Neural Computing and Applications (2023) 35:13565–13582
https://doi.org/10.1007/s00521-023-08435-x

ORIGINAL ARTICLE

EmoDNN: understanding emotions from short texts through a deep neural network ensemble

Sara Kamran[1] · Raziyeh Zall[1] · Saeid Hosseini[2] · MohammadReza Kangavari[1] · Sana Rahmani[1] · Wen Hua[3]

Received: 20 September 2022 / Accepted: 20 February 2023 / Published online: 15 March 2023

Abstract

The knowledge obtained from emotions via online communities is substantially valuable in various domains, including social management, resource planning, politics, and market predictions. Affective computing, as a multi-aspect realm, aims to exploit emotion-pertinent details from various contents via connecting artificial intelligence to cognitive science. The hidden personality cues in daily brief contents can reveal the cognitive aspect of authors and uncover both similarities and contrasts between them. However, the main challenge lies in devising a cognition-aware algorithm to trace emotional cues in brief contents. To solve the challenge, we develop a novel framework that infers the cognitive aspect of individuals. We propose a deep ensemble method, supplied with a novel dropout algorithm, that aggregates outcomes from various classifiers to extract emotions from short texts. We employ a new embedding approach to enrich emotion-relevant features, collectively assembled via lexicons and attention actuates, resulting in a preferable set of vectors. The experimental results show that our proposed framework can achieve better accuracy in recognizing emotions versus other trending competitors. We empirically observe that detecting emotion latent cues via relying on personality features can effectively distinguish short text authors. Furthermore, the deep learning models overcome conventional methods, including the SVM, categorization, and heuristic rules.

Keywords Neural network architecture · Cognitive factors · Emotion recognition · Ensemble learning

✉ Saeid Hosseini
sahosseini@su.edu.om

Sara Kamran
sara.kamran72@gmail.com

Raziyeh Zall
zall_razieh@comp.iust.ac.ir

MohammadReza Kangavari
kanagvari@iust.ac.ir

Sana Rahmani
rahmany.sana@gmail.com

Wen Hua
w.hua@uq.edu.au

1 School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

2 Faculty of Computing and Information Technology, Sohar University, Sohar, Oman

3 School of IT and Electrical Engineering, University of Queensland, Brisbane, Australia

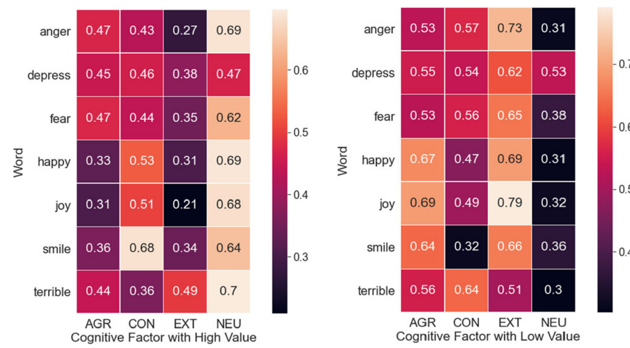
1 Introduction

Understanding emotional information from short text content has prominent applications in various fields: (i) In conversation transcripts in user contents [1]. (ii) In the political context, to foster the prediction of the ballet results [2]. (iii) In health, to better recognize the people affected by extreme depression [3, 4] (iv) In sales and finance, to enhance the product development [5] and predict the market fluctuations [6]. Nowadays, with the spread of social networks, people share their brief contents conveying latent cues such as personality that can identify the contrast between authors [7]. Given a single short text s_i with corresponding cognitive cues p_j, we aim to identify the emotions explaining s_i. Unlike formal documents, individuals are more to reveal their emotions within brief contents, like social media posts. Accordingly, one can use such valuable data to extract emotion-pertinent groups. However, challenges abound:

Challenge 1 (Ignoring Author Latent Information)

Many previous efforts [8, 9] exploit emotional cues from textual contents without considering the contrast between authors. However, personality differences can distinguish how people express their feelings [10]. Hence, Holtgraves et al. [11] claim that the personality relevance between people is reflected via emotion-related content. Table 1 demonstrates the correlation between a tweet and relevant emotion and cognitive vectors. Here emotion vector comprises the score for each emotion [anger, disgust, etc.], where we use a similar vector to express personality through Extraversion (EXT), Openness (OPE), etc. The respective binary and floating values in the emotion and personality vectors imply whether the short text corresponds to each emotion or cognitive factor.

We use the SemEval dataset [12] to observe the distribution probabilities of the words that help us investigate the impact of personality on emotions within textual contents. We select the expressions with various emotional intensities that are further pertinent to diverse cognitive factors. Figure 1 depicts the distribution probabilities for each emotional word differ in various cognitive factors. For example, to manifest the relationship between NEU cognitive factor and emotional words, we first retrieve the short text with the high value in neuroticism and then calculate the percentage of the selected short texts that contain pertinent words reflecting the anger. Users with high Neuroticism (NEU) have low emotional stability and frequently use distressing words, like terrible. Hence, their short text contents include more emotional words compared to the user of the low NEU weights. Interestingly, users with low EXT are reluctant to interact with others and tend to express negative emotions such as fear, depression, and anger more frequently than other users with high EXT loads. Individuals with a high rate of Conscientiousness (CON) use a greater number of positive emotional words, such as happy, joy, and smile, in their contents. From another perspective, people with high Agreeableness (AGR) weights rarely utilize emotional-related words. Therefore, the proposed consensus results reveal how cognitive cues can influence emotional expressions in short text contents. Since individuals with identical cognitive cues exhibit similar emotions, we leverage the latent personality aspects to enhance our ability to detect emotions.



(a) High Cognitive Factors (b) Low Cognitive Factors

Fig. 1 Emotion words connect to cognitive factors

Challenge 2 (Noisy Short Text)

Short texts include important information but they are brief and error-prone. Hence, it is a tedious task to associate emotion cues with cognitive features.

Challenge 3 (The Scarceness of Annotated Dataset)

All prevailing datasets have either emotional labels [12, 13] or cognitive labels [14]. Hence, due to the scarceness of the dataset which contains both emotion and cognitive annotations, one of our challenges is to create datasets that are annotated by emotion and cognitive cues.

Contributions. While our prior works [15, 16] handle short text contents to detect attributed segments [17] and identify concepts, removing the perturbation caused by external knowledge bases, this paper recognizes the emotions through leveraging the cognitive factors from similar brief context. To this end, we utilize cognitive factors of individuals to categorize short texts to feed our novel ensemble classifier. Furthermore, in our proposed framework, we detect the emotion embedding of brief contents via an external database that is associated with emotion lexicons. Accordingly, we extract multichannel features in short texts to enrich the short text inference models. The contributions are as follows:

Table 1 Conceptual relevance

	Emotion					Cognitive factors				
Tweet	Anger	Disgust	Fear	Joy	Sadness	OPE.	CON.	EXT.	AGR.	NEU.
I've never been so excited to start a semester!				✓		4.1	3.5	3.38	3.623	2.559
I'm a shy person					✓	4.11	3.498	3.392	3.622	2.561
I am just so bitter today	✓	✓			✓	4.0	3.508	3.366	3.564	2.58

- We develop a framework for emotion recognition through ensemble learning which leverages the cognitive cues to distinguish between short text authors.
- We propose a regression approach for inferring latent cues about short text authors.
- We design a multichannel feature extraction algorithm based on emotion lexicons and attention mechanisms that fuse various embedding models to retrieve preferable vectors.
- We devise a cognitive aware aggregation function to collectively learn via various classifiers using an ensemble module.

The rest of the paper is as follows: Sect. 2 explains the related work; Sect. 3 elucidates both the problem and framework; Sect. 4 explains the model and Sect. 5 discusses experiments, concluded in Sect. 6.

2 Related work

The related work is in two parts: the personality estimation and emotion recognition (Table 2).

2.1 Personality prediction

Personality is a psychological construct, and it aims to understand various human behaviors with stable and measurable individual characteristics [21]. Various studies predict and identify personality traits [14] from social networks. The methods for cognitive prediction are: 1) Machine learning methods; 2) Deep learning models. Traditional machine learning algorithms [18] leverage various features [44] to detect the personality traits of individuals in social networks. In this context, [19] extracts linguistic statistics, including word count, to enrich the feature set and train a Bayesian method for personality detection. Quercia et al. [20] count on regression analysis by utilizing social network attributes such as the number of friends that can estimate personality cues. Sun et al. [22] combine bidirectional LSTMs (Long Short-Term Memory networks) with CNNs (Convolutional Neural Networks) to infer the structures of texts. Similarly, [21] devised the AttRCNN structure to understand the semantic features that are amended by the statistical linguistic features. However, [23] utilizes essential statistics, including Mairesse features, the number of words, and the average length of sentences that are combined within convnets in a hybrid manner. We initially exploit latent personal characteristics by a modified Support Vector Regression (SVR) model and subsequently adopt CNN convents to extract feeling sentiments.

Table 2 Literature

Category	Approaches	References
Personality prediction	Machine learning	[18–20]
	Deep learning	[21–23]
Emotion recognition	Lexicon-Based	[24–26]
	Machine learning	[8, 27–33]
	Deep learning	[1, 9, 34–43]

