computer & Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China

* **Correspondence:** Email: thma@nuist.edu.cn, caoxinyu0033@163.com; Tel: +8613584061562.

Abstract: With the rapid development of online social networks, text-communication has become an indispensable part of daily life. Mining the emotion hidden behind the conversation-text is of prime significance and application value when it comes to the government public-opinion supervision, enterprise decision-making, etc. Therefore, in this paper, we propose a text emotion prediction model in a multi-participant text-conversation scenario, which aims to effectively predict the emotion of the text to be posted by target speaker in the future. Specifically, first, an *affective space mapping* is constructed, which represents the original conversation-text as an *n*-dimensional *affective vector* so as to obtain the text representation on different emotion categories. Second, a similar scene search mechanism is adopted to seek several sub-sequences which contain similar tendency on emotion shift to that of the current conversation scene. Finally, the text emotion prediction model is constructed in a two-layer encoder-decoder structure with the emotion fusion and hybrid attention mechanism introduced at the encoder and decoder side respectively. According to the experimental results, our proposed model can achieve an overall best performance on emotion prediction due to the auxiliary features extracted from similar scenes and the adoption of emotion fusion as well as the hybrid attention mechanism. At the same time, the prediction efficiency can still be controlled at an acceptable level.

Keywords: text sentiment analysis; time series prediction; deep learning

1. Introduction

With the rapid development of Internet, communicating with others on the social network or making comments on the given object by text has already become an indispensable part of our daily life. As shown in Figure 1, multiple-participants can talk with each other by text. In such the scenario, conversation text can implicitly reflect the subjective sentiment or emotion of the publisher. Therefore, if the sentiment or emotion hidden behind the conversation text published by the target speaker could be predicted in advance, it would bring about more benefit to the supervision on public opinion or the adjustment on marketing strategy.

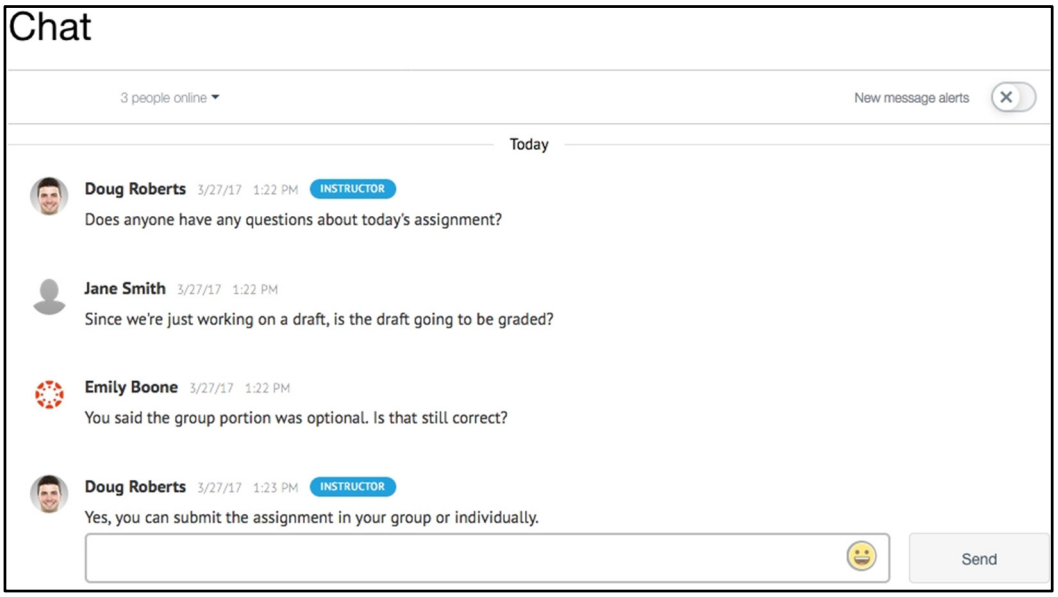


Figure 1. The demonstration of text-conversation in a multi-participant scenario (Note: Doug Roberts is the target speaker).

However, in order to effectively realize the text emotion prediction in a multi-participant conversation scenario as shown in Figure 1, several challenges should be overcome. First, different from common numeric features, the emotion of the given conversation text is hard to be extracted directly. Therefore, an *affective space* should be established to particularly represent the emotion on the given conversation text. Second, as shown in Figure 1, the available conversation text is relatively short, which means a single piece of text could only reflect the limited portion of the conversation context, let alone the tendency on emotion shift within a given period of time. In other words, it is difficult to predict text emotion with high accuracy, only depending on a single piece of conversation text and more auxiliary features should be introduced. Third, in a multi-participant conversation scenario, there exists interaction on emotions among different participants. Therefore, in order to improve the prediction accuracy, it is necessary to introduce additional mechanism to process the emotion interaction among participants to provide more critical features to enhance the text emotion prediction performance in terms of the target speaker.

In order to overcome above challenges, in this paper, we propose a text emotion prediction model oriented to the multi-participant text conversation scenario, denoted as TEP2MP, with the support of emotion fusion and hybrid attention mechanism. Specifically, the proposed model TEP2MP first

conducts an "*Affective Space Mapping*", which particularly extracts emotion features from the given conversation text collection. In this way, all the conversation texts can be converted into *n*-dimensional *affective vectors*, where each dimension of the *affective vector* represents the preference to one emotion category. Based on the affective vector output by the "*Affective Space Mapping*", the proposed model TEP2MP assembles all the *affective vectors* into multiple data-units, according to the order in which the conversation text has been published. Here, each data-unit contains the affective vectors of the corresponding conversation texts, in the form of <1 *Target Speaker, m Other Participants*>. In other words, one data-unit is a combination of ***one*** *n*-dimensional *affective* vector corresponding to the text published by the target speaker and ***m*** *n*-dimensional *affective* vectors of the texts published by all the other participants. In this way, as shown in Figure 1, the original conversation text collection can be converted into a time-series particularly on affective vectors and each element in such the time-series is a basic data-unit formed as <1 *Target Speaker, m Other Participants*>.

Moreover, due to the fact that in the given time-series, there may exist multiple sub-sequences which present the similar tendency on the shift of emotion category occurring at different intervals [1], therefore the proposed model TEP2MP takes two adjacent data-units in above time-series as the current scene, in the form of [<1 *Target Speaker, m Other Participants*>$_t$; <1 *Target Speaker, m Other Participants*>$_{t+1}$]. Obviously, the current scene is a sub-sequence consisting of several *n*-dimensional *affective vectors*. Then, according to the current scene, multiple similar scenes will be sought across the entire conversation history, or the whole time-series, by different time-spans such as day, week and month. Here, a similar scene is the sub-sequence having similar change on emotion category compared with that of the current scene. In addition, the similar scenes will be considered as auxiliary features used to support following text-emotion prediction for target speaker. Finally, the input conversation text collection, or the time-series consisting of *n*-dimensional *affective vectors*, will be aligned up to the similar scenes, both of which will be fed into a two-layer encoder-decoder prediction model constructed by Long Short Term Memory (LSTM) [2]. As a result, the emotion of the conversation text to be published by target speaker in the future can be predicted in a window-rolling manner, even if the specific content of such the conversation text is still unknown.

More importantly, in terms of the prediction model constructed as the two-layer encoder-decoder, emotion fusion mechanism is introduced at the encoder side to merge the *n*-dimensional *affective vector* of the conversation text published by target speaker and those belonging to other participants, both of which are stored in the same data-unit (i.e., <1 *Target Speaker, m Other Participants*>$_t$). The result of emotion fusion will be forced to be "close" to the *n*-dimensional *affective vector* of the conversation text published by the target speaker in next data-unit. In this way, our proposed prediction model can capture the emotion interaction among participants at the encoder side as early as possible. Moreover, at the decoder side, a hybrid attention mechanism is adopted to compute two context vectors derived from the input conversation text collection and the similar scenes. Such two context vectors will be merged by a "gate switcher" before fed into decoder, aiming at obtaining critical features on both the current scene (i.e., derived from the input conversation text) and similar scene to improve the final performance on text emotion prediction with regard to the target speaker.

In conclusion, the contribution of this paper is that a text emotion prediction model oriented to multi-participant text-conversation scenario has been proposed with the support of emotion fusion and hybrid attention mechanism. The proposed model aims at effectively predicting the emotion of the conversation text to be published by the target speaker in the future, even if the specific content of the corresponding conversation text is still unknown. To achieve above goal, first, "*Affective Space Mapping*" has been constructed to represent each conversation text as an *n*-dimensional *affective vector*

on particular emotion categories. Second, scenes with similar tendency on emotion category shift have been sought across the entire conversation history to provide auxiliary features for the following prediction on text emotion. Third, in terms of the emotion interaction among different participants, an emotion fusion mechanism is adopted at the encoder side to merge the *affective vectors* corresponding to the conversation texts posted by different participants, and a hybrid attention mechanism is adopted at the decoder side to obtain global observation on the current scene and similar scene across the entire conversation history.

2. Related works

In recent years, text sentiment (emotion) analysis has attracted attention from more and more researchers [3]. Particularly, the general process of text sentiment (emotion) analysis can be described as: given a collection of texts, the sentiment or emotion category of each text should be recognized correctly by a classifier, based on the proper text representation. Such the process can also be considered as text sentiment (or emotion) prediction [4]. In addition, the text representation means the feature vector of corresponding text which can be obtained by word embedding algorithms (i.e., Skip-Gram [5]) or other neural components like Auto-Encoder [6]. Moreover, the semantics or syntactics of given text should also be incorporated into the corresponding representation.

More importantly, the term "emotion" refers to the intensive and instant feelings such as happiness, anger, sadness, fear and surprise, etc. [7]. However, the term "sentiment" often means a feeling or an opinion, especially based on emotions. Typically, in terms of the text sentiment analysis, the existing models or algorithms often classify the sentiment of corresponding texts as positive, neutral or negative [8]. Cambria E et al. [9] have constructed a commonsense knowledge base "SenticNet 6" for sentiment analysis, in which each term has been assigned with a polarity value in the range of [-1, 1] with the help of logical reasoning by deep learning architectures. It is worthwhile to be mentioned that, since the sentiment is derived from emotion and it is also feasible to categorize the specific emotion as "positive" or "negative", therefore in this paper, we consider the term "sentiment" and "emotion" as mutually equivalent.

Specifically, Basiri M E et al. [10] have extracted features from text segments and constructed a deep bi-directional network with attention mechanism to conduct sentiment prediction. Gong C et al. [11] have adopted BERT, a pre-trained language model to embed words into vectors, based on which the semantical dependency between terms can be analyzed. Similarly, Peng H et al. [12] have projected word representation into a sentiment category space and a mapping function like "word representation→sentiment category" has been learned. In addition, in order to improve the effectiveness to capture critical features from the given text, Yang C et al. [13] and Cai H et al. [14] have adopted Co-Attention Network and Graph Convolutional Network to build the projection from word embedding to text sentiment category. In terms of the text context, Phan M H et al. [15] have considered the document being processed as the local context and the given corpora as the global context. Then, a pre-trained BERT language model is adopted to encode the local and the global context into vectors, followed by the fusion of above two context vectors via self-attention mechanism, based on which the sentiment of the given text will be determined.

Recently, sentiment or emotion analysis in dialog systems has attracted more and more attention. Typically, Ma et al. [16] have concentrated on the literature of empathetic dialogue system whose goal is to generate response in a text-conversation scenario with proper sentiment, more adapted to the

interaction during conversation. In such the survey, emotion-awareness, personality-awareness and knowledge-accessibility have been considered as three critical factors to enhance the perception and expression of emotional states. Moreover, Poria S. et al. [17] have also pointed out that the sentiment reasoning aiming at digging into the detail that causes sentiment is of prime importance with regard to sentiment perception in dialog systems, whose main purpose is to analyze the motivation behind sentiment shift via conversation. In order to improve the performance on sentiment analysis in dialog systems, Li W et al. [18] have proposed a bidirectional emotional recurrent unit (BiERU) whose main innovation is the generalized recurrent tensor block followed by two-channel classifier designed to perform context compositionality and sentiment classification simultaneously. In this way, the bi-directional emotional recurrent unit presents to be fast, compact and parameter-efficient in terms of conversational sentiment analysis. Similarly, Lian Z et al. [19] have improved the transformer network, making it more adapted to the conversational emotion recognition, where the conversation texts are embedded into vectors and the bi-directional recurrent units are utilized to revise above representations with the help of multi-head attention mechanism. In this way, an emotion classification model on conversation text can be constructed to determine the specific emotion of each conversation text. In addition, Zhang Y et al. [20] have utilized the LSTM network, a variant of recurrent-unit-based model, to capture the interaction during communication so as to realize more precisely conversational sentiment analysis. Finally, Wang J. [21] have introduced the topic-aware mechanism into the conversational sentiment analysis, where the sentiment or emotion of each conversation text is determined based on the text representation, meanwhile the topic of current conversation context should also be recognized correctly.

Moreover, in a daily communication scenario, texts are published in sequence. Therefore, the principle of time-series prediction has offered an inspiration to the task of emotion-prediction in a multi-participant text-communication scenario. Typically, representative time-series prediction models are listed as follows. First, a time-series prediction model Attn-CNN-LSTM [22] based on hybrid neural networks and attention mechanism has been proposed, which conducts "phase reconstruction" on given time-series. In terms of the prediction, spatial features are extracted first, followed by the extraction of temporal features via LSTM network. Then, the periodic tendency on sub-sequence is also mined out, working together with above temporal-spatial features extracted from the entire time-series so as to enhance the prediction performance. Similarly, an attention-mechanism based two-phase bi-directional neural network DSTP-RNN [23] conducts time-series prediction by temporal-spatial features as well. Such the temporal-spatial features are extracted from the target sequence and the corresponding exogenous sequence across different periods. Besides, the time-series prediction model DeepAR [24] constructed by auto-regressive recurrent neural network splits the input sequence into main sequence Z, which contains unknown data point to be predicted. And, the remaining co-variate sequence X without any unknown data point is taken to assist the prediction of Z. Such two sequences are aligned up by time step, formed as (z, x), and merged by Recurrent Neural Network to fit a proper distribution, based on which the unknown data point in main sequence Z will be predicted. In addition, the Clockwork-RNN network [25] has also been adopted for time-series prediction [26]. Specifically, temporal features of given time-series are extracted by Clockwork-RNN. Then, dependencies among data points are learned by traditional feed-forward neural network with proper distribution fit by Vector Autoregression, based on which a time-series prediction model has been constructed according to the principle of Stacking ensemble learning [27]. Such the prediction model is required to be fine-tuned by a large amount of high quality data points so

as to obtain stable prediction performance. Finally, a time-series prediction method based on fuzzy cognitive map, denoted as EMD-HFCM [28] has been proposed which extracts features from the input sequence by empirical mode decomposition. The extracted features are used to construct high-order fuzzy cognitive map iteratively used for prediction. Similarly, the prediction model CNN-FCM [29] adopts residual block to extract features from time-series, and the prediction process is completed by Fully Connected Neural Network, as a substitute of the fuzzy cognitive map.

In conclusion, it can be found that existing methods still have two disadvantages with regard to the text emotion prediction problem in a multi-participant conversation scenario. First, most text sentiment (emotion) prediction models need to ascertain the text content in advance, then the category on text sentiment or emotion can be determined by classification. Such requirement has limitation on real-world application, where the content of the text to be published in the future often presents to be unknown. Second, the existing time-series prediction model has insufficient consideration on the similar sub-sequence of the given time-series, resulting in the requirement on a large amount of training instances to provide more critical features [24]. In other words, sub-sequence with similar tendency in the given time-series should be analyzed further to provide more auxiliary features available for final prediction.

Consequently, in this paper, we convert the text emotion prediction in a multi-participant conversation scenario into the task of time-series prediction. However, current time-series model conducts prediction mainly by analyzing the tendency on target variable, not involved with the processing on sentiment or emotion interaction among different participants. In other words, existing time-series prediction models can not be applied directly. Therefore, effort should be devoted to construct more advanced time-series-based text emotion prediction model, making it more adapted to the multi-participant text conversation scenario.

3. The proposed method

3.1. Text affective space mapping

In this paper, we focus on the text-emotion prediction task in a multi-participant conversation scenario, whose ultimate goal is to correctly categorize the emotion or sentiment of the conversation-text to be posted in the future. Consequently, the conversation text should be first represented or embedded into vectors so as to be analyzed further [3,4]. However, different from the existing embedding algorithms [5,6], whose principle is to "shorten" the vector distance for terms semantically related or similar, features on emotion or sentiment should also be incorporated into text representation in order to enhance the emotion prediction performance. Therefore, we first turn to "*Affective Space Mapping*", aiming at obtaining text representation containing details on corresponding emotion or sentiment categories.

It is obvious that, by the time at which the text has been published, the original conversation text collection can be converted into a sequence, like $D = \{d_1, d_2, ..., d_i | 0 < i + 1 < T\}$. Here d_i represents a single text published by one participant in the conversation. And, the process of *Affective Space Mapping* on text has been illustrated in Figure 2.

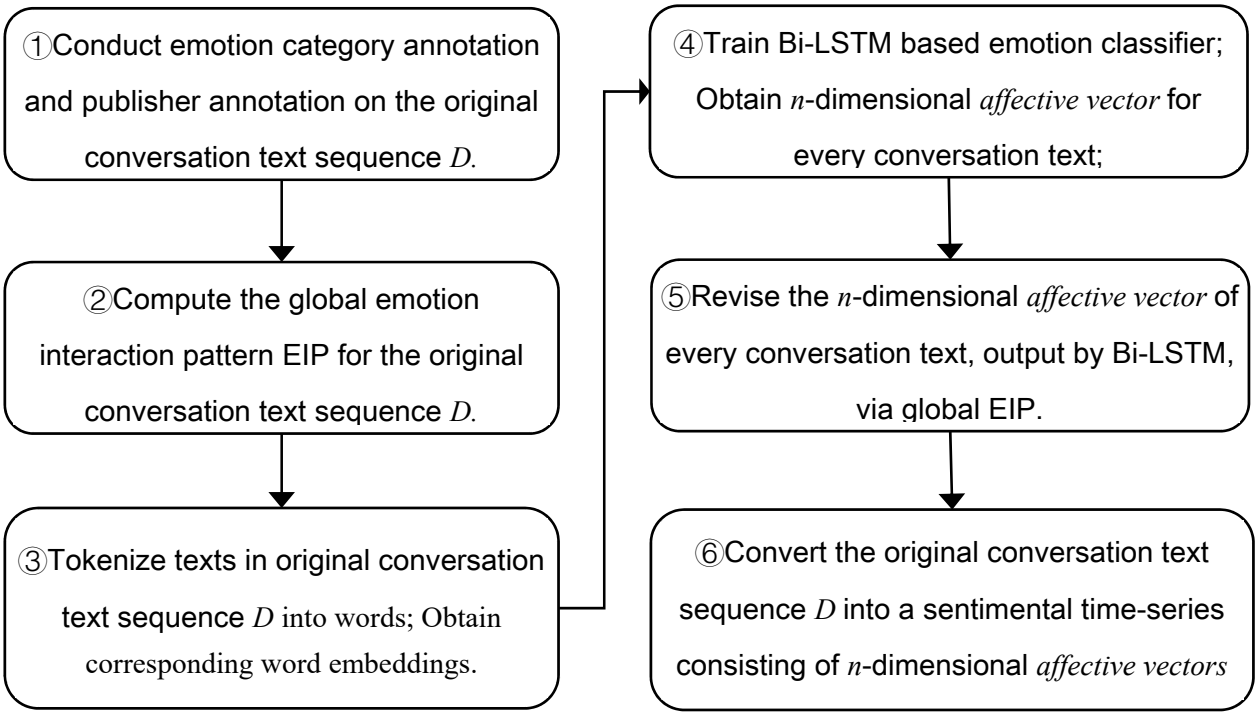


Figure 2. The process of *Affective Space Mapping*.

First, the sentiment or emotion of the original conversation text sequence $D = \{d_1, d_2, ..., d_i | 0 < i + 1 < T\}$ will be annotated like [*Target_Flag*, *Emotion_Index*]. Here, *Target_Flag* (0/1) means if the current text is published by the target speaker (i.e., 1 for target speaker) and the *Emotion_Index* means the specific sentiment or emotion category annotated for each conversation text. Based on the text sentiment or emotion annotation, the original conversation text sequence $D = \{d_1, d_2, ..., d_i | 0 < i + 1 < T\}$ can be converted into an *Emotion Category Sequence*, in which each element is the *Emotion_Index* or the specific emotion category of the corresponding conversation text.

Second, as shown in Algorithm 1, the global Emotion Interaction Pattern (EIP) [30] of the original conversation text sequence D will be computed based on above *Emotion Category Sequence*. Here, the global EIP is used to depict the distribution on n emotion categories in terms of the entire original conversation text sequence $D = \{d_1, d_2, ..., d_i | 0 < i + 1 < T\}$.

In Algorithm 1, at step 1, a global emotion interaction dictionary *dict* is initialized by n emotion categories ($e_1, e_2, ..., e_n$). In terms of such the *dict*, each emotion category e has its own n-dimensional list, initialized as $[0, 0, ...0] \in R^n$, which represents the co-occurrence frequency with other emotion categories when encountering an emotion category e in the original conversation text sequence D. In this way, the interaction among different emotion categories can be incorporated in one EIP dictionary. At Step 2, a time-window with size 2, stride 1 has been created. Such the time-window is used to scan the whole *Emotion Category Sequence* like (Emotion_Index$_i$, Emotion_Index$_j$), based on which the ith and jth dimension of the n-dimensional list of emotion category e_i is updated (i.e., add 1 frequency count) to revise the global EIP dictionary *dict*. Finally, at Step 3, when the above time-window has moved to end of *Emotion Category Sequence*, "*softmax*" operation is conducted towards all the n-dimension list in the global EIP dictionary to obtain the final normalized EIP matrix, which can be considered as the emotion distribution on n emotion categories in terms of the given conversation text sequence D. Specifically, when selected four emotions such as "happy", "sad", "angry" and "other",

the global EIP output by Algorithm 1 is shown in Figure 3 and the index of the maximum value in each line of EIP represents the corresponding emotion category.

Algorithm 1. The process to compute the global Emotion Interaction Pattern.

Input: The *Emotion Category Sequence* corresponding to the original conversation text sequence D

Output: The global EIP of the original conversation text sequence D

1. Select n emotion categories (e_1, e_2, ..., e_n) and initialize the global EIP dictionary as follows:

   $dict$ = {e_1:[0, 0, ..., 0], e_2:[0, 0, ..., 0], ..., e_n:[0, 0, ..., 0]};

2. Create a time-window with size=2, stride=1 to scan the entire *Emotion Category Sequence*. Revise the global EIP dictionary according to the pair of emotion category (i.e., e_1, e_2) in the current time-window, as:

   ① (Emotion_Index$_1$, Emotion_Index$_2$) → $dict$ = {e_1: [1, 1, ..., 0], e_2: [0, 0, ..., 0], ..., e_n: [0, 0, ..., 0]};

   ② (Emotion_Index$_2$, Emotion_Index$_n$) → $dict$ = {e_1: [1, 1, ..., 0], e_2: [0, 1, ..., 1], ..., e_n: [0, 0, ..., 0]};

   ③ (Emotion_Index$_n$, Emotion_Index$_2$) → $dict$ = {e_1: [1, 1, ..., 0], e_2: [0, 1, ..., 1], ..., e_n: [0, 1, ..., 1]};

   ④ (Emotion_Index$_2$, Emotion_Index$_1$) → $dict$ = {e_1: [1, 1, ..., 0], e_2: [1, 2, ..., 1], ..., e_n: [0, 1, ..., 1]};

   .....