2.2 Emotion recognition

Affective Computing [45] as an emerged field of research has attracted wide attention since it results in systems that can automatically recognize human emotions and influence decision-making procedures [46]. Emotions can be detected by heuristic, machine learning, and deep learning methods [43]. The lexicon-based heuristic approaches [24, 25] find keywords in textual contents and assign emotion labels based on lexicon tags, referenced from knowledge-base tools such as NRC-EIL [24] and DepecheMood [25]. Nevertheless, context-based methods can detect emotions. For instance, Tao et al. [26] devise the ESiN framework that combines the content and emotion-operative words to estimate the final pertinent emotion cues. Such methods merely leverage the emotional terms provided in their affective lexicon and neglect the context of the given words in the short texts. However, observations in various contexts can create diverse emotional meanings. Hence, we incorporate the strength of emotion lexicons and text embedding modules to attain efficient feature vectors for the proposed learning model. Machine learning (ML) models [27, 32] not only consider lexicons but also extract effective textual features from input corpus, concluded by decision rules to recognize emotions based on the trained explicit labels. To enhance the learning process of the Bayesian methods, [27] attains the importance of emotional words in preprocessing step. Despite the efficient utilization of various word classes in the prediction module, the accuracy does not turn inflated. Random forest (RF) [33] combines the multiple tree estimators, each depending on a random vector and sampled with the same distribution of the forest. RF approaches are more efficient than the support vector machines (SVM) methods [27, 29] but empirically gain less accuracy. Both Esmin [8] and Xu [28] et al. use hierarchical classification techniques to perceive emotion cues. Such techniques integrate three levels: neutrality versus emotionality, sentiment analysis, and emotion recognition. Utilizing domain-specific emotion lexicons via n-grams and part of speech (pos) tagging can foster the classification performance of the SVM modules [30]. Traditional machine learning methods rely on a particular set of engineering features, resulting in losing beneficial information for emotion detection during the feature extraction procedure. To tackle the challenge, we devise a novel feature engineering model that leverages semantic and emotional cues, structural elements, and syntactic patterns to broaden the feature set in the learning step. In addition, we use deep neural networks that do not require manual feature adjustments. On the contrary, the intelligible rule-based methods share the goal of finding regularities in data, expressing the form of If-Else rules [31]. Here Orizu et al. [31] recognize the emotions if the brief contents adapt to the given rule set. In contrast, we rely on the learning process and collectively utilize the embedding and automatic multichannel feature extraction. DNN models [1, 34] have promoted the performance of prior Natural Language Processing (NLP) techniques, including emotion analysis and recognition. CNN [9, 35] and RNN (Recurrent NN)[9, 36] are two common deep learning architectures that are often integrated at the top of embedding modules, e.g., GloVe and Word2Vec, to infer emotion-pertinent cues in textual contents. While the convnets can effectively extract n-gram features, they are not as productive as the RNN schemes in attaining correlation within long-term sequences. Nonetheless, CNN models are distorted toward subsequent context and neglect previous words. To address the issue, LSTM models [37] exploit the intensity of emotions out of brief contents in a bidirectional manner which results in preferable outputs in single and multi-label classification tasks. The Deep Rolling model [38] combines LSTM and CNN into an ensemble to create a nonlinear emotion-prediction model. SpanEmo [40] leverages multi-label emotion recognition as a span prediction problem to extract the label dependencies. The proposed method utilizes the Bidirectional Encoder Representations from Transformers, the so-called BERT [41]. Furthermore, it presents a loss function that infers multiple coexisting emotions in the input sentence to maximize the distance between the positive and negative labels. Similarly, [42] detects emotions in text contents via utilizing the LSTM and transfer learning models. However, it adopts the learning phase's multi-attention mechanism to preserve emotional information better. Nevertheless, the approaches above [45][60-61] neglect the implicit latent cues of individuals in the emotion detection procedure. To tackle this issue, our proposed method leverages an ensemble of learners to infer various categories of short texts according to the latent cognitive cues. Kumar et al. [39] devise a dual-channel system to detect emotions in multiple classes from textual contents. They adopt the BERT [41] to extract the textual features and feed them into CNN and LSTM modules. Unlike our proposed method, which exploits the dropout rate via the weights drawn from the data-specific uniform distribution, [39] is confined to the traditional dropout. Inspired by the appealing performance of the DNN models such as [43], we devise an ensemble classifier with novel dynamic dropout convnets, consuming individual latent aspects that we name cognitive cues. In addition, we propose a nontrivial method to extract features from emotions and semantic contents to feed them into the convnets in ensembles.

3 Problem statement

In this section, we explain primary concepts, problems, and the unified framework. We elucidate the notations in Table 3.

3.1 Primary concepts

Definition 1 (short text message) $s_i \in S$ is a short text written by an author. Accordingly, S is our corpus which includes all short texts.

Definition 2 (cognitive factors) $p_i \in P$ ($P = \{p_i \mid i \in [1, ..., q]\}$) refers to a cognitive factor (e.g., neuroticism). Each factor reveals one latent perception for the given message.

Definition 3 (emotion) $e_i \in E$ ($E = \{e_i \mid i \in [1, ..., k]\}$) refers to a basic emotion such as happiness. Each emotion can be correlated with multiple cognitive factors.

Definition 4 (cognitive category) Each cognitive category $c_i \in C$ can represent a set of short texts that are highly correlated based on pertinent cognitive factors.

Definition 5 (ensemble) $h_{c_i} \in H$ refers to a base classifier that is induced from cognitive category $c_i (c_i \in C)$. Accordingly, H ($H = \{h_{c_1}, h_{c_2}, ..., h_{c_n}\}$) includes all base classifiers.

Definition 6 (emotion Vector) $\vec{V}_{s_i} = [v_{i,j} \mid j \in [1, ..., k]]$ denotes the emotion scores for short text $s_i \in S$ where $v_{i,j} \in \mathbb{R}$ is an emotion score for j^{th} basic emotion $e_j \in E$.

Definition 7 (cognitive Vector) $\vec{F}_{s_i} = [f_{i,j} \mid j \in [1, ..., q]]$ denotes the cognitive factor scores for the short text $s_i \in S$ where $f_{i,j}$ is a cognitive score for j^{th} factor $p_j \in P$. $\vec{F}_{s_i}$ represents short text s_i in cognitive space with q dimension according to P.

Table 3 Table of important symbols

Notation	Definition
s_i	A short text of an author
p_i	A cognitive factor
e_i	A basic emotion
c_i	A category obtained via cognitive inference.
D^P	Cognitive annotation dataset
D^E	Emotional annotation dataset
α_j	The partitioning threshold vector
$\vec{O}_j^{\text{context}}$	Context-wise vector
$\vec{O}_j^{\text{emotion}}$	Emotion-wise word vector
$\vec{O}_j^{pos}$	POS-tagged vector
$\vec{O}_j^{\text{attention}}$	Attention vector
$\hat{Q}$	Input of emotion learning component
$\hat{v}_{j,i}$	The network prediction output for the j^{th} emotion of the i^{th} short text
A	Consensus matrix for ensemble module
$\vec{V}_{s_q}$	Final emotion vector for short text s_q in the online section
$\vec{F}_{s_i}$	Cognitive Vector for short text s_i

3.2 Problem definition

Problem 1 (*identifying cognitive factors*) Given a message s_i, our goal is to retrieve a cognitive vector denoted by $\vec{F}_{s_i}$ to categorize users into a set of cognitive groups $C = (c_1, .., c_n)$, attaining the similarity between users based on the correlation between pertinent cognitive cues in each category.

Problem 2 (*extracting multichannel features*) Given the set of short texts relevant to a cognitive category c_i, we aim to extract features from emotion-pertinent cues, engineering features to detect latent emotions within short text contents.

Problem 3 (*recognizing emotions by cognitive factors*) Given the message s_i and the associated cognitive vector $\vec{F}_{s_i}$, our goal is to detect the emotions of s_i, represented by vector $\vec{V}_{s_i}$.

3.3 Framework overview

The problem of emotion recognition through cognitive cues in brief contents, illustrated in Fig. 2, includes two steps: (1) extracting the categories of short text messages via inclusive cognitive cues. (2) Learning an effective ensemble classifier to identify emotions by leveraging the extracted categories. In the *offline* part, since a dataset enclosed with both emotion and cognitive annotations is scarce, as a prerequisite to emotion recognition, we initially augment the cognitive annotations within the dataset. To this end, we adopt the SVR to retrieve the cognitive vectors. Considering the impact of each cognitive factor on emotion-related expressions, we then apply categorization to textual contents. As a result, each category can include highly correlated short texts aligned with the associated cognitive vector. Consequently, we extract the set of features by tracking the emotional cues in the short texts of each cognitive category. We then apply the emotion lexicons together with word vectors to learn each of the corresponding base classifiers (e.g., extraversion). To continue, given the short text contents enclosed with emotion annotations, we induce the ensemble classifier to convey emotion recognition. We can aggregate the base classifiers into an ensemble to convey helpful information according to the cognitive similarities. In the *online* part, we aim to predict emotion labels for the input short text. To

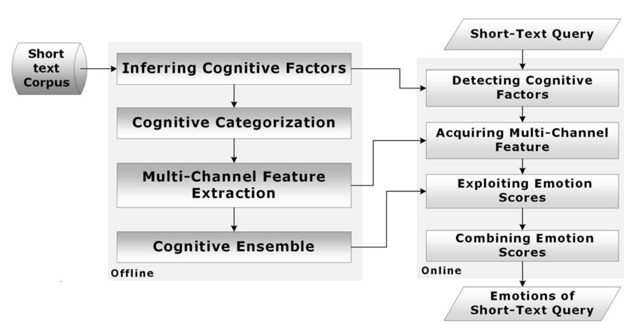


Fig. 2 Framework

accomplish the task, we firstly extract cognitive vectors from the input. We then select a set of relevant classifiers based on the input cognitive features. Finally, we aggregate various outputs from classifiers to make the final prediction.

4 Methodology

4.1 Offline phase

4.1.1 Inferring cognitive factors

To investigate the influence of cognitive factors on emotion recognition, we need a dataset that includes both emotional and cognitive annotations. We assume that our dataset includes emotion annotations as ground truth. Hence, we aim to compensate for the cognitive annotations. To this end, we can leverage another source dataset with cognitive features to annotate our target dataset. We use a cognitive annotation dataset $D^P = \{s_i, \vec{F}_{s_i}\}(i = 1, ..., n)$ to infer cognitive vectors of short texts in emotional annotation dataset $D^E = \{s_i, \vec{V}_{s_i}\}(i = 1, ..., m)$, where P and E denote the source and target datasets. Here m and n denote the number of short texts in source and target datasets. To identify cognitive scores, we train a diverse model for each cognitive factor q on D^P to infer the proportionate cognitive space of D^E. In each model, we adopt the effective Support Vector estimation tool [47] to designate a decision surface to maximize the distance between different classes. In the source dataset, there are a set of points $(s_i, f_{s_{i,j}})$ where x_i is the feature vector extracted from s_i and $f_{s_{i,j}} \in \vec{F}_{s_i}$ is the target value for each model $j \in [1, ..., q]$. We intend to find a regression model denoted by $f(x_i) = w^T \varphi(x_i) + b$, where w is the slope of the line and b is the intercept. Our aim is to use SVR to find a surface that minimizes the prediction error in optimization function, Eq. 1. In regression, a soft-margin (ϵ) approach is employed similar to SVM. We add slack variables $\xi_i + \xi_i^*$ to guard against outliers.

$$w, b = \mathbf{M}in \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \tag{1}$$

Here ξ and ξ^* are the distance of data points in out of the ϵ margin.

$$\mathbf{M} = \begin{cases} f_{s_i,j} - w^T x_i - b + \xi_\mathrm{i} \\ w^T x_i + b - f_{s_i,j} + \xi_\mathrm{i}^* \\ \xi_i, \xi_i^* 0 \end{cases} \tag{2}$$

We can achieve the conditions in Eq. 2 via a Lagrange multiplier method in Eq. 3. Here, α_j and α_j^* are nonnegative real numbers to signify Lagrange multipliers.

$$\mathbf{f}(x_i) = \sum_{j=1}^{n} (\alpha_j^* - \alpha_j) \kappa(x_i, x_j) + b \tag{3}$$

We utilize $\kappa(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ as the kernel function to achieve the inner product of x_i and x_j in the higher-dimensional feature space.