3. Repeat Step 2, scanning to the end of *Emotion Category Sequence* by moving above time window with stride = 1, revising the global EIP dictionary according to the pair of emotion category (i.e., e_1, e_2) in the current time-window iteratively;

4. Normalize the global EIP dictionary by "*softmax*", which is output as the final global EIP of the original conversation text sequence D;

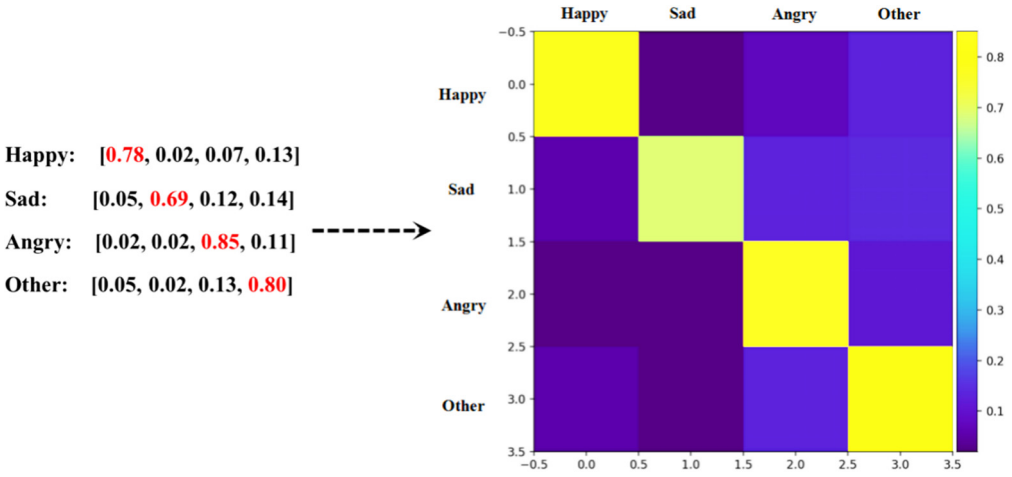


Figure 3. The example of global EIP.

Third, in terms of the *Affective Space Mapping* as shown in Figure 2, the original conversation text in sequence D will be tokenized into words, which are then embedded into vectors by Skip-gram [5] iteratively. Fourth, based on above word vectors, the original conversation text sequence D will be fed into Bi-LSTM [18] to be represented as a sequence of word vectors. Here, the adopted Bi-LSTM works as an n emotion category classifier. Then, the trained Bi-LSTM will be reused to compute the emotion distribution on n emotion categories (with *softmax* normalization) for each conversation text in D. In

this way, the normalized n-dimension emotion distribution, or *affective vector*, for each conversation text can be obtained. Fifth, for every conversation text in sequence D, the index of the maximum value of the n-dimensional *affective vector* will be checked, ensuring the consistency to the Emotion_Index$_i$ annotated at the beginning. If a conflict occurred, then according to the Emotion_Index$_i$ annotated before, the corresponding n-dimensional list in EIP as shown in Figure 3 will be extracted, working as a substitution of the n-dimensional *affective vector* of the current conversation text. Finally, the original text conversation sequence D can be represented as a sequence of n-dimensional *affective vectors*, denoted as $D' = \{E_1, E_2, ..., E_i, E_i = (e_1, e_2, ..., e_n) \in R^n | 0 < i + 1 < T, e_n \in R\}$, where each $E_i = (e_1, e_2, ..., e_n) \in \mathrm{R}^n$ is an n-dimensional *affective vector*.

More importantly, at the final step of *Affective Space Mapping* as shown in Figure 2, according to the *Target_Flag* (0/1) annotated at the beginning, all the n-dimensional *affective vectors* corresponding to the conversation texts published by the target speaker (*Target_Flag* = 1) will be extracted from $D' = \{E_1, E_2, ..., E_i, E_i = (e_1, e_2, ..., e_n) \in R^n | 0 < i + 1 < T, e_n \in R\}$, denoted as $D'_{target} = \{E_{target\_1}, E_{target\_2}, ..., E_{target\_i} | 0 < i + 1 < T, E_{target\_i} \in R^n\}$. Here, E_{target_i} refers to an n-dimensional *affective vector* of the conversation text published by the target speaker at time step i. Moreover, in terms of the adjacent time step i and $i + 1$, *all* the texts published by *other participants* between time step i and $i + 1$ will be collected and the corresponding n-dimensional *affective vector collection* between time step i and $i + 1$ is denoted as $E_{others\_i\_i+1}$. As a result, the n-dimensional *affective vector* corresponding to the conversation text published by target speaker at time step i (i.e., E_{target_i}), along with a set of n-dimensional *affective vectors* derived from the conversation texts published by all the other participants between time step i and $i + 1$ will be assembled into a basic data-unit altogether, denoted as $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$.

After *Affective Space Mapping* illustrated in Figure 2, the original conversation text sequence D will be converted into a time-series $X = \{x_1, x_2, ..., x_i | 0 < \mathrm{i} + 1 < T\}$, consisting of multiple n-dimensional *affective vectors*. Each element in time-series X is $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$, which is a basic data-unit containing n-dimensional *affective vectors* corresponding to the target and other participants between the adjacent time step i and $i + 1$.

3.2. *Similar scene search and attention sequence extraction*

As mentioned above, after *Affective Space Mapping*, the original conversation text sequence D can be represented as a time-series $X = \{x_1, x_2, ..., x_i | 0 < i + 1 < T\}$, consisting of multiple n-dimensional *affective vectors*. Particularly, every element in the input series X is a basic data-unit, or a sub-sequence, like $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$, which contains one n-dimensional *affective vector* of the conversation text published by the target speaker at time step i and a set of n-dimensional *affective vectors* of texts published by all the other participants between time step i and $i + 1$.

Generally, most time series may have pseudo-periodicity, which means several sub-sequences may present similar tendency on the change of values, occurring at regular or irregular intervals [31]. Such the similar sub-sequence brought about by the pseudo-periodicity can provide extra features for the prediction of the given time-series [32]. Consequently, inspired by such principle, when it comes to the time-series $X = \{x_1, x_2, ..., x_i | 0 < i + 1 < T\}$ derived from the multi-participant text-conversation scenario, although the shift among different sentiment or emotion categories may not present to be periodical, yet there may exist several sub-sequences containing similar tendency on the emotion category shift to that of the given sub-sequence $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$. For instance, the emotion of the target speaker could shift from "sad" to "angry" at different time steps across the entire

conversation history. Therefore, sub-sequences with similar tendency on emotion category shift should be collected for the given sub-sequence $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$ for further processing, so that more auxiliary features can be collected for analyzing the sub-sequence $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$.

Based on above inspiration, as shown in Figure 4, when given the input time-series $X = \{x_1, x_2, ..., x_i | 0 < i + 1 < T\}$, where $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$, an input sequence $X_P = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ spanning p_t time steps will be selected from X. Here, each element in X_p can be represented as $x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}$. Then, a target sequence $Y_P = \{y_{p_1}, y_{p_2}, ..., y_{p_t}, p_t \in T\}$, or the ground truth to be predicted later, will also be selected from X. It should be noted that the start point of the target sequence Y_p is the time step of x_{p_2} in X_p and the target sequence Y_p spans p_t time steps as well. In other words, for $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$, $y_{p_1} = x_{p_2}$, $y_{p_2} = x_{p_3}$, ..., $y_{p_t} = x_{p_{t+1}}$, the target sequence Y_p is actually obtained by shifting the input sequence X_p one time step forward, in the direction to t+1. Finally, as shown in Figure 4, the current scene (or a sub-sequence) can be constructed as the combination of one element in X_p and another element in Y_p, denoted as $s_{p_t} = \{x_{p_t}, y_{p_t}\}$.

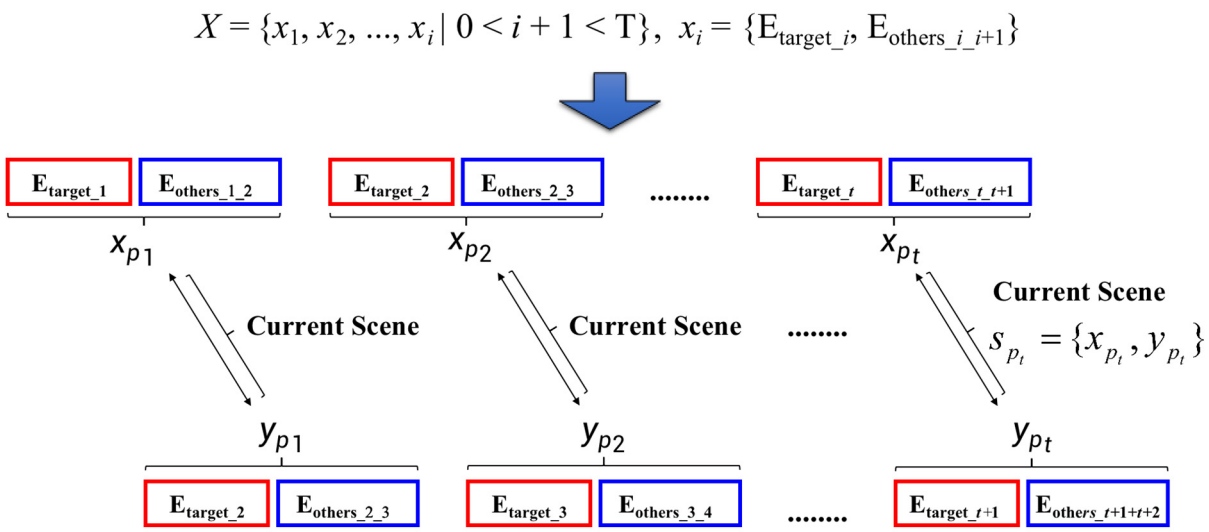


Figure 4. The construction of current scene from the given sentimental time-series X.

In this way, when given the input sequence $X_P = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ and a target sequence $Y_P = \{y_{p_1}, y_{p_2}, ..., y_{p_t}, p_t \in T\}$, $\boldsymbol{t}$ current scenes (or sub-sequences) in all will be extracted from the entire time-series $X = \{x_1, x_2, ..., x_i | 0 < i + 1 < T\}$, where $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$. Here, $x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}$, $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$.

Moreover, in terms of the input time-series $X = \{x_1, x_2, ..., x_i | 0 < i + 1 < T\}$ where $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$, as shown in Figure 5, a period-offset searching method is adopted to extract multiple similar scenes (or several sub-sequences) from the input time-series X by different periods (i.e., day, week or month), according to the given current scene (sub-sequence) $s_{p_t} = \{x_{p_t}, y_{p_t}\}$. Here, the extracted similar scenes must have the similar change on emotion categories compared to that of the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$. More specifically, when given a current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$, $x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}$ and $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$, $\boldsymbol{K}$ similar scenes (sub-sequences), denoted as $s_m = \{x'_{m_j}, y'_{m_j}\}$, $j \in [1, K]$, will be extracted from the input time-series $X = \{x_1, x_2, ..., x_i$

$|0 < i + 1 < T\}$, $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$, satisfying the requirement that $x'_{m_j} = \{E_{target\_j}, E_{others\_j\_j+1}\}$, $y'_{m_j} = \{E_{target\_j+1}, E_{others\_j+1\_j+2}\}$, and $x'_{m_j} \approx x_{p_t}$. .

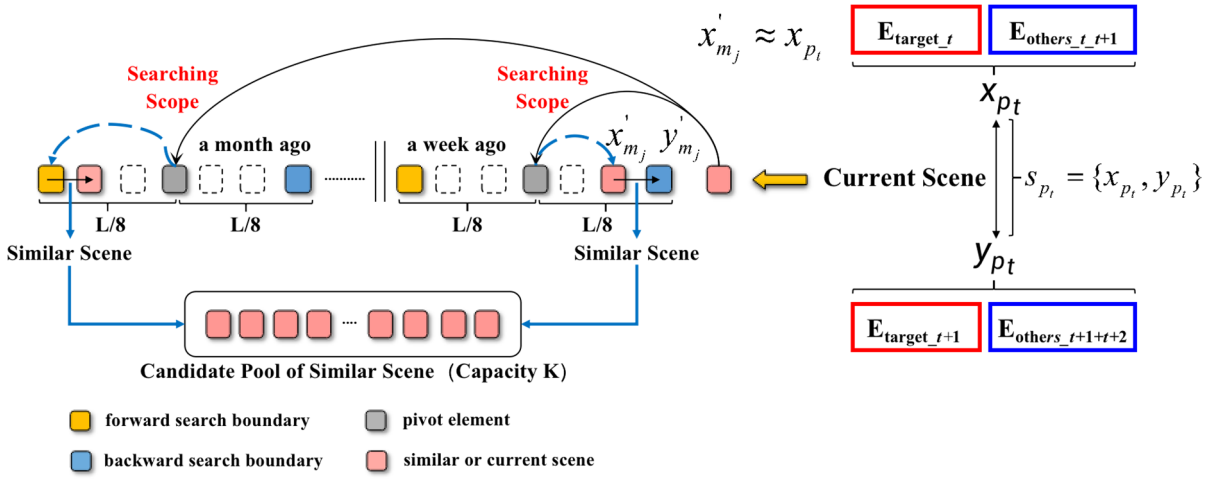


Figure 5. The searching of similar scenes according to the given current scene.

More explicitly, as illustrated in Figure 5, when given a current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$, the adopted period-offset searching method will find a pivot element at first, which is a similar sub-sequence nearest to the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ across the input time-series *X*. For instance, in Figure 5, a pivot element is first found by week. Then, starting from the pivot element, a searching scope with size *L*/8 will be expanded from left and right respectively, in which an element x'_{m_j} most similar to x_{p_t} in $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ will be determined and the next element y'_{m_j} is taken to construct a similar scene (sub-sequence) like $s_m = \{x'_{m_j}, y'_{m_j}\}$. Such the similar scene will be stored in a pool of candidate similar scene with capacity *K*. Repeating above searching process, traversing the entire time-series *X* by different time periods (i.e., day, week and month) until the amount of the candidate similar scenes has achieved the threshold ***K***, in this way, ***K*** similar scenes (sub-sequences) corresponding to the current scene (sub-sequence) $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ will be obtained by different periods, denoted as $s_m = \{x'_{m_j}, y'_{m_j}\}, j \in [1, K]$. In conclusion, for each current scene (sub-sequence) $s_{p_t} = \{x_{p_t}, y_{p_t}\}$, where $x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}$ and $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$, a set of ***K*** similar scenes (sub-sequences) $s_m = \{x'_{m_j}, y'_{m_j}\}, j \in [1, K]$ will be extracted from the input time-series X by the period-offset searching method as shown in Figure 5. Here, $x'_{m_j} = \{E_{target\_j}, E_{others\_j\_j+1}\}$, $y'_{m_j} = \{E_{target\_j+1}, E_{others\_j+1\_j+2}\}$, and $x'_{m_j} \approx x_{p_t}$.

In addition, when given a current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$, along with a set of ***K*** similar scenes, where $s_m = \{x'_{m_j}, y'_{m_j}\}, j \in [1, K]$, an attention vector of the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ towards ***K*** similar scenes will be computed via the element y_{p_t} in the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ and all the elements $y'_{m_j}, j \in [1, K]$, in the corresponding ***K*** similar scenes, denoted as $s_{p_t} = \{x_{p_t}, y_{p_t}\} \rightarrow a_t$ in Eq (1). Therefore, the attention vector a_t represents the observation from the element y_{p_t} in current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ towards all the corresponding elements $y'_{m_j}, j \in [1, K]$ in similar scenes.

$$\begin{cases} e_{tj} = v_a^T \tanh(W_{p_t} y_{p_t} + W_{m_j} y'_{m_j} + b_{p_t}) \\ \beta_{tj} = \dfrac{\exp(e_{tj})}{\sum_{j'=1}^{K} e_{tj'}} \\ a_t = \sum_{j=1}^{K} \beta_{tj} \cdot y'_{m_j} \leftarrow s_{p_t} = \{x_{p_t}, y_{p_t}\} \end{cases} \tag{1}$$

In this way, as shown in Figure 4, when given an input sequence $X_p = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ selected from X and the corresponding target sequence $Y_p = \{y_{p_1}, y_{p_2}, ..., y_{p_t}, p_t \in T\}$ which is the ground truth of prediction, ***t*** current scenes (i.e., $s_{p_t} = \{x_{p_t}, y_{p_t}\}$) will be created immediately. For each current scene, the period-offset searching method as shown in Figure 5 will be adopted to extract ***K*** similar scenes (i.e., $s_m = \{x'_{m_j}, y'_{m_j}\}$, $j \in [1, K]$) from the input time-series X, followed by the observation from y_{p_t} to y'_{m_j}, $j \in [1, K]$, so as to obtain the attention vector a_t of the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ derived from ***K*** similar scenes as defined in Eq (1). Furthermore, for the ***t*** current scenes, an attention-feature sequence can be obtained, denoted as $A = \{a_1, a_2, ..., a_t, t \in \mathrm{T}\}$. Such the attention-feature sequence can be considered as the auxiliary feature of the input sequence $X_p = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ to support following emotion prediction. Finally, the attention-feature sequence $A = \{a_1, a_2, ..., a_t, t \in \mathrm{T}\}$, the input sequence $X_p = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ along with the target sequence $Y_p = \{y_{p_1}, y_{p_2}, ..., y_{p_t}, p_t \in T\}$ will be aligned up by time step.

3.3. Two-layer encoder-decoder with emotion fusion and hybrid attention

Generally, the proposed model TEP2MP conducts text emotion prediction in a multi-participant communication scenario by the two-layer encoder-decoder prediction model as shown in Figure 6. First, the prediction model in Figure 6 consists of the encoding and decoding phase. Specifically, the prediction model encodes the input sequence $X_p = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ and the attention-feature sequence $A = \{a_1, a_2, ..., a_t, t \in \mathrm{T}\}$ via two-layer bi-directional LSTM. Here, at the encoding stage, the input sequence X_p and the attention-feature sequence A have been aligned up by time step. For decoder, on the one hand, the two-layer decoder decodes the input sequence X_p step by step to obtain the hidden state s_t^y at time step t. On the other hand, the hybrid attention mechanism has also been introduced to compute the context vector o_t corresponding to the attention-feature sequence $A = \{a_1, a_2, ..., a_t, t \in \mathrm{T}\}$ and the context vector c_t corresponding to the input sequence X_p. Such two context vectors will be merged by the gate switcher g to obtain another hidden state derived from the hybrid attention mechanism at different time steps, denoted as $g(o_t, c_t) \rightarrow s_t^a$. Finally, the prediction result $\widetilde{y}_{p_t} = \{\widetilde{E}_{target\_t+1}, \widetilde{E}_{others\_t+1\_t+2}\}$ will be output step by step according to the hidden state s_t^y and s_t^a.

Obviously, the predicted n-dimensional *affective vector* $\widetilde{E}_{target\_t+1}$ in the prediction result $\widetilde{y}_{p_t} = \{\widetilde{E}_{target\_t+1}, \widetilde{E}_{others\_t+1\_t+2}\}$ is corresponding to the conversation text published by the target speaker at t+1 time step, which should be close to the element E_{target_t+1} in the sub-sequence

$y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$. Here, y_{p_t} is an element in the target sequence Y_p, or the ground truth of the predicted $\widetilde{E}_{target\_t+1}$.

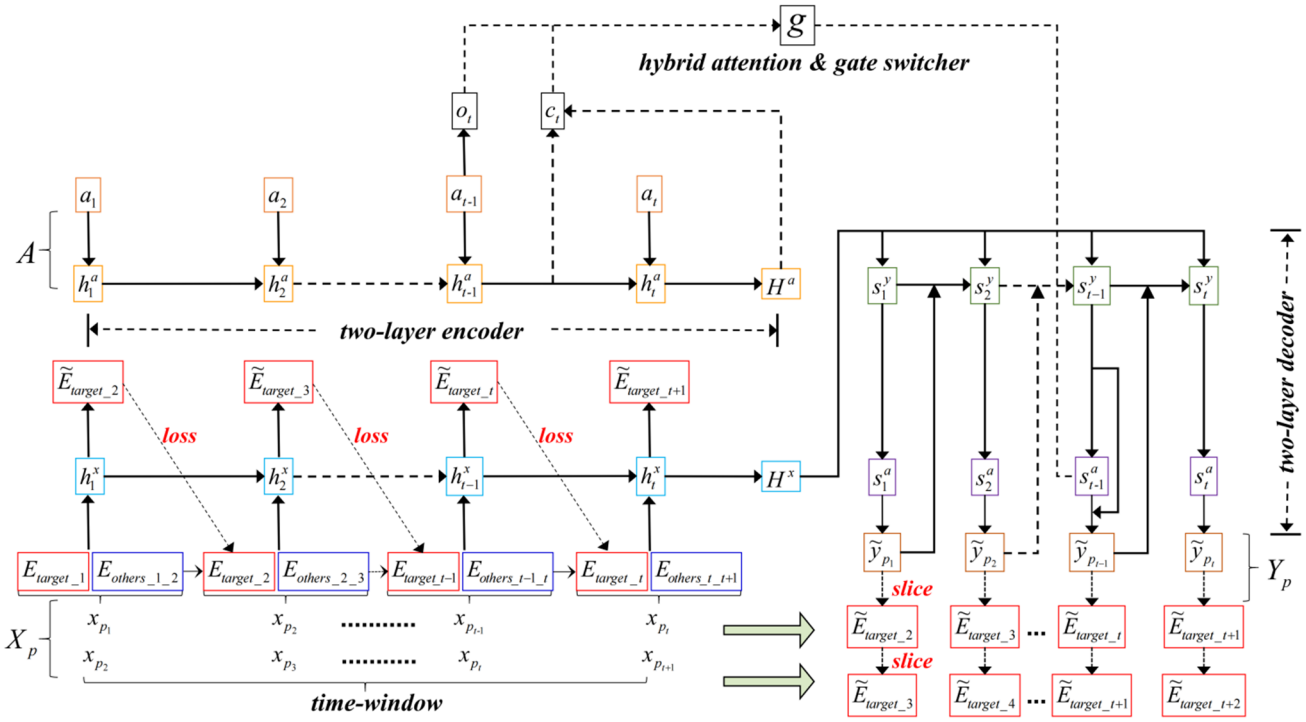


Figure 6. The structure of the two-layer encoder-decoder prediction model.

Moreover, in Figure 6, a time-window spanning t time steps will be set before prediction, based on which the two-layer encoder-decoder will select the corresponding input sequence $X_p = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$ and target sequence $Y_p = \{y_{p_1}, y_{p_2}, ..., y_{p_t}, p_t \in T\}$ (the ground truth of prediction) from the given time-series $X = \{x_1, x_2, ..., x_i | 0 < i + 1 < T\}$, where $x_i = \{E_{target\_i}, E_{others\_i\_i+1}\}$. Here, $y_{p_t} = x_{p_{t+1}}$, $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$. In other words, the target sequence Y_p is obtained by shifting the input sequence X_p one time step forward, along the direction to t+1. And, the n-dimensional *affective vector* E_{target_t+1} in $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$ is corresponding to the emotion of the text to be published by the target speaker at time step t + 1. Obviously, the target sequence Y_p can be considered as the ground truth of the predicted sequence $\widetilde{Y}_p = \{\widetilde{y}_{p_1}, \widetilde{y}_{p_2}, ..., \widetilde{y}_{p_t}, p_t \in T\}$ output by the tow-layer decoder. Furthermore, as shown at the bottom of Figure 6, when the time-window moves forward by the fixed stride, the input sequence X_p will move towards the end of the given time-series X. As a result, the emotion of the text to be published by the target speaker from time step $t + 2 \rightarrow t + N$ will be predicted in sequence.

$$h_t^x = LSTM(FC(x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}), h_{t-1}^x) \rightarrow \widetilde{E}_{target\_t+1} \tag{2}$$

$$l_1 = \frac{1}{t}\sqrt{\sum_1^t (E_{target\_t+1} - \widetilde{E}_{target\_t+1})^2} \tag{3}$$

In addition, as shown in Figure 6, after selecting the input sequence X_p, the emotion fusion mechanism will be incorporated into the first-layer encoder. Here, the first-layer encoder is the LSTM

network. And, as defined in Eq (2), the emotion fusion mechanism will merge the *n*-dimensional *affective vector* E_{target_t} and $E_{others\_t\_t+1}$ in $x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}$ by fully connected layer, after which the emotion fusion result will be further processed by LSTM (the second-layer encoder), denoted as $\widetilde{E}_{target\_t+1}$. Here, the emotion fusion result $\widetilde{E}_{target\_t+1}$ will be forced to approach the *n*-dimensional *affective vector* E_{target_t+1} (the *affective vector* of the conversation text to be published by the target speaker at next time step $t + 1$). Finally, the loss l_1 defined in Eq (3) will be adopted to measure the difference between emotion fusion result and the ground truth of the *affective vector* input at the next time step. In this way, by incorporating the emotion fusion mechanism into the first-layer encoder and forcing the emotion fusion result to approach the ground truth of the *affective vector* corresponding to the conversation text published by the target speaker at next time step, the prediction model in Figure 6 will be guided to learn emotion interaction among the target speaker and all the other participants so as to capture the emotion stimulation to the target speaker triggered by the other participants.