4.1.2 Cognitive categorization

Given emotion annotated dataset D^E, we aim to acquire a set of cognitive categories, denoted by C. We designate each cognitive factor $p_i \in P$ with continuous values ranked on a scale between the two extreme ends for each short text to obtain cognitive categories. For instance, an individual's score for extraversion factor determines the pertinent weight. Given the specific levels for each cognitive factor, used to measure the personality contrasts, every individual can hold a variety of facets. Inherently, we can appoint each facet within a range; for example, *extraversion* varies from friendly to reserved, *agreeableness* from authentic to self-interested, *conscientiousness* from organized to sloopy, *neuroticism* from comfortable to uneasy, and *openness* from imaginative to practicals. As elucidated in Sect. 1, authors with different cognitive cues express their emotions using various vocabs in emotion. For instance, authors of the short text content with high Neuroticism weight are less likely to be emotional and vice versa. Hence, we segregate individuals into two categories of low and high scores [48]. Given such an intuition and based on the cognitive vector $\vec{F}_{s_i}$ of the set P, we can map each short text $s_i \in D^E$ to q dimensions. Subsequently, we can obtain the set C of cognitive categories from the emotion annotation dataset D^E by splitting the space into two subspaces. Where we can define the lower and upper subspaces for each cognitive factor $p_j \in P$. Accordingly, the short texts with lower and higher levels of a cognitive factor, p_j, will be, respectively, appended to the pertinent categories of c_{2j-1}, c_{2j}. Hence, clustering [49] and border classifiers, like SVM [27], cannot cohesively distinguish the categories. To this end, we diversely adopt an entropy-based categorizing method to acquire partitioning value in each cognitive factor to obtain lower and higher bounds. To get the partitioning threshold vector $\vec{a} = \{\alpha_j \mid j \in [1, ..., q]\}$, we presumed that the source dataset incorporates the cognitive factor classes. Hence, we employed the entropy approach to attain the partition value of α_j for each cognitive factor p_j that minimized the impurity in the resultant categories.

For each given cognitive factor p_j, we considered a set of partitioning points in a similar range as p_j in D^P and evaluated them based on entropy to find the best partitioning point, α_j. In this regard, based on each partitioning point T, we split the D^P into two subsets of d_1 and d_2 and computed the entropy of resulting subsets accordingly to input cognitive class of p_j. We determined two classes $k = \{k_1, k_2\}$ for each cognitive factor $p_j \in P$ where the entropy of $d_i(i = \{1, 2\})$ was defined as Eq. 4.

$$Entropy(d_i) = \sum_{m \in \{1,2\}} P(k_m, d_i) \log P(k_m, d_i) \tag{4}$$

Here, $P(k_m, d_i)$ is the probability of short texts in d_i pertaining class k_m. Given input dataset D^P, cognitive factor p_j, and partitioning point T, Eq. 5 computes the class information entropy $E(p_j, T, D^P)$ for the splits made by T. Here, α_j is the partitioning criterion in Eq. 6, which minimizes the impurity in resultant categories.

$$E(p_j, T, D^P) = \sum_{m \in \{1,2\}} \frac{| d_m |}{| D^P |} Entropy(d_m) \tag{5}$$

$$\alpha_j = \underset{T}{argmin}\{E(p_j, T, D^P)\} \tag{6}$$

We then used α_j in Eq. 7 to obtain cognitive categories including c_{2j} and c_{2j-1}.

$$\forall p_j \in P : \begin{cases} c_{2j-1} = \{[s_i, \vec{V}_{s_i}] \mid f_{i,j} < \alpha_j\} \\ c_{2j} = \{[s_i, \vec{V}_{s_i}] \mid f_{i,j} \geq \alpha_j\} \end{cases} \tag{7}$$

Hence, the pair of sets (c_{2j-1},c_{2j}) formed by α_j on each parameter p_j can constitute the short texts with a low and high order of cognitive cues. We now elucidate various properties for each set of cognitive categories ($C = (c_1, ..., c_n)$):

Lemma 1 *a cognitive category c_i can not be empty* ($\forall i \in [1, ..., 2q] : c_i \neq \varnothing$).

Proof Since the max. and min. values for each cognitive factor i do not equate and the splitter parameter α_i is between min. and max. values so we can justify that c_i can not be empty (Eq. 8):

$$\begin{cases} if\, \alpha_i \geqslant min(D^E_i) \Rightarrow \exists s_j : f_{j,i} \leqslant \alpha_i \\ if\, \alpha_i < max(D^E_i) \Rightarrow \exists s_j : f_{j,i} > \alpha_i \end{cases} \Rightarrow c_i \neq \varnothing \tag{8}$$

Lemma 2 *The aggregated data for all cognitive categories form the original dataset D^E* ($\forall_i \in [1, ..., 2q] : \bigcup c_i = D^E$).

Proof Suppose we have two categories of $c_{2j} \subseteq D^E$ and $c_{2j-1} \subseteq D^E$ where we assign the short text $s_k \in D^E$ to either c_{2j} or c_{2j-1}. Hence, we can justify the rules in Eq. 9 for the cognitive factor j in s_k:

$$\begin{aligned} &\forall s_k \in D^E : s_k \in c_{2j} \text{ or } s_k \in c_{2j-1} \Rightarrow \\ &c_{2j} \cup c_{2j-1} = D^E \Rightarrow \bigcup_{i \in [1,...,2q]} c_i = D^E \end{aligned} \tag{9}$$

Lemma 3 *The intersection of two cognitive categories for the same parameter (e.g., c_{2j} and c_{2j-1}) partitioned by the specified threshold α_j will result in null. As stated in Eq. 10, the intersection of the same pair (c_{2j},c_{2j-1}) with other cognitive categories c_i can be non-empty.*

$$\begin{cases} \forall j \in [1, ..., q], \\ \forall i \in [1, ..., 2q] \end{cases} \Rightarrow \begin{cases} c_{2j-1} \cap c_i \neq \varnothing & \text{if } 2j-1 \neq i \\ c_{2j} \cap c_i \neq \varnothing & \text{if } i \neq 2j \\ c_{2j-1} \cap c_{2j} = \varnothing \end{cases} \tag{10}$$

Proof Given cognitive factor j, each short text can be associated either with low c_{2j-1} or high c_{2j} status, resulting in $c_{2j-1} \cap c_{2j} = \varnothing$. However, given all q cognitive factors for each short text with q cognitive categories, every pair of c_{2j-1} and c_i can share common textual contents.

Lemma 4 *Short texts in a cognitive category c_m have a similar feature according to their cognitive factor $\lfloor \frac{m+1}{2} \rfloor$. The $p_{\lfloor \frac{m+1}{2} \rfloor}$ in all of contents is larger than $\alpha_{\lfloor \frac{m+1}{2} \rfloor}$ or is smaller than $\alpha_{\lfloor \frac{m+1}{2} \rfloor}$.*

Proof Suppose $\forall m \in [1, ..., 2q]$ and $\forall i, j \in [1, ..., | c_m |]$, we can use contradiction logic to prove $s_i, s_j \in c_m$ stated by $f_{i,\lfloor \frac{m+1}{2} \rfloor}, f_{j,\lfloor \frac{m+1}{2} \rfloor} > \alpha_m$ or $f_{i,\lfloor \frac{m+1}{2} \rfloor}, f_{j,\lfloor \frac{m+1}{2} \rfloor} \leq \alpha_m$. Let $s_i \in c_m$ and $s_j \notin c_m$ where $f_{i,\lfloor \frac{m+1}{2} \rfloor}, f_{j,\lfloor \frac{m+1}{2} \rfloor} > \alpha_m$. According to Eq. 7:

$$\begin{cases} 1.\ s_i \in c_m \Rightarrow f_{i,\lfloor \frac{m+1}{2} \rfloor} \geqslant \alpha_m \\ 2.\ s_j \notin c_m \Rightarrow f_{j,\lfloor \frac{m+1}{2} \rfloor} < \alpha_m \end{cases} \tag{11}$$

Therefore, the assumption $s_j \notin c_m$ with condition $f_{j,\lfloor \frac{m+1}{2} \rfloor} > \alpha_m$ contradicts our initial hypothesis. Therefore, if $f_{i,\lfloor \frac{m+1}{2} \rfloor} > \alpha_m$ and $f_{j,\lfloor \frac{m+1}{2} \rfloor} > \alpha_m$, $s_i \in c_m$, s_j will be in the same cognitive category(c_m). In this way, condition $f_{i,\lfloor \frac{m+1}{2} \rfloor}$, $f_{j,\lfloor \frac{m+1}{2} \rfloor} \leq \alpha_m$ can also be proved.

4.1.3 Multichannel features

We perform preprocessing step to prepare data for the feature extraction process. While targeting to reduce the noise of the short texts, we conduct multiple procedures such as Stemming, Lemmatization, and Removal of stop words and exceptional characters. Subsequently, we extract multichannel features for the learning process. The embedding models can capture syntactic and semantic regularities within the corpus to assign one vector for each word. GloVe [15, 50] relies on the words co-occurrence matrix, and CBOW [51] computes the word vector based on the context. However, the resultant word vectors fail to infer emotion cues in short text contents. To address this issue, we incorporate the affective lexical knowledge bases to determine words' emotional orientation and utilize them as an independent channel in the feature extraction.

Let τ be the corpus containing the set of words associated with textual contents of an emotion annotated dataset, D^E. We can then tokenize each short text s_i into a set of words ($s_i = \{o_1, o_2, ..., o_{|s_i|}\}$) where $o_j \in \tau(j \in \{1, ..., | s_i |\})$ is the j^{th} word in short text s_i. We can represent the context-wise word vector corresponding to the j^{th} word, denoted as $\vec{O}_j^{\text{context}}$ by utilizing embedding function $\Omega: \tau \rightarrow \mathbb{R}^d$ that adopts the GloVe [15] model.

For extracting emotional aspects of words, let $L = \{l_\imath \mid \imath \in [1, ..., t]\}$ be the set of emotion lexicons and t as the number of emotion lexicons where DepecheMood++ [25] and NRC-EIL [24] infer the lexical vectors to designate each word with continuous scores for emotional or polarized orientations. Consequently, we can propose a hybrid vectorization process to include emotional aspects of the words, assuming $\vec{O}_j^{\text{emotion}} = [o_{j,i} \mid i \in [1, ..., k]]$ as an emotion-wise word vector associated to o_j. Here, k denotes the number of basic emotions and $o_{j,i}$ represents the i^{th} emotion scores for o_j.

$$\vec{O}_j^{\text{emotion}} = \bigoplus_{\imath=1}^{t} \Psi(o_j, l_\imath) \tag{12}$$

As Eq. 12 formalizes, the function $\Psi : \tau \rightarrow \mathbb{R}^k$ receives the o_j and retrieves the emotion scores through the lexicon knowledge-base $l_\imath$ where t denotes the number of emotion lexicons and $\bigoplus$ gives the concatenation of resultant vectors. We can support $| \vec{O}_j^{\text{emotion}} | = \sum_{\imath=1}^{t} \Phi(l_\imath)$ where Φ specifies the number of emotions leveraged in $l_\imath$. We apply POS-tagging [52] to infer structural elements and syntactic patterns. Moreover, we can utilize ϱ: $\tau \rightarrow \mathbb{R}^\mu$ to alter each short text with pertinent feature vectors $\vec{O}_j^{pos}$. Here, the o_j and μ as the size of the POS-tagged vector are the inputs.