Afterwards, when given the input sequence $X_p = \{x_{p_1}, x_{p_2}, ..., x_{p_t}, p_t \in T\}$, a hidden state sequence will be output by the first-layer encoder (i.e., LSTM), denoted as $H^x = \{h_1^x, h_2^x, ..., h_t^x\}$. Similarly, the second-layer encoder (i.e., another LSTM) will encode the attention-feature sequence $A = \{a_1, a_2, ..., a_t, t \in T\}$, derived from the similar scene searching process illustrated in Figure 5 so as to obtain the corresponding hidden state sequence $H^a = \{h_1^a, h_2^a, ..., h_t^a\}$.

$$s_t^y = LSTM(\widetilde{y}_{p_{t-1}}, s_{t-1}^y, H^x) \tag{4}$$

At the decoding stage, as shown in Figure 6, first, when given the hidden state sequence $H^x = \{h_1^x, h_2^x, ..., h_t^x\}$ derived from the input sequence X_p, the top-layer decoder (i.e., LSTM) will decode the hidden state step by step, denoted as s_t^y at time step t as defined in Eq (4). Here, in Eq (4), $\widetilde{y}_{p_{t-1}}$ is the prediction result output by the decoder at time step t-1.

$$\begin{cases} o_t = \sigma(W_a a_t + b_t) \\ c_t = \gamma_t \cdot \tanh(H^x) \\ \gamma_t = Attn(W_H H^x + W_s^\gamma s_t^y + b_\gamma) \end{cases} \tag{5}$$

Moreover, as shown in Figure 6, a hybrid attention mechanism has been adopted. Specifically, as defined in Eq (5), the context vector o_t at time step t will be computed based on the attention vector a_t, when given the hidden state sequence $H^a = \{h_1^a, h_2^a, ..., h_t^a\}$ derived from the attention-feature sequence $A=\{a_1, a_2, ..., a_t, t \in T\}$. Similarly, the context vector c_t at time step t will be computed by aggregating the hidden state sequence $H^x = \{h_1^x, h_2^x, ..., h_t^x\}$ weighted by the attention-weight γ. Such the attention-weight γ is involved with the hidden state s_t^y. In this way, as shown in Figure 6, when decoding at time step t, the hidden state s_t^a will be computed by absorbing the context vectors o_t and c_t via the gate switcher g, as defined in Eq (6). And, the emotion prediction result $\widetilde{y}_{p_t} = \{\widetilde{E}_{target\_t+1}, \widetilde{E}_{others\_t+1\_t+2}\}$ output at time step t will be computed based on the two hidden states s_t^y and s_t^a.

$$\begin{cases} g = \sigma(W_h h_t^a + W_s^g s_t^y + b_g) \\ s_t^a = (1-g) \otimes o_t + g \otimes c_t \\ \widetilde{y}_{p_t} = \sigma(W_s^y s_t^y + W_s^a s_t^a + b_y) \end{cases} \tag{6}$$

Finally, as shown in Figure 6, an operation $slice(\widetilde{y}_{p_t}) \rightarrow \widetilde{E}_{target\_t+1}$ has been defined, where $\widetilde{y}_{p_t} = \{\widetilde{E}_{target\_t+1}, \widetilde{E}_{others\_t+1\_t+2}\}$. Moreover, similar to Eq (3), the loss l_2 is computed to measure the difference between the emotion prediction result $\widetilde{E}_{target\_t+1}$ and the corresponding ground truth E_{target_t+1}. Consequently, the two-layer encoder-decoder model will be trained by $LOSS=\lambda_1 l_1+\lambda_2 l_2$ to enhance the emotion prediction performance. Here, l_1 represents the "loss" on emotion fusion (defined in Eq (3)), aiming at guiding the two-layer encoder to capture the emotion interaction among multiple participants. And, l_2 represents the "loss" on emotion prediction of the two-layer decoder, aiming at forcing the emotion prediction result decoded at time step t to be "close" to the true n-dimensional *affective vector* corresponding to $t + 1$ time step.

4. Experimental results

In this section, the experimental results on the proposed text emotion prediction model TEP2MP will be analyzed. The impact of the time-window size, time-window stride, the searching scope of similar scenes, the capacity of the pool to store candidate similar scenes, the hybrid attention and emotion fusion mechanism incorporated in the two-layer encoder-decoder prediction model will also be discussed. Moreover, the existing time-series models which can be adapted to the text emotion prediction problem have also been compared. All the involved experiments are conducted via 5-fold cross validation and the average of three times running has been presented. In addition, when it comes to the proposed text emotion prediction TEP2MP, a four-layer bidirectional LSTM has been adopted to construct the two-layer encoder-decoder prediction model illustrated in Figure 6, and RELU [33] is used for neural activation trained by the RMSPropOptimizer [34] with learning rate set as 7×10^{-3}. The proposed model TEP2MP and the compared methods are all implemented by Python 3.7 and Tensorflow 1.15, on GPU, NVIDIA GeForce GTX 1080Ti, 11GB.

4.1. Dataset and emotion annotation

In order to reproduce a multi-participant text conversation scenario, four collections of movie lines are adopted. In terms of the participants of conversation, we take the main movie character as the target speaker and the remaining as other participants. The specific information of the four text collections are shown in Table 1. Here, for the following part of this section, "Gump." is used to represent dataset 1, "Shawshank." for dataset 2, "Scent." for dataset 3, "Bovary." for dataset 4 due to the limited space.

In terms of the emotion category, we annotate the emotion of texts contained in above four datasets as [*Target_Flag* (0/1), *Emotion_Index* (1~6)]. Here, the *Emotion_Index* (1~6) represents "Happy", "Sad", "Angry", "Anxious", "Surprise" and "Other" respectively. The emotion distribution on the involved datasets is shown in Table 2. The annotation processing is completed by five researchers with experience in Natural Language Processing and Time-Series Prediction. And, the

majority voting mechanism has been adopted to determine the final emotion of each text with the agreement from at least three annotators. Based on above annotation, the *Affective Space Mapping* illustrated in Figure 2 has been conducted, in which word embeddings with 256 dimensions have been used. In addition, the accuracy of the emotion classifier (i.e., Bi-LSTM) has been maintained by 80% and more. The *affective vectors* classified incorrectly are revised by the global emotion EIP. The specific statistics on emotion annotation has been demonstrated in Figure 7. More specifically, Figure 7(a) represents the annotation distribution on six emotion categories (i.e., Happy, Sad, Angry, Anxious, Surprise and Other) in terms of the conversation texts posted by all the participants, while Figure 7(b) represents the emotion distribution of the conversation texts particularly posted by the target speaker. Correspondingly, Figure 7(c),(d) represent the annotation shift on above six emotion categories with regard to Figure 7 (a),(b) respectively.

Table 1. Experimental datasets for testing TEP2MP.

No.	Movie Name	Prediction Target	Movie Duration	#Total Text	#Text of Target Speaker
1	Forrest Gump	Forrest Gump	142 min	2054	905
2	Shawshank Redemption	Ellis Boyd Redding	142 min	1729	590
3	Scent a of Woman	Lieutenant Frank	157 min	2226	1165
4	I Am Not Madame Bovary	XueLian Li	138 min	2323	505

Table 2. Emotion category distribution after annotation of different datasets.

No.	Dataset	Statistical Scope	Happy (%)	Sad (%)	Angry (%)	Anxious (%)	Surprise (%)	Other (%)
1	Gump.	ALL Conversation Participants	40.70	17.96	10.81	18.16	1.36	11.00
2	Shawshank.		42.34	12.78	21.75	12.67	0.29	10.18
3	Scent.		31.18	12.8	28.26	18.28	1.08	8.40
4	Bovary.		13.78	6.03	43.65	-------	-------	36.55
No.	Dataset	Statistical Scope	Happy (%)	Sad (%)	Angry (%)	Anxious (%)	Surprise (%)	Other (%)
1	Gump.	Only Target Speaker	47.29	22.54	0.88	17.35	0.55	11.38
2	Shawshank.		58.31	19.83	5.93	7.97	0.51	7.46
3	Scent.		35.45	18.03	29.18	4.03	0.34	12.96
4	Bovary.		9.11	7.33	74.06	-------	-------	9.50

In terms of the emotion category, we annotate the emotion of texts contained in above four datasets as [*Target_Flag* (0/1), *Emotion_Index* (1~6)]. Here, the *Emotion_Index* (1~6) represents "Happy", "Sad", "Angry", "Anxious", "Surprise" and "Other" respectively. The emotion distribution on the involved datasets is shown in Table 2. The annotation processing is completed by five researchers with experience in Natural Language Processing and Time-Series Prediction. And, the majority voting mechanism has been adopted to determine the final emotion of each text with the agreement from at least three annotators. Based on above annotation, the *Affective Space Mapping* illustrated in Figure 2 has been conducted, in which word embeddings with 256 dimensions have been used. In addition, the accuracy of the emotion classifier (i.e., Bi-LSTM) has been maintained by 80% and more. The *affective vectors* classified incorrectly are revised by the global emotion EIP. The specific statistics on emotion annotation has been demonstrated in Figure 7. More specifically, Figure 7 (a) represents the annotation distribution on six emotion categories (i.e., Happy, Sad, Angry, Anxious, Surprise and Other) in terms of the conversation texts posted by all the participants, while Figure 7 (b) represents the emotion distribution of the conversation texts particularly posted by the target speaker.

Correspondingly, Figure 7(c),(d) represent the annotation shift on above six emotion categories with regard to Figure 7(a),(b) respectively.

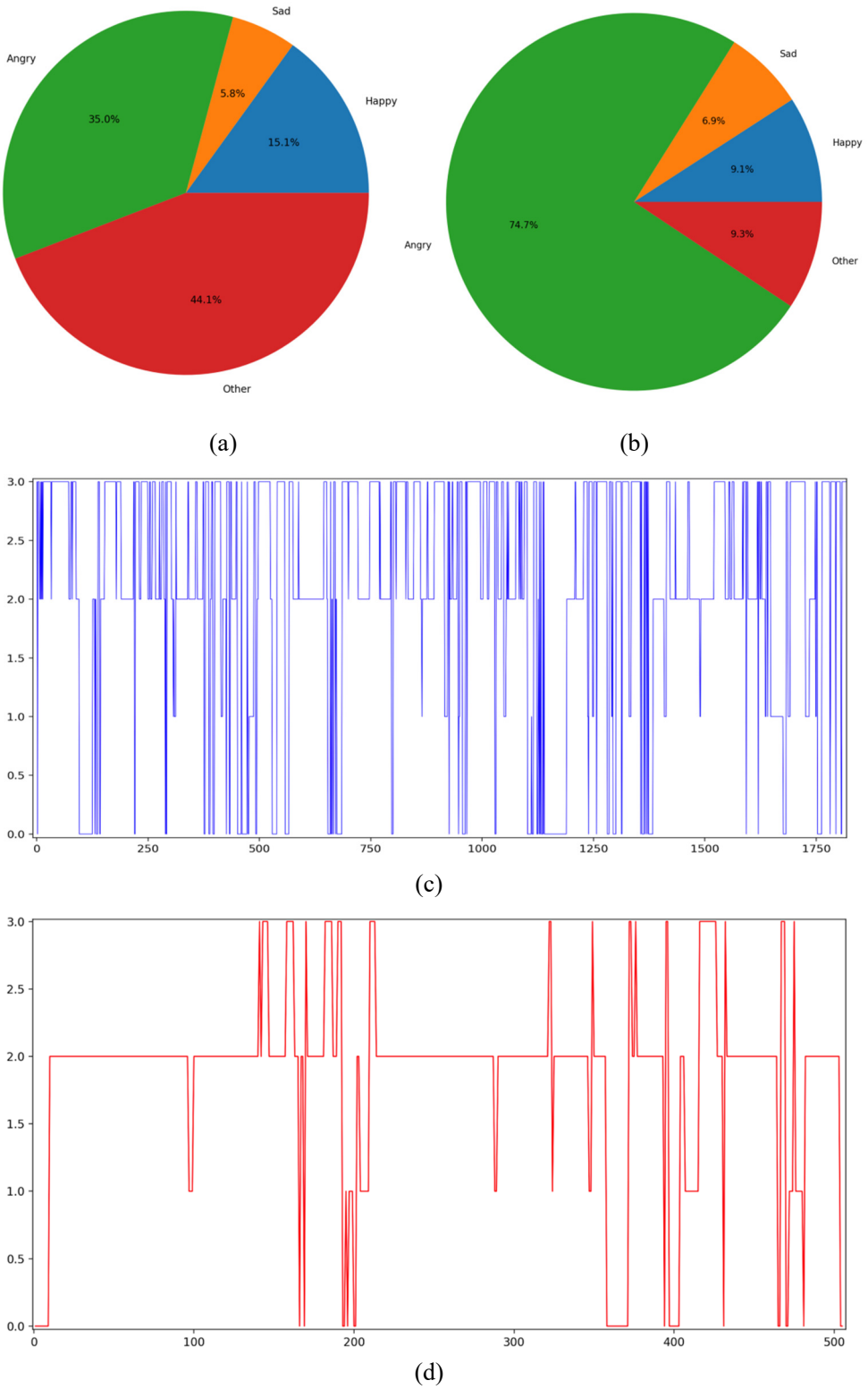


Figure 7. The visualization of emotion distribution after annotation: (a) The emotion distribution of the initial text set D. (b) The emotion distribution of target participant. (c) The emotion tendency on the initial text set D. (d) The emotion tendency on target participant.

4.2. The impact of the similar scene searching to the emotion prediction

When given the input sequence X_p and the target sequence Y_p (the ground truth of prediction), ***t*** current scenes will be created. After the similar scene searching process illustrated in Figure 4 and Figure 5, the period-offset searching method will seek ***K*** similar scenes for each current scene. Such the ***K*** similar scenes will be aggregated into an attention vector a_t, then an attention-feature sequence $A= \{a_1, a_2, ..., a_t, t \in \mathrm{T}\}$ will be computed corresponding to the ***t*** current scenes. Here, the attention-feature sequence $A= \{a_1, a_2, ..., a_t, t \in \mathrm{T}\}$ will be aligned up to the input sequence X_p by time steps, which is further taken as the auxiliary features input into the encoder at the second-layer shown in Figure 6. Moreover, for each current scene in Figure 5, the period-offset searching method will expand a searching scope $L/8$ from left and right respectively after a pivot element has been found out by different time spans. When it comes to the experiments involved in this section, **5 *minute*** is adopted to search ***K*** similar scenes according to the timestamp of each text in datasets.

Table 3. The impact of similar scene search mechanism on the text emotion prediction accuracy of TEP2MP model ($K = 5$, $\lambda_1 = \lambda_2 = 0.5$).

Dataset	Optimal Time Window	The Searching Scope L and Optimal Value	Accuracy	F1-Score
1-Gump	$win = 2$	$L = win \times 0 = 0$ **(No Similar Scene Search)**	**0.602039**	**0.602931**
		$L = win \times 4 = 8$	0.652314	0.662319
		$L = win \times 16 = 32$	0.782182	0.783418
		L = win × 64 = 128	**0.832741**	**0.826231**
		$L = win \times 256 = 512$	0.801248	0.803583
	The improvement on prediction precision（%）		+27.70%	+27.03%
2-Shawshank	$win = 3$	$L = win \times 0 = 0$ **(No Similar Scene Search)**	**0.623042**	**0.624617**
		$L = win \times 4 = 12$	0.702144	0.704294
		$L = win \times 16 = 48$	0.762024	0.770214
		***L = win* × 64 = 192**	**0.849317**	**0.846003**
		$L = win \times 256 = 768$	0.810423	0.814956
	The improvement on prediction precision（%）		+26.64%	+26.17%
3-Scent	$win = 3$	$L = win \times 0 = 0$ **(No Similar Scene Search)**	**0.759301**	**0.751280**
		$L = win \times 4 = 12$	0.810493	0.819203
		$L = win \times 16 = 48$	0.839301	0.830382
		***L = win* × 64 = 192**	**0.845665**	**0.841092**
		$L = win \times 256 = 768$	0.823540	0.824765
	The improvement on prediction precision（%）		+11.37%	+11.95%
4-Bovary	$win = 4$	$L = win \times 0 = 0$ **(No Similar Scene Search)**	**0.798394**	**0.793829**
		$L = win \times 4 = 16$	0.813894	0.812405
		$L = win \times 16 = 64$	0.853919	0.852834
		L = win × 64 = 256	**0.866187**	**0.855451**
		$L = win \times 256 = 1024$	0.859301	0.853526
	The improvement on prediction precision（%）		+7.83%	+7.20%

In order to measure the impact of $L/8$ searching scope on the emotion prediction performance, the

capacity of the pool to store similar scenes is set as $K = 5$ and the optimal value of L is selected by grid search, stopping at the first option when the prediction performance decreases. More importantly, for the $LOSS = \lambda_1 l_1 + \lambda_2 l_2$ used for training the TEP2MP, λ_1 and λ_2 are set as 0.5. Particularly, when $L = 0$, no similar scene will be searched for each current scene. The specific emotion prediction performance is shown in Table 3.

As shown in Table 3, first, the optimal size of time-window on each dataset is selected by grid search. Second, based on the optimal time window size, taking the main character on each dataset as the target speaker and the remaining characters as the other participants, the optimal searching scopes are listed as 1-Gump, *win* = 2, $L = 128$; 2-Shawshank, *win* = 3, $L = 192$; 3-Scent, *win* = 3, $L = 192$; 4-Bovary, *win* = 4, $L = 256$. Third, compared with the case when $L = 0$ (i.e., no similar scene has been searched), it can be observed that when adopting the similar scene searching mechanism, the precision on text-emotion prediction has been improved by approximately 10%~30%. The above phenomenon can be ascribed to the reason that the similar scene searched by the period-offset method is assembled in the same way as that of current scene, like $x_{p_t} = \{E_{target\_t}, E_{others\_t\_t+1}\}$ and $y_{p_t} = \{E_{target\_t+1}, E_{others\_t+1\_t+2}\}$ in $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ for the current scene, $x'_{m_j} = \{E_{target\_j}, E_{others\_j\_j+1}\}$ and $y'_{m_j} = \{E_{target\_j+1}, E_{others\_j+1\_j+2}\}$ in $s_m = \{x'_{m_j}, y'_{m_j}\}$ for the similar scene. The current scene and the similar scene have the similar tendency on emotion category shift (i.e., sad→angry). Therefore, before prediction, the proposed model TEP2MP has additionally collected auxiliary emotion features for the input sequence X_p, according to the given current scene (sub-sequence). Moreover, the similar scenes have been converted into attention-feature sequence as defined in Eq (1), which are further fed into the two-layer encoder shown in Figure 6. Consequently, when adopting the similar scene searching mechanism, the text emotion precision has been improved effectively.

Based on the optimal searching scope L selected in Table 3, the optimal capacity (K) of the pool to store similar scenes is also selected in Table 4. For the $LOSS = \lambda_1 l_1 + \lambda_2 l_2$ used for training TEP2MP, λ_1 and λ_2 are set as 0.5. Here, the capacity K represents the amount of similar scenes searched for each given current scene. It can be observed that the alteration on the pool capacity K can affect the performance on text emotion. In addition, as shown in Table 4, the emotion prediction performance can be further enhanced when assigned with the optimal pool capacity K (i.e., presented as font in Table 4) compared to that listed in Table 3. Finally, the optimal configuration on each dataset for text emotion prediction are listed as: 1-Gump, *win* = 2, $L = 128$, $K = 7$; 2-Shawshank, *win* = 3, $L = 192$, $K = 5$; 3-Scent, *win* = 3, $L = 192$, $K = 7$; 4-Bovary, *win* = 4, $L = 256$, $K = 9$.

4.3. Comparison on the Performance of the Text Emotion Prediction

Based on the optimal time-window size, window stride, searching scope $L/8$ and the pool capacity K, our proposed text emotion prediction TEP2MP is further compared with other existing time-series prediction models which can be adapted to the text emotion prediction problem.

First, in order to analyze the effectiveness of the similar scene searching, emotion fusion and hybrid attention mechanism incorporated into TEP2MP, all the compared methods are considered as "black box", whose inner principle has not been adapted. Second, the compared methods include DDAE [35], Restimator_EE [36], DA_RNN [37], R2N2 [38] and CNN_Series [39]. Specifically, DDAE [35] contains three hidden layers using "RELU" as neural activation, conducting unsupervised learning by Auto-Encoder [40]. Restimator_EE [36] uses the Convolutional Neural Network (CNN) and the multi-layer LSTM for time-series prediction. DA_RNN [37] constructs two context vectors by

two-layer attention mechanism via encoder-decoder structure. R2N2 [38] computes the residual error by multiple LSTM and the time-series prediction is conducted by minimizing the residual error. CNN_Series [39] contain three-layer one-dimensional convolution and the "Dilation Rate" in each layer is set as 1, 2 and 4 respectively with a receptive field set as 8.

Table 4. The selection of the optimal pool capacity (K) to store the similar scenes ($\lambda_1 = \lambda_2 = 0.5$).

Dataset	Optimal Time Window	The Searching Scope L and Optimal Value	Candidate Pool Capacity K	Accuracy	F1-Score
1-Gump	$win = 2$	$L = win \times 64 = 128$	1	0.792142	0.799324
			3	0.823721	0.810923
			5	0.832741	0.826231
			7	**0.849027**	**0.841416**
			9	0.847152	0.821397
2-Shawshank	$win = 3$	$L = win \times 64 = 192$	1	0.823726	0.831795
			3	0.839725	0.841782
			5	**0.849317**	**0.846003**
			7	0.812799	0.818972
			9	0.773142	0.793372
3-Scent	$win = 3$	$L = win \times 64 = 192$	1	0.821652	0.814601
			3	0.832954	0.827952
			5	0.845665	0.841092
			7	**0.861324**	**0.850179**
			9	0.856667	0.845873
4-Bovary	$win = 4$	$L = win \times 64 = 256$	1	0.844017	0.835784
			3	0.848148	0.845341
			5	0.866187	0.855451
			7	0.886531	0.886028
			9	**0.892712**	**0.894317**

Third, in terms of our proposed TEP2MP model, as shown in Figure 6, in the first-layer encoder, the *affective vectors* corresponding to the target speaker and all the other participants are merged together, whose result is forced to approach the *affective vector* of the target speaker input at the next time step. However, in terms of the compared methods including DDAE [35], Restimator_EE [36], DA_RNN [37], R2N2 [38] and CNN_Series [39], only the *affective vector* corresponding to the target speaker is involved without the introduce of emotion fusion and similar scene searching mechanism. In addition, except the CNN_Series [37] which only takes in the emotion category sequence (i.e., like [0, 1, 2, 3, 0, 2, ...]), all the other compared methods all require the n-dimensional *affective vector* sequence. Here, the n-dimensional *affective vector* is provided by the *Affective Space Mapping*, as shown in Figure 2. The specific emotion prediction performance is listed in Table 5. Here, for the $LOSS=\lambda_1 l_1+\lambda_2 l_2$ used for training the TEP2MP, λ_1 and λ_2 are set as 0.5.

As shown in Table 5, the text emotion prediction performance of our proposed model TEP2MP all exceeds that of the other compared prediction models, improved by 4.99, 6.09, 5.18 and 3.94% respectively in terms of the prediction accuracy. Such the advantage can be ascribed to the reason that, on the one hand, the similar scene searching process has been incorporated into TEP2MP. In other words, when given the input sequence X_p, multiple similar scenes, or sub-sequences of *affective vectors*

with similar emotion tendency to that of the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ have been extracted across the entire conversation history, which are taken as the auxiliary features to support following emotion prediction. In addition, the proposed model TEP2MP converts the $\boldsymbol{K}$ similar scenes of the current scene $s_{p_t} = \{x_{p_t}, y_{p_t}\}$ into an attention vector a_t as defined in Eq (1), which can be considered as the observation on the context of emotion category shift across the whole conversation history, so as to enrich the features available for predicting the final emotion category.