In a nutshell, the attention-based mechanisms [53] aim to signify the words with higher impacts to foster classification procedures. As Fig. 3 shows, we adopt the attention module to construct a corresponding vector for each word and signify prominent focus words. Hence, we utilize the context and emotion-wise vectors to acquire the attention values and apply the resultant weights on context-wise vectors to determine the significance of the words. Given each short text s_i, we can use the weighted sum of word vectors to compute the corresponding attention vector, $\vec{O}_j^{\text{attention}}$ (Eq. 13).

$$\vec{O}_j^{\text{attention}} = \sum_{y \neq j} \alpha_{j,y} . \vec{O}_y^{\text{context}} \tag{13}$$

As verbalized in Eq. 13, $\alpha_{j,y}$ ($\alpha_{j,y} \geqslant 0$) is the attention weight subjected to $\sum_y \alpha_{j,y} = 1$ where "." denotes the element-wise multiplication. We compute the attention weight $\alpha_{j,y}$ through a softmax function as formulated in Eq. 14.

$$\alpha_{j,y} = \frac{1}{2}\left(\frac{\exp(\text{score}(\vec{O}_j^{\text{context}}, \vec{O}_y^{\text{context}}))}{\sum_y \exp(\text{score}(\vec{O}_j^{\text{context}}, \vec{O}_y^{\text{context}}))} + \frac{\exp(\varphi(\vec{O}_j^{\text{emotion}}, \vec{O}_y^{\text{emotion}}))}{\sum_y \exp(\varphi(\vec{O}_j^{\text{emotion}}, \vec{O}_y^{\text{emotion}}))} \right) \tag{14}$$

The $score(., .)$ function quantifies the degree of relevance between the j^{th} and y^{th} words, and φ is the similarity function that determines the correlation ratio between the word pairs according to the pertinent emotion-wise vectors. Equation. 15 explains how the $score(., .)$ computes the relevance between the given pair of word o_j and word o_y.

$$\begin{aligned} &\text{score}(\vec{O}_j^{\text{context}}, \vec{O}_y^{\text{context}}) \\ &\quad = W_a[tanh(W_z[\vec{O}_j^{\text{context}} \oplus \vec{O}_y^{\text{context}}])] \end{aligned} \tag{15}$$

We randomly initialize the weights W_a and W_z and jointly learn them during the training process. The higher sentiment relevance between the given words in emotion classification, the larger inner-weights will be. To this end, we employ the simple but effective cosine similarity to calculate the weights between vectors, $\varphi(\vec{O}_j^{\text{emotion}}, \vec{O}_y^{\text{emotion}})$.

We collectively fuse triplets of lexicon resources, POS tags, and embedding vectors to attain the multichannel features. To collectively form the multichannel features, on the one hand, we concatenate the emotion-wise and context-wise word vectors, and on the other hand, we combine the context-wise and POS-wise vectors. Equation. 16 formulates how one can merge various vectors, including emotion-wise, context-wise, and attention vectors.

$$\vec{\Upsilon}_j^{emo} = \vec{O}_j^{\text{attention}} \otimes \vec{O}_j^{\text{context}} \otimes \vec{O}_j^{\text{emotion}} \tag{16}$$

Here the $\otimes$ is the vector concatenation operator. We can utilize Eq. 17 to obtain $M_{s_i}^{emo} \in \mathbb{R}^{|s_i| \times (2d + \sum_{\imath=1}^{t} \Phi(l_\imath))}$ as the matrix of vectors for s_i, where $| s_i |$ is the number of words.

$$M_{s_i}^{emo} = \vec{\Upsilon}_1^{emo} \oplus \vec{\Upsilon}_2^{emo} \oplus ... \oplus \vec{\Upsilon}_{|s_i|}^{emo} \tag{17}$$

We combine three vectors of POS, context, and attention in Eq. 18 to further improve the accuracy.

$$\vec{\Upsilon}_j^{pos} = \vec{O}_j^{\text{attention}} \otimes \vec{O}_j^{\text{context}} \otimes \vec{O}_j^{pos} \tag{18}$$

Also, Eq. 19 computes $M_{s_i}^{pos} \in \mathbb{R}^{|s_i| \times (2d + \mu)}$ as the matrix for s_i.

$$M_{s_i}^{pos} = \vec{\Upsilon}_1^{pos} \oplus \vec{\Upsilon}_2^{pos} \oplus ... \oplus \vec{\Upsilon}_{|s_i|}^{pos} \tag{19}$$

Given $s_i \in S$ comprises the various number of vocabs, pertinent vectors of $M_{s_i}^{pos}$ and $M_{s_i}^{emo}$ will follow $| s_i |$. Hence,

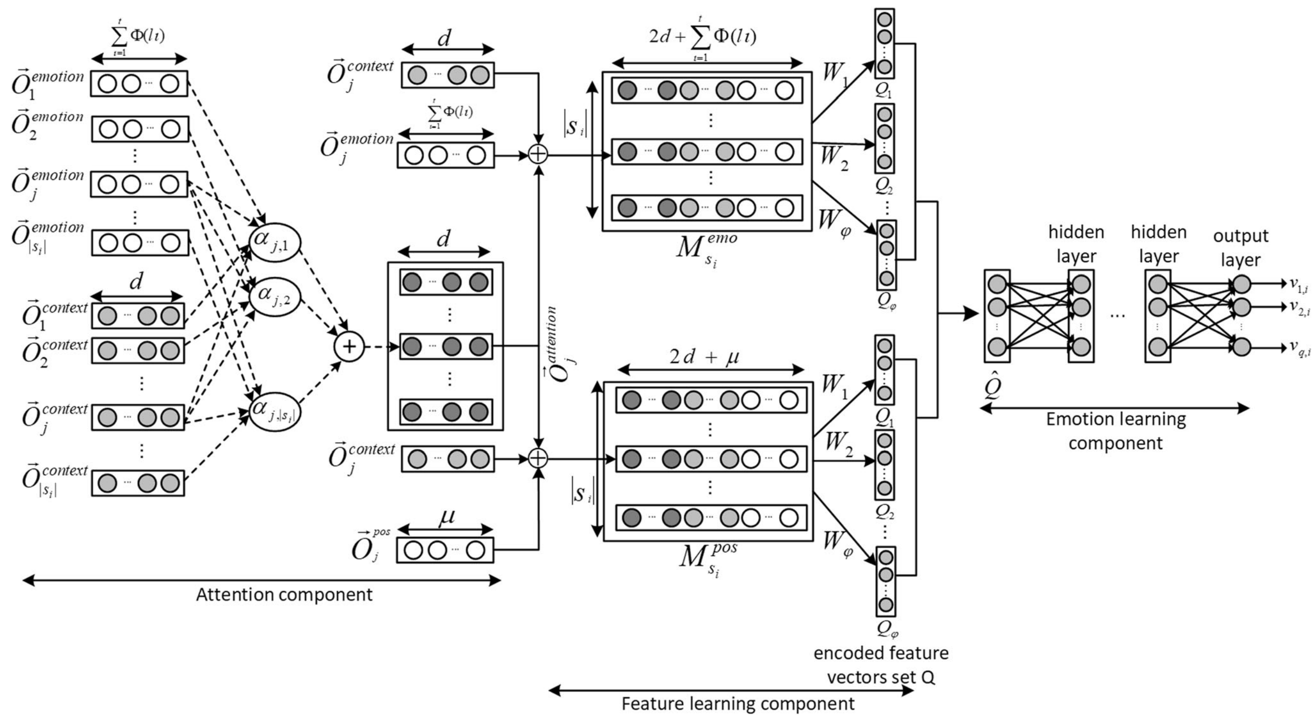


Fig. 3 Base classifier architecture

we adopt zero padding and use $\Gamma = *max_{\forall i=1,\ldots,m}(| s_i |)$ as the fixed length for M_{s_i} to unify short text matrices.

4.1.4 Cognitive ensemble

Given the dataset $D^E = \{(s_i, \vec{V}_{s_i}) \mid i = [1, \ldots, m]\}$ with m short texts, $\vec{V}_{s_i} \in \{0, 1\}^k$ can represent a binary emotion vector for s_i. Here, k denotes the number of emotion class labels. Moreover, where the j^{th} label of emotion is not null in $\vec{V}_{S_i}$, we assign $v_{i,j}$ with 1 or zero otherwise. Since s_i can be concurrently involved with diverse emotions, neither single-label classification models [54] nor regression algorithms [47] can model such multiplexity [34]. Hence, we employ multi-label classification by appointing each emotion label with a single task and adopting a parallel multi-task learner. An ensemble of convnet classifiers [54] can better learn the emotion features where we designate each cognitive category c_i to a distinctive multi-task classifier h_{c_i}. Finally, we aggregate the output of trained classifiers in H (Fig. 3).

Given the feature learning component, we appoint the width of the filters by the dimension of word vectors, denoted by $2d + \sum_{\iota=1}^{t} \Phi(l_\iota)$. We further alter the height of matrix $M_{s_i}^{emo}$ to acquire various sets for the encoded feature vectors. To this end, we obtain the encoded set of feature vectors $Q^{emo} = \{Q_1, Q_2, \ldots, Q_\varphi\}$ for the embedding matrix $M_{s_i}^{emo}$, by φ various window sizes, each denoted by $\delta_r \in \mathbb{N}$. Here $Q_r = [q_{r,j} \mid q_{r,j} \in \mathbb{R}$ and $j \in [1, \ldots, | s_i | + \delta_r - 1]]$ represent the feature vectors for the r^{th} window, Eq. 20.

$$\mathbf{q}_{r,j} = \phi(M^{emo}_{s_i\{j:j+\delta_r-1\}}.W_r + b) \tag{20}$$

$W_r \in R^{\delta_r \times (2d + \sum_{\iota=1}^{t} \Phi(l_\iota))}$ is the filter matrix, $b \in R^{\delta_r * 1}$ is a bias vector, and $M^{emo}_{s_i\{j:j+\delta_r-1\}}$ is the horizontal fragment for $M_{s_i}^{emo}$ of the size δ_r. The max layer uses output feature vector Q^{emo} to exploit the encoded vector $\hat{Q}^{emo}$ (Eq. 21).

$$\hat{Q}^{\text{emo}} = [\hat{q}_r | \text{r} \in [1, \ldots, \varphi] \text{ and } \hat{q}_r = * \max_{j \in [1, \ldots, |\text{s}_\text{i}| - \delta_r + 1]} (q_{r,j})] \tag{21}$$

Similarly, we acquire Q^{pos} and $\hat{Q}^{pos}$ associated with embedding matrix $M_{s_i}^{pos}$ where $\hat{Q}^{emo}$ and $\hat{Q}^{pos}$ can attain the output of the max layer $\hat{Q} = \hat{Q}^{emo} \otimes \hat{Q}^{pos}$.