Table 5. Comparison on the text emotion prediction performance between TEP2MP and other existing sequence prediction methods ($\lambda_1 = \lambda_2 = 0.5$).

Dataset	Methods	Window Size	Window Stride	Similar Scene	Emotion Fusion	Accuracy	F1-Score	Test Efficiency (*second*)
1-Gump	DDAE	58	41			0.804483	**0.809076**	7.022135
	Restimator_EE	25	5			0.793122	0.764111	47.231311
	DA_RNN	23	1	×	×	0.780848	0.774153	12915.61279
a	R2N2	12	9			**0.806690**	0.802319	25.42421
	CNN_Series	2	2			0.805897	0.803660	141.690461
	TEP2MP	2	1	√	√	**0.849027**	**0.841416**	84.626693
The improvement on prediction precision (%)						+4.99%	+3.84%	
2-Shawshank	DDAE	33	19			0.751064	0.676932	8.326790
	Restimator_EE	38	33			0.780627	0.787936	39.904229
	DA_RNN	9	1	×	×	0.709684	0.700783	1902.978992
	R2N2	7	3			**0.797596**	**0.790690**	23.530976
	CNN_Series	4	2			0.762712	0.719408	232.465386
	TEP2MP	3	3	√	√	**0.849317**	**0.846003**	80.372754
The improvement on prediction precision (%)						+6.09%	+6.54%	
3-Scent	DDAE	3	2			0.770115	0.714166	8.34821
	Restimator_EE	31	31			0.805665	0.811109	39.66678
	DA_RNN	13	2	×	×	0.685516	0.678652	3421.798897
	R2N2	54	38			**0.816667**	**0.815873**	19.577201
	CNN_Series	4	3			0.808083	0.808417	129.559072
	TEP2MP	3	2	√	√	**0.861324**	**0.850179**	96.475358
The improvement on prediction precision (%)						+5.18%	+4.04%	
4-Bovary	DDAE	53	34			0.836187	0.835451	7.828369
	Restimator_EE	26	2			0.809082	0.741626	132.577562
	DA_RNN	6	2	×	×	0.841528	0.844735	595.604229
	R2N2	5	5			**0.857500**	**0.859545**	13.721418
	CNN_Series	29	24			0.850000	0.858095	725.085449
	TEP2MP	4	3	√	√	**0.892712**	**0.894317**	86.917757
The improvement on prediction precision（%）						+3.94%	+3.89%	

On the other hand, in Figure 6, at the encoding stage, the *affective vectors* corresponding to the conversation texts published by the target speaker and all the other participants have been merged by time step via the emotion fusion mechanism. In this way, the stimulation to the emotion change of the

target speaker can be learned. Moreover, as shown in Figure 6, compared with other methods, the proposed model TEP2MP uses the hidden state from the input sequence X_p and the counterpart from the attention-feature sequence A to decode the prediction result at time step t, with the help of the gate switcher g. In other words, the input sequence X_p is decoded at much finer granularity. Moreover, the emotion prediction results, including the one output by the emotion fusion mechanism at the encoding stage (related to $\boldsymbol{l_1}$) and the counterpart output by the hybrid attention mechanism at the decoding stage (related to $\boldsymbol{l_2}$), are all forced to be "close" to the ground truth (i.e., the *affective vector* input at the next time step), based on which the final $LOSS=\lambda_1 l_1+\lambda_2 l_2$ has been adopted to train our proposed text emotion prediction model TEP2MP. Consequently, TEP2MP has achieved the best overall text emotion prediction performance compared to other methods.

(a) Prediction Performance on dataset Gump.

(b) Prediction Performance on dataset Shawshank.

(c) Prediction Performance on dataset Scent.

(d) Prediction Performance on dataset Bovary.

Figure 8. The comparison on the tendency of the text emotion prediction performance ($\lambda_1 = \lambda_2 = 0.5$).

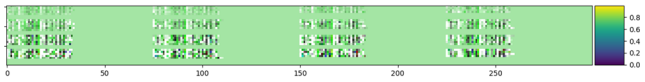
In addition, in Table 5, the test efficiency (measured by *second*) has also been compared. Here, the test efficiency means the time spent after data preparation before outputting the final prediction result. It can be observed from Table 5 that although extra data processing mechanism including similar scene searching, emotion fusion and hybrid attention have been incorporated into TEP2MP, yet the time spent on testing is still controlled at an acceptable level, less than that of the DA_RNN and CNN_Series. Moreover, the test efficiency of TEP2MP is worse than that of DDAE and R2N2, but the corresponding text emotion performance of TEP2MP is better than that of DDAE and R2N2.

Finally, as shown in Figure 8, our proposed text emotion prediction model TEP2MP has also been compared with several classical regression models, including Decision Tree (DT_R), Linear Regression (Linear_R), SVM Regression (SVM_R), KNN Regression (KNN_R), and Random Forest Regression (RF_R). All the compared regression models are implemented by the high-level AIP in Scikit-learn machine learning framework (*https://scikit-learn.org/stable*). Particularly, due to the requirement of the API in Scikit-learn, all the regression models only take in the *Emotion_Index* sequence like [0, 2, 4, 5, 1...] which is obtained by emotion annotation involved in *Affective Space Mapping*. It can be observed in Figure 8 that the proposed text emotion prediction model TEP2MP has achieved the best performance (i.e., Accuracy and F1-Score) on all the datasets, which can reflect the fact that the *n*-dimensional *affective vectors* obtained by the *Affective Space Mapping* contain more features on text emotion when compared with the *Emotion_Index* sequence.

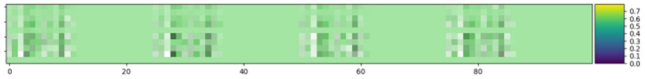
4.4. More discussion on TEP2MP

Finally, in this section, focusing on the dataset *Bovary*., we delve into the inner processing of the proposed text emotion prediction model TEP2MP. First, as demonstrated in Figure 9 (*a-i*) which takes the input basic data-unit $< E_{target\_1}, E_{others\_1\_2} >$ as an example, it can be observed that when it comes to the encoder in TEP2MP, the trivial features has blurred with the strengthening on more critical ones after processed by the fully connected layer, as shown in Figure 9 (*a-ii*). Then, based on the fully connected layer, the first-layer encoder (i.e., the LSTM component) will output the hidden state h^x which reflects the emotion distribution by considering the target speaker and all the other participants simultaneously (i.e., Figure 9 (*a-iii*)). At last, by Eq (1), the emotion distribution $\widetilde{E}_{target\_2}$ on target speaker can be output shown in Figure 9 (*a-iv*). Compared with the ground truth E_{target_2} shown in Figure 9 (*d*), the emotion prediction result $\widetilde{E}_{target\_2}$ in Figure 9 (*a-iv*) output at the encoder side at time step 1 still has evident difference to the ground truth.

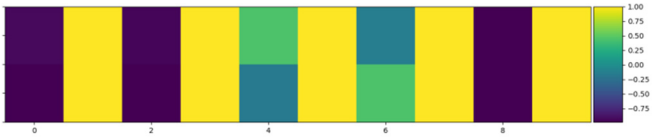
Second, when it comes to the decoder, Figure 9 (*b-i*) demonstrates the visualization on the hidden state sequence $H^x = \{h_1^x, h_2^x, ..., h_t^x\}$ computed by the encoder. Figure 9 (*b-ii*) and Figure 9 (*b-iii*) present the visualization on the hidden state s_1^y and s_1^a at the time Step 1. It can be found that such two hidden states concentrate more on the critical features at time step 1. Finally, Figure 9 (*b-iv*) demonstrates the visualization on emotion distribution of target speaker computed via Eq (6) with the help of the gate switcher *g* by merging the context vectors from the input sequence X_p and the attention-feature sequence *A*. Obviously, the emotion distribution output by decoder (shown in Figure 9 (*b-iv*)) is close to the ground truth E_{target_2} shown in Figure 9 (*d*), better than the emotion distribution output by the encoder at step 1 shown in Figure 9 (*a-iv*).



i) The input of the $< E_{target\_1}, E_{others\_1\_2} >$ at encoder side



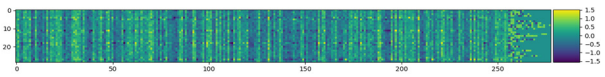
ii) The fusion result output by FC of $< E_{target\_1}, E_{others\_1\_2} >$



iii) The hidden state h_1^x derived from LSTM by Eq (1)

iv) The $\widetilde{E}_{target\_2}$ output by encoder at time step 1

(a) Visualization at encoder side.



i) The hidden state sequence H^x conveyed to decoder

ii) The hidden state s_1^y output at time step 1 in decoder

iii) The hidden state s_1^a output at time step 1 in decoder

iv) The visualization of $\widetilde{E}_{target\_2}$ output by the decoder at time step 1

(b) Visualization at decoder side.

(c) The visualization of $\widetilde{E}_{target\_2}$ output by the decoder at time step 1.

(d) The visualization of the ground truth E_{target_2} at time step 1.

Figure 9. The visualization of the *affective vector* and corresponding hidden state in the two-layer encoder-decoder of TEP2MP ($L = 256$, $K = 9$, $\lambda_1 = \lambda_2 = 0.5$ on the dataset Bovary.).

Table 6. The optimal (λ_1, λ_2) combination selection for the LOSS used for training TEP2MP ($L = 256$, K=9, on the dataset Bovary).

λ_1-Emotion Fusion	λ_2-Emotion Prediction	F1-Score ↑	Accuracy ↑	RMSE ↓	MAE ↓
0.1	0.9	0.912319	0.924483	0.179493	0.119514
0.3	**0.7**	**0.941667**	**0.942778**	**0.179313**	**0.119416**
0.5	0.5	0.892712	0.894317	0.179490	0.119519
0.7	0.3	0.894281	0.896427	0.179452	0.119502
0.9	**0.1**	**0.926565**	**0.938333**	**0.179448**	**0.119490**

It is also worthwhile to ascertain the optimal (λ_1, λ_2) combination in terms of the LOSS used to train TEP2MP. As shown in Table 6, the λ_1 related to the emotion fusion and the λ_2 related to the

emotion prediction vary from 0.1 to 0.9. It can be found that when λ_1 and λ_2 increase, the emotion prediction performance (i.e., such as F1-Score and Accuracy) presents to be better, higher than that obtained when $\lambda_1 = \lambda_2 = 0.5$. Moreover, the prediction error (i.e., RMSE [34] and MAE [34]) also presents to be lower. Specifically, in Table 6, it can be observed that a higher value the λ_2 (related to emotion prediction) has, a better emotion prediction performance will be obtained. In addition, the (0.3, 0.7) combination presents to be the best combination for training the TEP2MP model.

In addition, as shown in Figure 6, the proposed model TEP2MP conducts text emotion prediction in the multi-participant scenario via two-layer encoder-decoder model. In terms of the encoder-decoder structure, two critical mechanisms have been adopted which are beneficial to the improvement of text emotion prediction performance. Specifically, the *emotion fusion mechanism* merges the *affective vectors* corresponding to the target speaker and all the other participants at the first-layer encoder to learn emotion interaction among different participants. And, the similar scene search process illustrated in Figure 5 extracts sub-sequences with similar emotion tendency to that of the given current scene. The similar scenes are then converted into attention-feature sequence A, considered as the auxiliary features for decoding. Based on the attention-feature sequence A, the *hybrid attention mechanism* has been adopted to merge the context vectors o_t and c_t by the switcher gate g. Such two context vectors are derived from the hidden state sequence of the attention-feature sequence A and the hidden state sequence H^x output by the first-layer encoder, enriching the context information available for decoding. Consequently, in Table 7, an ablation study has been conducted on *emotion fusion mechanism* and *hybrid attention mechanism*, so as to measure the effectiveness of above two mechanisms on the text emotion prediction performance.

Table 7. The ablation study of emotion fusion and hybrid attention on the prediction performance of TEP2MP ($L = 256$, $K = 9$, on the dataset Bovary.)

Combo.	Emotion Fusion + Target & Other Participants	Similar Scene Search + Hybrid Attention	F1-Score ↑	Accuracy ↑	RMSE ↓	MAE ↓
1	×	×	0.711563	0.716046	0.216359	0.141079
2	√	×	**0.828261**	**0.849913**	**0.210058**	**0.136963**
3	×	√	0.880690	0.887596	0.179525	0.119538
4	√	√	**0.892712**	**0.894317**	**0.179490**	**0.119519**
The improvement rate between **Combo. 4** and **Combo. 2** (%)			+7.22%	+4.97%	+14.55%	+12.74%

Specifically, in Table 7, the Combo. 1 means that above two mechanisms are not adopted. In such the situation, TEP2MP only processes the sequence of *affective vectors* corresponding to the target speaker and no consideration on other participants is involved. Combo. 2 means that only the emotion fusion mechanism is adopted. In other words, only the first-layer encoder aimed at multi-participants exists in TEP2MP and the emotion fusion result output by the first-layer encoder is considered as the final prediction result. Combo. 3 means that only the hybrid attention mechanism is adopted. In other words, TEP2MP only considers target speaker at the encoder side. In addition, at the decoder side, the two context vectors (i.e., o_t and c_t) are still merged by the gate switch g. Such two context vectors are derived from the attention-feature sequence A and the hidden state sequence of the first-layer encoder respectively. Combo. 4 is the scheme illustrated in Figure 6, where the emotion fusion and hybrid

attention mechanism are all adopted.

In Table 7, it can be observed that Combo. 1 has achieved the worst performance where both of the emotion fusion and hybrid attention mechanism are not adopted. However, for Combo. 2, when adopting the emotion fusion mechanism, the emotion prediction performance has increased which means that the emotion interaction learned among multiple participants at the encoder side is beneficial to enhancing the emotion prediction performance. In addition, when comparing Combo. 1 & 2 with Combo. 3 & 4, it can be found that combinations with hybrid attention is better than those without hybrid attention mechanism. Specifically, when on other participants are considered, like Combo. 3, the emotion prediction performance is worse than the one adapted to the multi-participant scenario, like Combo. 4. Finally, the Combo. 4 is the best one among all the combinations.

5. Conclusions & discussion

Text can reflect the opinion or attitude of the publisher, therefore analyzing the emotion of the conversation text is beneficial to the decision-making and public opinion supervision. In this paper, a text emotion prediction model with emotion fusion and hybrid attention enhancement has been proposed in a multi-participant conversation scenario. The proposed model computes affective vectors of each conversation text via Affective Space Mapping. Then, scenes with similar emotion tendency have been searched across the entire conversation history, which are further converted into attention-feature sequence. Next, a two-layer encoder-decoder prediction model has been constructed where the emotion fusion mechanism is introduced at the encoder side to capture emotion interaction among multiple participants and the hybrid attention mechanism is introduced at the decoder side to compute the context vector used for decoding. According to the experimental results, our proposed model can achieve the overall best performance on text emotion prediction due to the auxiliary features extracted from similar scenes and the adoption of emotion fusion as well as the hybrid attention mechanism. Moreover, the text emotion prediction efficiency of our proposed model can still be controlled at an acceptable level.

However, it can also be observed from the experimental results that compared with other existing methods, the emotion prediction performance of our proposed model is merely improved within a limited scope. Such the limitation can be ascribed to that, since the affective vectors corresponding to the conversation texts posted by all the other participants have been input to the encoder, therefore emotions of those irrelevant participants have also been considered when predicting emotion for the target speaker. In others words, there may exist an optimal combination of other participants with regard to the target speaker, which has not been considered by our work. Second, it is evident that different participants may have strong or weak associations with the target speaker, however, when it comes to the emotion fusion mechanism adopted by our model, we have not assigned a proper "weight" to the corresponding affective vector of each conversation text. Therefore, more effort should be devoted to process the emotion of conversation text at much finer granularity.

Acknowledgments

This work is sponsored by the National Natural Science Foundation of China (NO. 62102187).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. D. Bertero, F. B. Siddique, C. S. Wu, Y. Wan, R. H. Y. Chan, P. Fung, Real-time speech emotion and sentiment recognition for interactive dialogue systems, in *Proceedings of the 2016 conference on empirical methods in natural language processing*, (2016), 1042–1047. https://doi.org/10.18653/v1/D16-1110
2. Y. Zhang, P. Tiwari, D. Song, X. Mao, P. Wang, X. Li, et al., Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis, *Neural Networks*, **133** (2021), 40–56. https://doi.org/10.1016/j.neunet.2020.10.001
3. F. Hemmatian, M. K. Sohrabi, A survey on classification techniques for opinion mining and sentiment analysis, *Artif. Intell. Rev.*, **52** (2019), 1495–1545. https://doi.org/10.1007/s10462-017-9599-6
4. A. Yadav, D. K. Vishwakarma, Sentiment analysis using deep learning architectures: a review, *Artif. Intell. Rev.*, **53** (2020), 4335–4385. https://doi.org/10.1007/s10462-019-09794-5
5. Y. Song, S. Shi, J. Li, H. Zhang, Directional skip-gram: Explicitly distinguishing left and right context for word embeddings, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies*, **2** (2018), 175–180. https://doi.org/10.18653/v1/N18-2028
6. C. Y. Liou, W. C. Cheng, J. W. Liou, D. R. Liou, Autoencoder for words, *Neurocomputing*, **139** (2014), 84–96. https://doi.org/10.1016/j.neucom.2013.09.055
7. J. Deng, F. Ren, A survey of textual emotion recognition and its challenges, *IEEE Trans. Affective Comput.*, 2021. https://doi.org/10.1109/TAFFC.2021.3053275
8. I. Chaturvedi, E. Cambria, R. E. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: Survey and challenges, *Inf. Fusion*, **44 (**2018), 65–77. https://doi.org/10.1016/j.inffus.2017.12.006
9. E. Cambria, Y. Li, F. Z. Xing, S. Poria, K. Kwok, SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis, in *Proceedings of the 29th ACM international conference on information & knowledge management*, (2020), 105–114. https://doi.org/10.1145/3340531.3412003
10. M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, U. R. Acharya, ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis, *Future Gener. Comput. Syst.*, **115** (2021), 279–294. https://doi.org/10.1016/j.future.2020.08.005
11. C. Gong, J. Yu, R. Xia, Unified feature and instance based domain adaptation for end-to-end aspect-based sentiment analysis, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2020), 7035–7045. https://doi.org/10.18653/v1/2020.emnlp-main.572
12. H. Peng, L. Xu, L. Bing, F. Huang, W. Lu, L. Si, Knowing what, how and why: A near complete solution for aspect-based sentiment analysis, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 8600–8607. https://doi.org/10.1609/aaai.v34i05.6383

13. C. Yang, H. Zhang, B. Jiang, K. Li, Aspect-based sentiment analysis with alternating coattention networks, *Inf. Process. Manage.*, **56** (2019), 463–478. https://doi.org/10.1016/j.ipm.2018.12.004
14. H. Cai, Y. Tu, X. Zhou, J. Yu, R. Xia, Aspect-category based sentiment analysis with hierarchical graph convolutional network, in *Proceedings of the 28th International Conference on Computational Linguistics*, (2020), 833–843. https://doi.org/10.18653/v1/2020.coling-main.72
15. M. H. Phan, P. O. Ogunbona, Modelling context and syntactical features for aspect-based sentiment analysis, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (2020), 3211–3220. https://doi.org/10.18653/v1/2020.acl-main.293
16. Y. Ma, K. L. Nguyen, F. Z. Xing, E. Cambria, A survey on empathetic dialogue systems, *Inf. Fusion*, **64** (2020), 50–70. https://doi.org/10.1016/j.inffus.2020.06.011
17. S. Poria, D. Hazarika, N. Majumder, R. Mihalcea, Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research, *IEEE Trans. Affective Comput.*, 2020. https://doi.org/10.1109/TAFFC.2020.3038167
18. W. Li, W. Shao, S. Ji, E. Cambria, BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis, *Neurocomputing*, **467** (2022), 73–82. https://doi.org/10.1016/j.neucom.2021.09.057
19. Z. Lian, B. Liu, J. Tao, CTNet: Conversational transformer network for emotion recognition, *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, **29** (2021), 985–1000. https://doi.org/10.1109/TASLP.2021.3049898
20. Y. Zhang, P. Tiwari, D. Song, X. Mao, P. Wang, X. Li, et al., Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis, *Neural Networks*, **133** (2021), 40–56. https://doi.org/10.1016/j.neunet.2020.10.001
21. J. Wang, J. Wang, C. Sun, S. Li, X. Liu, L. Si et al., Sentiment classification in customer service dialogue with topic-aware multi-task learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 9177–9184. https://doi.org/10.1609/aaai.v34i05.6454
22. W. J. Huang, Y. T. Li, Y. Huang, Prediction of chaotic time series using hybrid neural network and attention mechanism, *Acta Physica Sinica*, **70** (2021), 010501. https://doi.org/10.7498/aps.70.20200899
23. Y. Liu, C. Gong, L. Yang, Y. Chen, DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction, *Expert Syst. Appl.*, **143** (2020), 113082. https://doi.org/10.1016/j.eswa.2019.113082
24. D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks, *Int. J. Forecast.*, **36** (2020), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001
25. J. Koutnik, K. Greff, F. Gomez, J. Schmidhuber, A clockwork rnn, in *International Conference on Machine Learning*, PMLR, **32** (2014), 1863–1871.
26. X. Liu, J. Zheng, Research on time series forecasting based on integrating clockwork recurrent neural network, *Comput. Digital Eng.*, **48** (2020), 1590–1594.
27. O. Sagi, L. Rokach, Ensemble learning: A survey, *Wiley Interdiscip. Rev., Data Mining Knowl. Discovery*, **8** (2018), e1249. https://doi.org/10.1002/widm.1249
28. Z. Liu, J. Liu, A robust time series prediction method based on empirical mode decomposition and high-order fuzzy cognitive maps, *Knowl.-Based Syst.*, **203** (2020), 106105. https://doi.org/10.1016/j.knosys.2020.106105

29. P. Liu, J. Liu, K. Wu, CNN-FCM: System modeling promotes stability of deep learning in time series prediction, *Knowl.-Based Syst.*, **203** (2020), 106081. https://doi.org/10.1016/j.knosys.2020.106081
30. H. Zhou, M. Huang, T. Zhang, X. Zhu, B. Liu, Emotional chatting machine: Emotional conversation generation with internal and external memory, in *Thirty-Second AAAI Conference on Artificial Intelligence*, **32** (2018), 730–738.
31. Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Aït-Bachir, Period-aware content attention RNNs for time series forecasting with missing values, *Neurocomputing*, **312** (2018), 177–186. https://doi.org/10.1016/j.neucom.2018.05.090
32. J. Chen, K. Li, H. Rong, K. Bilal, K. Li, S. Y. Philip, A periodicity-based parallel time series prediction algorithm in cloud computing environments, *Inf. Sci.*, **496** (2019), 506–537. https://doi.org/10.1016/j.ins.2018.06.045
33. L. K. Hansen, P. Salamon, Neural network ensembles, *IEEE Trans. Pattern Anal. Machine Intell.*, **12** (1990), 993–1001. https://doi.org/10.1109/34.58871
34. T. Chai, R. R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE), *Geosci. Model Dev. Discuss.*, **7** (2014), 1525–1534. https://doi.org/10.5194/gmdd-7-1525-2014
35. Q. Song, Y. J. Zheng, Y. Xue, W. G. Sheng, M. R. Zhao, An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination, *Neurocomputing*, **226** (2017), 16–22. https://doi.org/10.1016/j.neucom.2016.11.018
36. F. Krebs, B. Lubascher, T. Moers, P. Schaap, G. Spanakis, Social emotion mining techniques for Facebook posts reaction prediction, in *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART)*, **2** (2018), 211–220. https://doi.org/10.5220/0006656002110220
37. Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, (2017), 2627–2633. https://doi.org/10.24963/ijcai.2017/366
38. H. Goel, I. Melnyk, A. Banerjee, R2N2, residual recurrent neural networks for multivariate time series forecasting, preprint, arXiv:1709.03159.
39. A. Borovykh, S. Bohte, C. W. Oosterlee, Conditional time series forecasting with convolutional neural networks, preprint, arXiv,1703.04691.
40. G. E. Hinton, R. S. Zemel, Autoencoders, minimum description length, and Helmholtz free energy, *Adv. Neural Inf. Process. Syst.*, **6** (1994), 3–10.

AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)