Successively, the emotion learning component consumes $\hat{Q}$, where we collectively utilize the inter-connected layers to address the perceptual multi-label classification problem and segregate tasks in the output layer. Let $l \in \{1, .., L\}$ be the layer index of the network in the fully connected component. Given L as the number of hidden layers, the index zero can determine the input for the emotion learning module. Figure 4 shows the schema of a neuron in the hidden layer l where x^0 equates to $\hat{Q}$ and x^l denotes the load for l. w^l and z^l can also indicate the weighting matrix and the output of layer l to be fed into the next layer, i.e., $l+1$. Eq. 22 attains the input of the k^{th} neuron in layer l.

$$\mathbf{x}_k^l = b_k^l + \sum_{i=1}^{N_l-1} w_{ik}^{l-1} z_i^{l-1} \tag{22}$$

Here, l and $l-1$ are, respectively, the current and previous hidden layers. x_k^l and b_k^l can indicate the input and the bias values for the k^{th} neuron in layer l. The output of the i^{th} neuron in layer $l-1$ is z_i^{l-1} and w_{ik}^{l-1} associates the weights of the i^{th} neuron in $l-1$ to the k^{th} neuron in l. Aiming to prevent overfitting, we transform Eq. 22 into Eq. 23 to neutralize the effect of both nonessential features and trivial terms.

$$\mathbf{x}_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} \mathscr{P}(w_{ik}^{l-1}) z_i^{l-1} \tag{23}$$

Moreover, the function $\mathscr{P}(w_{ik}^{l-1})$ attains weights between a pair of neurons. Hence, we can pass x_k^l through activation function ReLU to retrieve the intermediate output z_k^l.

The output layer constitutes k output units, each dedicated to a single task. The output of the last layer in hidden layers, as the common feature representation learned for the k tasks, can be fed to the output layer. The constraint can be accommodated by Eq. 24, computing the network prediction output for the j^{th} emotion, denoted by $\hat{v}_{j,i} \in [0, 1]$.

$$\hat{v}_{j,i} = \frac{1}{1 + e^{-x_k^l}} \tag{24}$$

To reduce the error rate, we use the back-propagation algorithm. As formalized in Eq. 25, we then employ a modified binary cross-entropy to compute the joint loss function by the predicted labels.

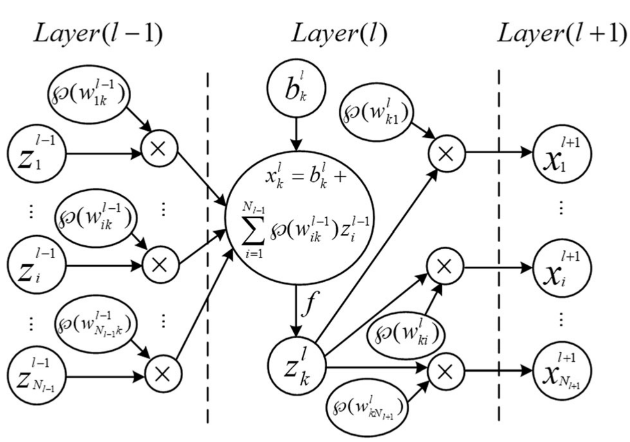


Fig. 4 Input and output of a neuron at hidden layer l

$$\mathcal{L} = \frac{1}{m} \sum_{i \in m} \left(-\sum_{j=1}^{k} v_{j,i} \log(\hat{v}_{j,i}) + (1 - v_{j,i}) \log(1 - \hat{v}_{j,i}) \right) \tag{25}$$

Here $\hat{v}_{j,i}$ is the prediction outcome, and $v_{j,i}$ denotes the ground truth for $e_i \in E$, associated with s_j. k and m, respectively, count the number of emotion labels and short texts. We apply Adam optimizer [55] that leverages both AdaGrad and RMSProp to update the weights and bias.

4.1.5 Weight regularization

As explained in Sect. 4.1.4, overfitting as a deep learning dilemma is caused by a contradiction where optimization aims to adjust the model to foster effectiveness, and generalization solely points toward inferring the unforeseen data. The dropout [56] is the most credible regularization approach to revoke the overfitting issue. However, resolving the tension between bias and variance is not a trivial task. To this end, the dropout approaches like DropConnect [57] eliminate the arbitrary weights and ignores the selected nodes in the connected layer.

The DropConnect algorithm (Alg. 1) turns out to be Naïve in the elimination process. Because it arbitrarily zeros out the selected weights. From another perspective, even the fixed dropout rate in DropConnect can reduce the model expressiveness and increase manual tuning requirements. Hence, we must initially infer the statistical cues from the embedding weights and then adjust the dropout rate consciously. To this end, we enhance the flexibility by retrieving the dropout rate based on the weights drawn from a data-specific uniform distribution.

Algorithm 1 General DropConnect

Input: $rate, w$
Output: $\acute{w}$

```
f = flatten(w) ,ẃ = w
l = []
while len(l) < rate do
    r = randint(0, | f |)
    if not r in l then
        l.append(r)
    end if
end while
for p in l do
    ẃ_{p/|f|, p%|f|} = 0
end for
return ẃ
```

In a tractable approach, we can refer to each weight in the bell curve to empirically initialize the elimination procedure. As depicted in Fig. 5, we can instantiate a weight matrix to preserve the value of the specified cell, removing or reducing the value in selected cells to adjust the activation process for neurons. Here we can reduce the value of the given point in the matrix according to trilateral coefficients. As coded in algorithm 2, we propose an efficient dropout technique to alter imperceptive arbitrary regularization with a distribution-aware model. We multiply the outputs of neurons, highlighted in Eq. 23, by the justified weights based on coefficients to gain $\acute{w}$ new weights. We then specify the drop-rate using dataset-oriented parameters, the standard deviation and the mean, denoted by μ and σ. As shown in Fig. 6, the weights are of three categories: *least* (λ), *minor* (β), and *common* (α). Due to the subtle connection between the significance of the neurons and the dropout, we sufficiently reduce the weights for the least and minor sections in the curve. This not only leads to a faster convergence rate but also reduces the activation weights, the overfitting cause. Also, we relatively increase the significance of weights in the minor and common regions, coefficients of μ and σ. The changes make our model more mature through subsequent epochs, avoiding the neuron outputs to excessively rely on the least and minor weights where the model can attain high-impact weights from the common section.

Algorithm 2 None Significant Weight reduction

Input: $w, \alpha, \beta, \lambda$
Output: $\acute{w}$

$f = flatten(w)$, $\acute{w} = w$
$\mu = \frac{1}{|f|}\sum_{i=1}^{|f|} f_i$, $\sigma = \sqrt{\frac{1}{|f|}\sum_{i=1}^{|f|}(f_i - \mu)^2}$
for i in $|f|$ **do**
  $r = i/|f|$, $c = i\%|f|$
  if $\mu - 2\sigma < f_i < \mu + 2\sigma$ **then**
    if $\mu - \sigma < f_i < \mu + \sigma$ **then**
      $\acute{w}_{r,c} = \alpha \times w_{r,c}$
    else
      $\acute{w}_{r,c} = \beta \times w_{r,c}$
    end if
  else
    $\acute{w}_{r,c} = \lambda \times w_{r,c}$
  end if
end for
return $\acute{w}$

Scalability: To promote vertical scalability, as verbalized in Eq. 26, we obtain O_{EmoDNN} by aggregating the time complexity of the comprising components.

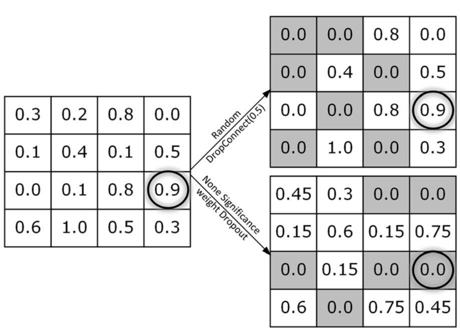


Fig. 5 Random dropout versus weight regularization

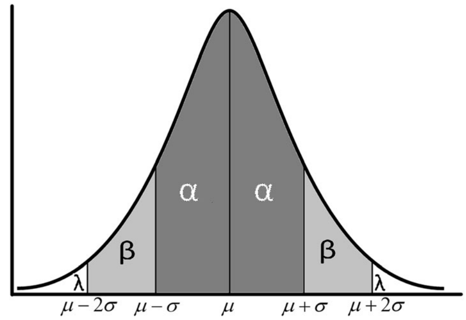


Fig. 6 Distribution of coefficient weights

$$O_{\text{emoDNN}} = O\left(\alpha m^3 + \beta\left(\sum_{l=1}^{k} n_{l-1}.\mid w_l\mid^2.n_l.Q_l^2\right).m.e\right) \tag{26}$$

Where n_l and n_{l-1} are, respectively, the number of filters and input channels of the l^{th} layer, $\mid w_l \mid$ can signify the filter length. Also, Q_l is the spatial size of the output feature and e counts the epochs. Note that we consider α, γ, and β as constant multipliers.

4.2 Online phase

For online section, with the cognitive vector $\vec{F}_{s_q}$ of the text s_q, the model approximates the emotion vector via two functions: Detecting the cognitive factors and Estimating the emotions.

Algorithm 3 Cognitive Aware Aggregation method

Input: $H, \alpha_j (j \in [1, ..., q]), \vec{F}_{s_q}$
Output: $\vec{V}_{s_q}$

$M_{s_i}^{emo}, M_{s_i}^{pos} = FeatureExtraction(s_q)$
for i in $[1, ..., q]$ **do**
  if $f_{s_q,i} < \alpha_i$ **then**
    $A_{:,i} = h_{2i-1}(M_{s_i}^{emo}, M_{s_i}^{pos})$
  else
    $A_{:,i} = h_{2i}(M_{s_i}^{emo}, M_{s_i}^{pos})$
  end if
end for
$A_{:,q+1} = h_{q+1}(M_{s_i}^{emo}, M_{s_i}^{pos})$
$\vec{V}_{s_q} = aggregation(A)$
return $\vec{V}_{s_q}$

4.2.1 Cognitive factors detection

Social media supply our propositional datasets. Hence, our framework must process high volume of brief contents in the Online phase. To meet efficacy requirements, we train the model in Sect. 4.1.1 infrequently and attain the cognitive vector $\vec{F}_{s_q}$ of the input query s_q using pre-trained model.

4.2.2 Cognitive aware aggregation method

In this step, we consume the cognitive vector of the input query to select a set of relevant base classifiers. We utilize inference algorithm 3 to collectively select the base classifiers and combine them accordingly.

In algorithm 3, we firstly extract multichannel features out of the input query s_q using the method in Sect. 4.1.3. Given cognitive vector $\vec{F}_{s_q}$, we can select either of the base classifiers, h_{2i} or h_{2i-1}. To this end, we compare each element $f_{s_q,i} \in \vec{F}_{s_q}$ to the splitter parameter α_i that divides the short text messages into various categories of low and high. By designating the whole corpus to the multi-task base-classifier h_{q+1}, we can disregard the cognitive factors in learning. We can subsequently predict the emotion vectors corresponding to s_q by applying the selected base-classifiers. This results in the consensus matrix A where each element $A_{i,j}$ represents the predicted class by the base classifier j for an emotion i. Each column j depicts the binary opinion of model j about each of emotions in $\vec{V}_{s_q}$.

We combine the base classifiers into binary values for the emotion classes, resulting in better approximation and improving the overall performance [58]. We then adopt the simple but effective majority voting method [59] to anticipate the outcomes. Given s_q, Eq. 27 predicts final sentiments where $A_{j,t}$ is the prediction of the classifier t via function $g(y, c)$ where g returns 1 for $y = c$ and 0 otherwise.

$$\vec{V}_{s_q} = \bigcup_{j=1}^{k} {*}\mathrm{argmax}_{c_i \in \{0,1\}} \left(\sum_{t=0}^{q+1} g\left(A_{j,t}, c_i\right) \right) \tag{27}$$

Scalability: It is noteworthy that the time complexity of the online section is $O(\sum_{l=1}^{k} n_{l-1}. \mid w_l \mid^2 .n_l.Q_l^2)$, mainly dedicated to exploiting emotions from a single short text.

5 Experiment

We performed various experiments on datasets [12, 14] to compare our model versus other novel methods in emotion detection. Taking advantage of various Python libraries and interfaces for neural networks, we ran the experiments using a computer with Intel Core i7-7700K CPU of 4.20 GHz, equipped with 64GB RAM. Also, the codes can be downloaded[1].

5.1 Data

We used three datasets to examine our method in detecting personality and emotions from brief contents.

- *MyPersonality* [14]: This dataset (D^P) is utilized to assign the cognitive labels to our target emotion dataset [12] and comprises the cues for EXT, AGR, CON, and NEU. We eliminate the effect of OPE due to minor significance.
- *SemEval2018 (Sem)* [12]: This dataset (D^E) is annotated by 11 emotion tags. Like Ekman's standard [60], we include fear, anger, joy, disgust, and sadness.
- *WASSA-2017 (WAS)* [13]: This dataset (D^E) includes fear, joy, sadness, and anger emotions. We utilize this emotion annotated dataset to investigate the performance of our proposed model in multi-class labeling. We enclose the statistics pertaining MyPersonality, SemEval2018, and WASSA-2017 in Tables 4 and 5, respectively.

Figure 7 shows short text distribution for each given cognitive factor, where the x-axis reports the weights and y counts the frequency. The middle threshold differentiates the low and high domains with respective light and dark colors.

5.2 Benchmark

We define hypothesis metrics to evaluate the effectiveness: we count a true positive (TP) when a short text has emotion e_i and the model correctly predicts the same. False positive (FP) is when the short text does not relate to the emotion e_i but the model predicts oppositely. Similar to TP and FP we can determine true negative (TN) and false negative (FN). To measure the performance, we use evaluation metrics, including accuracy, precision, recall, and F1-measure, which we derive from hypothetical parameters. We can then judge the best performance by F-measure via tenfold cross-validation. In multi-label problems, we achieve metrics for each emotion independently. The accuracy metric quantifies the ratio of short texts predicted correctly according to the truth label of the given emotion compared to all short texts in the corpus. Similarly, the precision measure obtains the ratio of short texts with the given emotion that the model predicts correctly. Through measuring recall, we can determine how effectively our proposed method can detect short texts containing the given emotion. F1 Score is the harmonic mean of the precision and recall.

[1] https://sites.google.com/view/cognition-computing-lab.

Table 4 Statistics pertaining MyPersonality dataset

#Tweet	NEU		CON		EXT		AGR	
	Low 6200	High 3717	Low 5361	High 4556	Low 5707	High 4210	Low 4649	High 5268
Max	4.75		5		5		5	
Min	1.25		1.45		1.33		1.65	
Mean	2.6		3.47		3.35		3.62	
STD	0.76		0.74		0.85		0.68	
Median	2.6		3.4		3.4		3.65	

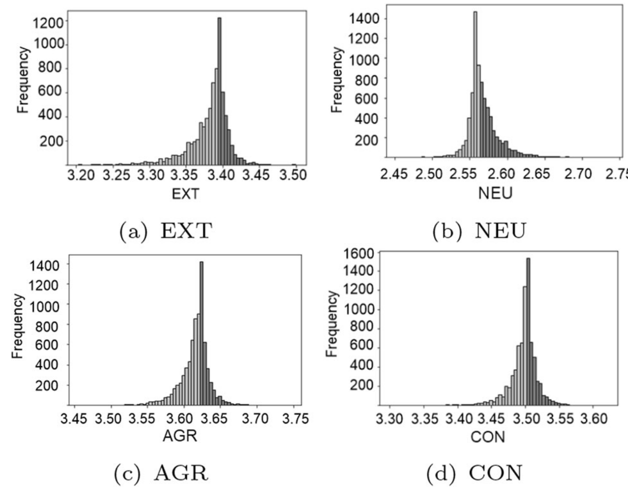


Fig. 7 Models

Table 6 Impact of batch-size on accuracy

Batch size	Anger	Disgust	Fear	Joy	Sadness
30	0.7985	0.7744	**0.9232**	0.7728	**0.8233**
50	0.8006	0.7645	0.9224	0.7801	0.8185
80	0.804	0.769	0.9199	0.7742	0.8154
100	0.8016	0.7729	0.9177	0.7791	0.8198
128	**0.8173**	**0.7767**	0.8939	**0.8305**	0.781
150	0.8015	0.7682	0.901	0.7682	0.7802

Best values are highlighted in bold

5.3 Baselines

We use the benchmark (Sect. 5.2) to measure the performance of the competitor methods in emotion recognition:

- *Unison*: This baseline [9] leverages a variety of deep learning modules, such as word and character-based RNN and CNN, to improve traditional classifiers including BOW and latent semantic indexing.
- *$Senti_{HC}$*: This model is a hierarchical classification scheme that comprises three levels in the learning process: neutrality (neutrality versus emotionality), polarity, and emotions (five basic emotions) [8].
- *SVM-Behavior*: Similar to [27], it combines unigrams and emotion lexicons, using SVM-Behavior to classify text by emotion cues.
- *Lexicon-Based*: Instead of word embedding [25], this model is performed by emotion lexicon.
- *SpanEmo* [40]: treats the multi-label emotion classification task as span prediction, helping emotion recognition models to learn the correlation between words and labels within a sentence.
- *MLE* [42]: is a robustly optimized BERT approach (transformer Networks) with multiple attention modules that expose the impact of each word on pertinent emotion.
- *Bert* [41]: is a bidirectional encoder representation from transformers that pre-trains representations from the unlabeled text.
- *SENN* [61]: utilizes LSTM and CNN to extract semantic and emotional features.
- *Convnet* [62]: drops the cognitive ensemble module to analyze the impact.
- *EmoML*: is the proposed categorization method that alters the learning component by SVM classifier.
- *$EmoDNN_{wd}$*: Replaces multichannel feature learning with text embedding [16, 50].
- *EmoDNN*: Our proposed framework (Sect. 3.3)

Table 5 Statistics of datasets (WAS and Sem)

#Tweet	Anger		Disgust	Fear		Joy		Sadness	
	Sem 2859	WAS 1701	Sem 2921	Sem 1363	WAS 2252	Sem 2877	WAS 1611	Sem 2273	WAS 1533
Max	1	0.976	1	1	1	1	0.98	1	1
Min	0	0.032	0	0	0.06	0	0	0	0.083
AVG	0.37	0.499	0.378	0.18	0.498	0.37	0.499	0.2943	0.501
STD	0.48	0.17	0.485	0.38	0.197	0.48	0.21	0.455	0.196
Median	0.0	0.479	0.0	0.0	0.479	0.0	0.5	0.0	0.485

Table 7 Impact of number of epoch on accuracy

epoch	Anger	Disgust	Fear	Joy	Sadness
40	0.8051	0.7667	0.9201	0.7727	0.817
50	**0.8173**	**0.7767**	0.8939	**0.8305**	0.781
80	0.8032	0.7627	**0.9202**	0.7749	0.8153
100	0.8035	0.7679	0.9176	0.7779	**0.8188**

Best values are highlighted in bold

Table 8 Impact of text-embedding on accuracy

glove	Anger	Disgust	Fear	Joy	Sadness
25	0.7955	0.7707	0.9112	0.7464	0.8058
50	0.8149	0.7629	0.9177	0.7329	**0.8222**
100	0.7996	0.7492	**0.92**	0.7697	0.8218
200	**0.8173**	**0.7767**	0.8939	**0.8305**	0.781

Best values are highlighted in bold

Table 9 Impact of learning rate on accuracy

learning rate	Anger	Disgust	Fear	Joy	Sadness
10^{-1}	0.7986	0.7574	0.9151	0.7665	0.8028
10^{-2}	0.7981	0.771	**0.9203**	0.7616	0.8182
10^{-3}	0.7948	0.7594	0.9191	0.7933	0.8086
10^{-4}	0.8041	0.7694	0.912	0.7768	**0.8184**
10^{-5}	0.8111	0.7715	0.8949	0.8297	0.767
10^{-6}	**0.8173**	**0.7767**	0.8939	**0.8305**	0.781
10^{-7}	0.8035	0.754	0.8912	0.8281	0.779

Best values are highlighted in bold

5.4 Effectiveness

5.4.1 Impact of learning parameters on emotion recognition

Given the importance of the batch size in the dynamics of deep learning algorithms, we designate this section to measure the accuracy for each given emotion where the batch size varies. Table 6 shows where the batch size varies the accuracy fluctuates up to 5% with minimum and maximum for disgust and joy emotions. As a result, we select the best value of 128 tweets for the batch size in our method to maximize the performance. Similarly, we have attained the best batch size for other rivals. Similarly, Table 7 investigates the impact of the number of epochs on the accuracy. Excluding the fear and sadness emotions, where the epoch is set to 50, we gain the best effectiveness.

Furthermore, we need to evaluate the embedding module. Hence, Table 8 reports the accuracy for various embedding dimensions where we opt for the value of 200 to get the best overall performance. In retrospect, the lower dimensions can better adjust to fewer data, like for fear and sadness. Because the higher the dimension in low sampling, the bigger the data sparsity will be. The learning rate parameter can significantly affect the robustness of the proposed model as it can directly influence the optimization weights. As for larger learning rates, the chance to exceed the extreme point will be bigger, causing an unstable system. Conversely, where the learning rate decreases, the training time can exquisitely take longer. As observed in Table 9, the learning rate of 10^{-6} results in higher accuracies.

5.4.2 Impact of multichannel features

To investigate the impact of various channels in feature extraction (Sect. 4.1.3), we report the combinatory performance of channels in Table 10. The results show that the context-wise uni-module gains the least performance. Moreover, equipping the context-wise channel with emotion and POS will not improve the baseline considerably. Compared to NRC-ENL, DepecheMood++ handles more aspects and gets trained with a more comprehensive corpus. However, with a slight loss in fear, the amount of improvement is minor. To conclude, we observe that the unanimous inclusion of the entire channel set can outperform other combinations, where the major raise is reflected in fear and joy, respectively, %2 and %1.

5.4.3 Impact of weight regularization

This section studies the impact of weight regularization on emotion recognition. Hence, comparing the performance of

Table 10 Impact of various channels on accuracy

Feature extraction	Anger	Disgust	Fear	Joy	Sadness
Context	0.8101	0.7705	0.87	0.82	0.776
Context+POS	0.8105	0.7712	0.875	0.8231	0.7764
Context+Emotion	0.8152	0.7741	0.8825	0.8272	0.7791
Context+Emotion (NRC-ENL)+ Attention	0.8123	0.7748	0.8917	0.8275	0.7805
Context+Emotion (DepecheMood++) +Attention	0.813	0.7756	0.889	0.8283	0.7807
Context+Emotions +POS+Attention	**0.8173**	**0.7767**	**0.8939**	**0.8305**	**0.781**

Best values are highlighted in bold

Table 11 Impact of regularization on accuracy

dropout method	Anger	Disgust	Fear	Joy	Sadness
No-dropout	0.7925	0.7538	0.8723	0.8094	0.7338
DropConnect	0.796	0.7588	0.8597	0.8099	0.7504
Dropout	0.799	0.7611	0.8646	0.815	0.7725
Weight regularization	**0.8173**	**0.7767**	**0.8939**	**0.8305**	**0.781**

Best values are highlighted in bold

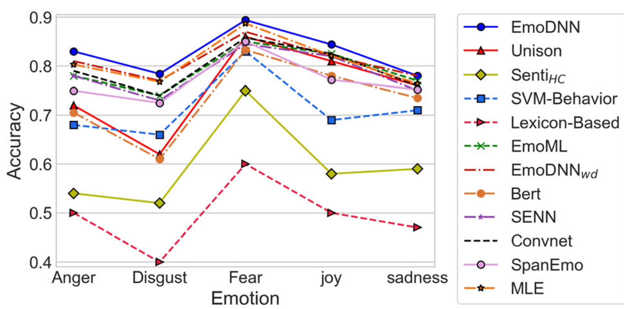


Fig. 8 Compare baselines accuracy

various regulatory methods, including No-Drop, Dropout [56], DropConnect [57], and weight regularization in Table 11 reveals that regularization methods can improve the results. Our method outperforms random dropout models, including DropConnect and Dropout, which neglect the influence of effective elements. However, our regularization module infers the statistical cues from weight distributions and adjusts the dropout consciously. Our model deliberately fluctuates the degree of weight impacts. For setting, we use 1.5, 1, and 0 for α, β, and λ.

5.4.4 Effectiveness of EmoDNN in multi-label dataset

We employ the benchmark (Sect. 5.2) to compare the rivals (Sect. 5.3) in inferring the emotions from brief contents. We observe in Fig. 8 that the performance of all the methods is more than 40%, which is even better for the neural network models. However, all versions of our proposed approach, including *EmoDNN*, *EmoML*, and *EmoDNN$_{wd}$*, turn out to be the best classifiers with an improvement of up to 3.2% versus the best performing competitor, MLE. MLE and SpanEmo baselines outperform ML-based methods and other essential deep neural networks models such as a convent and RNN. Nevertheless, our proposed method overpasses other competitors. Given the F1-Score metric, our proposed method surpasses the best second method, MLE, in all emotions and most emotion factors in Accuracy, with a partial decrease in fear, representing the EmoDNN with the best overall performance. From another perspective, lack of training procedure justifies why the lexicon-based methods attain

Table 12 Precision, Recall, and F-measure (F1) of each emotion in different methods

Methods	Anger (p value<0.005)			Disgust (p value<0.001)			Fear (p value<0.005)			Joy (p value<0.05)			Sadness (p value<0.005)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Unison	**0.8401**	0.275	0.4144	**0.9335**	0.2701	0.419	**0.839**	0.2842	0.4246	0.795	0.6811	0.7336	**0.835**	0.217	0.3444
Senti$_{HC}$	0.5365	0.5319	0.5342	0.5267	0.5197	0.5232	0.6636	0.6628	0.6632	0.5	0.5099	0.5049	0.5501	0.5553	0.5527
SVM-Behavior	0.676	0.6772	0.6766	0.6537	0.6562	0.6549	0.7245	**0.7818**	0.7521	0.6827	0.6853	0.684	0.5204	**0.7091**	0.6003
Lexicon-Based	0.3301	0.4034	0.3631	0.3429	0.4006	0.3695	0.535	0.6001	0.5649	0.3337	0.4825	0.3945	0.6042	0.3979	0.4798
SENN	0.7176	0.7084	0.7129	0.6492	0.658	0.6537	0.6842	0.6635	0.6737	0.7591	0.7302	0.7443	0.5961	0.54.3	0.5683
Convnet	0.7168	0.7562	0.7359	0.6708	**0.7254**	0.697	0.6941	0.6617	0.6675	0.8104	0.6964	0.7491	0.6125	0.5963	0.6043
Bert	0.7356	0.7269	0.7312	0.6815	0.6831	0.6823	0.7285	0.7357	0.7321	0.7348	0.7479	0.7413	0.6045	0.6035	0.604
SpanEmo	0.734	0.746	0.7401	0.6852	0.6989	0.692	0.738	0.7505	0.7442	0.762	0.7249	0.743	0.613	0.605	0.609
MLE	0.752	0.7381	0.745	0.7031	0.6969	0.7	0.7481	0.7785	**0.763**	0.772	0.7216	0.746	0.5808	0.6602	0.618
EmoML	0.7723	0.6101	0.6819	0.7048	0.5735	0.6324	0.7848	0.4018	0.5315	0.8149	0.6317	0.7117	0.6849	0.4122	0.5147
EmoDNN^{wd}	0.695	**0.8017**	0.7445	0.7736	0.6273	0.6928	0.6751	0.6359	0.6549	0.7498	**0.7548**	0.7523	0.6211	0.5348	0.5747
EmoDNN	0.7547	0.7545	**0.7546**	0.7041	0.7029	**0.7035**	0.7547	0.6047	0.6714	**0.8336**	0.6883	**0.754**	0.6445	0.6053	**0.6243**

Best values are highlighted in bold

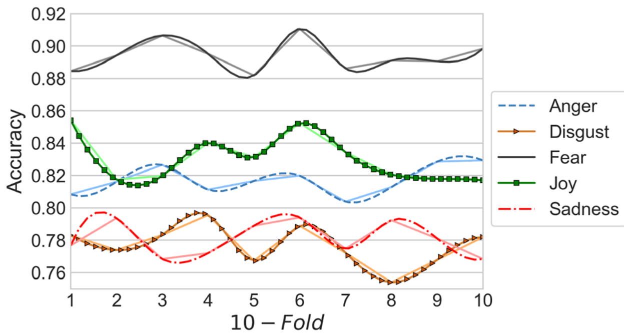


Fig. 9 Comparison of EmoDNN in various emotions

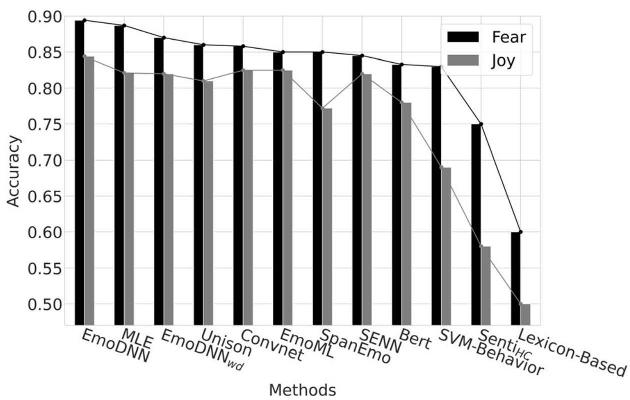


Fig. 10 Baselines Comparison in fear and joy emotions

Table 13 Compare baselines in multi-class dataset

Methods	Precision	Recall	F1 (p value<0.09)	Accuracy
Unison	0.847	0.834	0.841	0.85
SVM-Behavior	0.803	0.822	0.812	0.8
Convnet	0.8226	0.8275	0.825	0.8436
SENN	0.8286	0.8396	0.8341	0.8451
Bert	0.8525	0.8286	0.8404	0.8512
EmoDNN	**0.86**	**0.856**	**0.858**	**0.858**

Best values are highlighted in bold

the lowest accuracy. Even though we integrated our framework with shallow machine learning methods, e.g., SVM, the modified solution was still capable of overcome other baselines, where applying deep learning modules assured better accuracy. To prevent overfitting, we introduced a new improved dropout mechanism to foster the classification task, with further improvement of 1% compared to the arbitrary dropout. We also utilized a modified emotion-aware embedding approach instead of a pre-trained vector module, improving the accuracy by up to 1.07%.

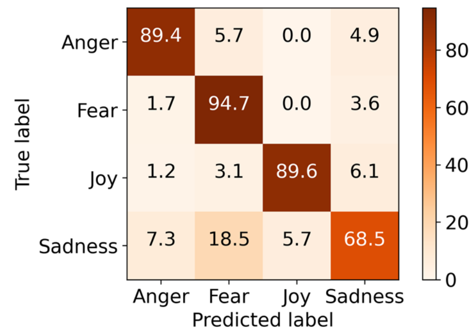


Fig. 11 Confusion matrix for emotion prediction

Accordingly, the results in Table 12 show that compared to all baselines, EmoDNN turns statistically significant in all emotions of anger, disgust, joy, and sadness, gaining reasonable F1-measures of 75.47%, 70.35%, 75.4%, and 62.43% for the respective p-values lower than 0.005, 0.001, 0.05, and 0.005. The modules behind reasons are threefold: cognitive inference, dynamic dropout, and emotion-aware feature representation. We also perform ablation tests to investigate the impact of different modules. The results show that $EmoDNN_{wd}$ and EmoDNN notably perform better than EmoML, demonstrating the positive impact of ensemble convnet on the emotion recognition procedure. Also, EmoDNN shows superiority over the competitors in ablation tests, including both $EmoDNN_{wd}$ and convnet, due to incorporating emotion cues in the multichannel feature embedding and further leverage of attention-based component in nominating prominent focus words. Finally, EmoDNN proves better effectiveness compared to convnet, rooted in the utilization of cognitive cues in the learning process.

Aiming to test out-of-samples in various folds, we study how the accuracy of our proposed method fluctuates in different emotions. We observe (Fig. 9) that fear and disgust gain the best and the least accuracies. While recognition of fear and joy is convenient, the detection of sadness and disgust is tedious in brief contents. We leverage the intuition in Fig. 10 to compare the effect of the two highest accuracies based on fear and joy, where EmoDNN surpasses other rivals and the lexicon-based attains the least accuracy.

5.4.5 Effectiveness of EmoDNN in multi-class dataset

We opt for the multi-class WASSA-2017 [13] dataset to evaluate mutual co-existed labels. Therefore, we compare our method with other competitors that address the multi-class challenge, conducting a t test comparison between EmoDNN and Unison, the best rival. Table 13 shows that EmoDNN has statistically significant superiority over Unison (p value < 0.09). Moreover, EmoDNN overcomes all competitors where those equipped with deep learning modules further overpass conventional classifiers such as SVM. Since we include the cognitive cues in emotion features, EmoDNN can upgrade Unison by up to 2% in the F1-measure.

Figure 11 illustrates the prediction confusion matrix for our model, where the weights show the percentage of the correctly predicted samples. EmoDNN has successfully recognized fear in 94.7% of the labeled tweets. Where the highest prediction performance is for fear and sadness is the least, 68.5% of the correct labels. Evidently, the sadness is mostly misclassified as fear in more than 18% of the cases where the least incorrect labels for joy is 5.7%. Predicting fear or anger by the joy emotion is impossible, resulting in a 0.0% chance. Hence, there is no confusion between the mentioned emotions with joy.

6 Conclusion

Our proposed unified framework in this paper leverages individual cognitive cues to recognize emotions from short text contents. Most previous efforts on emotion recognition disregard user-specific characteristics. To fill the gap, we firstly categorize short texts according to the cognitive cues. Subsequently, we then utilize the emotion lexicons alongside embedding models to obtain the emotion-aware short text vectors. Consequently, we learn corresponding base classifiers and employ a novel ensemble learning approach to aggregate the classification outputs. Our proposed framework achieves the accuracies of 0.82, 0.78, 0.894, 0.843, and 0.78, respectively, for the anger, disgust, fear, joy, and sadness emotions. Hence, the results from extensive experiments on real-world datasets confirm the superiority of our proposed framework over state-of-the-art rivals in emotion recognition. However, we need to integrate transfer learning to make inner ensemble classifiers better collaborate. Moreover, we will have to empirically study the effect of various distributions on the proposed dropout module. We leave these tasks for future.

Data availability SemEval2018 and WASSA-2017 datasets are available at competitions.codalab.org. Also, MyPersonality dataset is provided at http://mypersonality.org.

Declarations

Conflict of interest The authors declare that they have no conflict of interest

References

1. Phan D-A, Matsumoto Y, Shindo H (2018) Autoencoder for semisupervised multiple emotion detection of conversation transcripts. IEEE Trans Affective Comput 12:682
2. Budiharto W, Meiliana M (2018) Prediction and analysis of Indonesia presidential election from twitter using sentiment analysis. J Big Data 5:1–10
3. Corazza M, Menini S, Cabrio E, Tonelli S, Villata S (2020) A multilingual evaluation for online hate speech detection. ACM Trans on Internet Tech 20:1–22
4. Desmet B, Hoste V (2013) Emotion detection in suicide notes. Expert Syst Appl 40:6351
5. Rahat Ullah, Naveen Amblee, Wonjoon Kim, Hyunjong Lee (2016) From valence to emotions: exploring the distribution of emotions in online product reviews. DSS 81:41
6. Bollen J, Mao H, Zeng X (2011) Twitter mood predicts the stock market. J Comput Sci 2:1–8
7. Rahmani S, Hosseini S, Zall R, Kangavari MR, Kamran S, Hua W (2022) Transfer-based adaptive tree for multimodal sentiment analysis based on user latent aspects. Knowledge-Based Syst 261:110219
8. Esmin Ahmed AA, De Oliveira Jr Roberto L, Matwin S (2012) Hierarchical classification approach to emotion recognition in twitter. In 11th International Conference on ML and Apps, volume 2. IEEE
9. Colneric N, Demsar J (2018) Emotion recognition on twitter: comparative study and training a unison model. IEEE Trans Affective Comput 11:433
10. Watson D, Clark LA (1992) On traits and temperament: general and specific factors of emotional experience and their relation to the five factors model. J Personality 60(2):441–476
11. Holtgraves T (2011) Text messaging, personality, and the social context. J Res Personality 45:92
12. Mohammad S, Bravo-Marquez F, Salameh M, Kiritchenko S (2018) Semeval-2018: affect in tweets. In Procceeding of the 12th workshop on semantic evaluation
13. Saif M (2017) Mohammad and Felipe Bravo-Marquez. Emotion intensities in tweets. In Conference Lexical Computer, Semantics
14. Kosinski M, Stillwell D, Graepel T (2013) Private traits and attributes are predictable from digital records of human behavior. Proc Nat Acad Sci 110(15):5802–5805
15. Najafipour S, Hosseini S, Hua W, Mohammad RK, and Xiaofang Z (2020) Short-text author linking through multi-aspect temporal-textual embedding. IEEE TKDE, Soulmate
16. Saeid H, Saeed N, Ngai-Man C, Hongzhi Y, Mohammad Reza K, Xiaofang Z (2020) time-aware text embedding approach to generate subgraphs. Data Min Knowl Discov 34:1136–1174
17. Hosseini S, Unankard S, Zhou X, Sadiq S (2014) Location oriented phrase detection in microblogs. In DASFAA, pages 495–509
18. Golbeck J, Robles C, Edmondson M, Turner K (2011) Predicting personality from twitter. In Proc. 3rd IEEE International Conference on Social Computing, pages 149–156. IEEE
19. Alam F, Stepanov E A, Riccardi G (2013) Personality traits recognition on social network-facebook. In Seventh Int. AAAI Conference on Weblogs and Social Media
20. Quercia D, Kosinski M, Stillwell D, Crowcroft J (2011) Our twitter profiles, our selves: Predicting personality with twitter. In: IEEE International Conference on Social Computer
21. Xue D, Wu L, Hong Z, Guo S, Gao L, Wu Z, Zhong X, Sun J (2018) Deep learning-based personality recognition from text posts of online social networks. Appl Intell 48:4232
22. Sun X, Liu B, Cao J, Luo J, Shen X (2018)Who am i? personality detection based on deep learning for texts. In: IEEE ICC

23. Majumder N, Poria S, Gelbukh A, Cambria E (2017) Deep learning-based document modeling for personality detection from text. IEEE Intell Syst 32:74
24. Mohammad Saif M (2018) Word affect intensities. In Proceeding of the 11th Edition of the Language Re-sources and Evaluation Conference (LREC-2018),Miyazaki, Japan
25. Araque O, Gatti L, Staiano J, Guerini M (2019) Depechemood++: a bilingual emotion lexicon built through simple yet powerful techniques. IEEE Trans Affective Comput 13:496
26. Tao Jianhua (2004) Context based emotion detection from text input. In: Eighth International Conference on Spoken Language Processing
27. Jain VK, Kumar S, Fernandes SL (2017) Extraction of emotions from multilingual text using intelligent text processing and computational linguistics. J Comput Sci 21:316
28. Xu H, Yang W, Wang J (2015) Hierarchical emotion classification and emotion component analysis on Chinese micro-blog posts. Expert Syst Appl 42:8745
29. Li W, Xu H (2014) Text-based emotion classification using emotion cause extraction. Expert Syst Appl 41:1742
30. Bandhakavi A, Wiratunga N, Padmanabhan D, Massie S (2017) Lexicon based feature extraction for emotion text classification. Pattern Recognit Lett 93:133
31. Udochukwu O, He Y (2015) A rule-based approach to implicit emotion detection in text. In: International Conference on Apps of NL to Information System
32. Canales L, Carlo Strapparava, Ester Boldrini, Patricio Martinez-Barco (2017) Intensional learning to efficiently build up automatically annotated emotion corpora. IEEE Trans Affective Comput 11:335
33. Halim Z, Waqar M, Tahir M (2020) A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. Knowledge-Based Syst 208:106443
34. Deng J, Ren F(2020) Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning. IEEE Trans Affective Comput
35. Erdenebileg B, Meijing L, Ho Ryu K (2019) Semantic-emotion neural network for emotion recognition from text. IEEE Access 7:111866–111878
36. Cai L, Hu Y, Dong J, Zhou S (2019)Audio-textual emotion recognition based on improved neural networks. Math Probl Eng 1–19
37. Wang X, Kou L, Sugumaran V, Zhang H (2020) Emotion correlation mining through deep learning on natural language text. IEEE Trans Cybern 51:4400
38. Rong H, Ma T, Cao J, Tian Y, Al-Dhelaan A, Al-Rodhaan M (2019) Deep rolling: a novel emotion prediction model for a multi-participant communication context. Inf Sci 488:158
39. Kumar P, Raman B (2022) A bert based dual-channel explainable text emotion recognition system. Neural Netw 150:392–407
40. Alhuzali H, Ananiadou S (2021) SpanEmo: Casting multi-label emotion classification as span-prediction. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics
41. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In North American Association for Computational Linguistics
42. Ameer I, Bolucu N, Siddiqui MHF, Can B, Sidorov G, Gelbukh A (2023) Multi-label emotion classification in texts using transfer learning. Expert Syst Appl 213:118534
43. Deng J, Ren F (2021) A survey of textual emotion recognition and its challenges. IEEE Trans Affective Comput
44. Hosseini S, Yin H, Zhang M, Elovici Y, Zhou X (2018) Mining subgraphs from propagation networks through temporal dynamic analysis. In: IEEE MDM, pages 66–75
45. Rosalind W (2000) Affective computing. MIT press, NY
46. Zall R, Kangavari MR (2022) Comparative analytical survey on cognitive agents with emotional intelligence. Cognit Comput 45:1–24
47. Awad M, Khanna R (2015) Support vector regression. In: Efficient learning machines, pages 67–80. Springer
48. Aman S, Szpakowicz S(2008) Using rogets thesaurus for fine-grained emotion recognition. In: proceeding of the 3rd Inter. Joint Conference on Natural Language Processing
49. Milligan Glenn W, Martha C (1987) Clustering methods. Appl Psychol Measure, Methodol Rev
50. Pennington J, Socher R, Christopher D (2014) Manning. Glove: Global vectors for word representation. In: Proceeding of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)
51. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: International Confernce on Learning Representations
52. Manning C D, Surdeanu M, Bauer J, Finkel J R, Bethard S, McClosky D (2014) The stanford corenlp natural language processing toolkit. In Proceeding of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations
53. Zhao Z, Wu Y (2016) Attention-based convolutional neural networks for sentence classification. In: INTERSPEECH
54. Wei Y, Xia W, Lin M, Huang J, Ni B, Dong J, Zhao Y, Yan S (2015) Hcp: a flexible cnn framework for multi-label image classification. IEEE Trans Pattern Anal Mach Intell 65:1901–1907
55. Diederik P, Jimmy B (2015) A method for stochastic optimization. In Int Conf. on Learning Representations, Adam
56. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929
57. Wan L, Zeiler M, Zhang S, Le Cun Y, Fergus R (2013) Regularization of neural networks using dropconnect. In: International Conference on ML, pages 1058–1066
58. Zall R, Kangavari MR (2019) On the construction of multi-relational classifier based on canonical correlation analysis. Int J Artif Intell I:17
59. Zall R, Keyvanpour MR (2016) Semi-supervised multi-view ensemble learning based on extracting cross-view correlation. Adv Elec Comp, Eng 25:8
60. Ekman P (1999) Basic emotions. Handbook of cognition and emotion 98(45–60):16
61. Batbaatar E, Li M, Ryu KH (2019) Semantic-emotion neural network for emotion recognition from text. IEEE Access 7:111866–111878
62. Feng Y, Cheng Y (2021) Short text sentiment analysis based on multi-channel cnn with multi-head attention mechanism. IEEE Access 25:87

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.