Depression Detection via Capsule Networks with Contrastive Learning

Han Liu[1], Changya Li[1], Xiaotong Zhang[1]*, Feng Zhang[2], Wei Wang[3], Fenglong Ma[4], Hongyang Chen[5], Hong Yu[1], Xianchao Zhang[1]

[1] Dalian University of Technology, Dalian, China
[2] Peking University, Beijing, China
[3] Shenzhen MSU-BIT University, Shenzhen, China
[4] The Pennsylvania State University, Pennsylvania, USA
[5] Zhejiang Lab, Hangzhou, China
liu.han.dut@gmail.com, lichangya.dut@gmail.com, zxt.dut@hotmail.com, zfeng.maria@gmail.com, ehomewang@ieee.org, fenglong@psu.edu, dr.h.chen@ieee.org, {hongyu, xczhang}@dlut.edu.cn

Abstract

Depression detection is a challenging and crucial task in psychological illness diagnosis. Utilizing online user posts to predict whether a user suffers from depression seems an effective and promising direction. However, existing methods suffer from either poor interpretability brought by the black-box models or underwhelming performance caused by the completely separate two-stage model structure. To alleviate these limitations, we propose a novel capsule network integrated with contrastive learning for depression detection (DeCapsNet). The highlights of DeCapsNet can be summarized as follows. First, it extracts symptom capsules from user posts by leveraging meticulously designed symptom descriptions, and then distills them into class-indicative depression capsules. The overall workflow is in an explicit hierarchical reasoning manner and can be well interpreted by the Patient Health Questionnaire-9 (PHQ9), which is one of the most widely adopted questionnaires for depression diagnosis. Second, it integrates with contrastive learning, which can facilitate the embeddings from the same class to be pulled closer, while simultaneously pushing the embeddings from different classes apart. In addition, by adopting the end-to-end training strategy, it does not necessitate additional data annotation, and mitigates the potential adverse effects from the upstream task to the downstream task. Extensive experiments on three widely-used datasets show that in both within-dataset and cross-dataset scenarios our proposed method outperforms other strong baselines significantly.

Introduction

Depression is a serious health condition which can have severe consequences (Hu et al. 2020; Zhang et al. 2022; Zhang, Yang, and Ananiadou 2023), including emotional distress, social withdrawal, and even suicide. Despite the widespread awareness of its gravity, there are still many individuals with depression who have not been identified. As people usually tend to express their emotions on social media, utilizing user online posts to identify early depression is a promising research direction. This type of automated diagnosis can not only assist patients in detecting early signs of depression tendencies but also provide doctors with valuable references to aid in their assessment and treatment.

Several methods (Yates, Cohan, and Goharian 2017; Wolohan et al. 2018) attempt to predict whether a user suffers from depression based on online posts. They utilize convolutional neural networks or the n-gram and LIWC features to extract post representations, and use a neural network or traditional classifier to fulfill the prediction. Although they can obtain promising results, these black-box models face a significant challenge in terms of interpretability. Specifically, by using these models, people cannot gain a deeper understanding about how they accurately detect depression signs from online posts. In clinical applications and medical decision-making, the lack of interpretability may hinder the model credibility and reliability seriously.

Recently, Nguyen et al. (2022) propose an effective depression detection method which contains two components: a questionnaire model used to detect the presence of symptoms from PHQ9 (Kroenke, Spitzer, and Williams 2001), and a depression model used to predict the label. This work has shown impressive performance and can generalize to similar datasets well. However, it still exists the following issues. First, it employs a two-stage training strategy, which inevitably causes the effectiveness of the upstream questionnaire model directly affects the performance of the downstream depression model. Second, it relies heavily on large amounts of human-labeled data to train the questionnaire model, which is labor-intensive and time-consuming.

In this paper, we propose a novel capsule network integrated with contrastive learning for depression detection (DeCapsNet). As illustrated in Figure 1, DeCapsNet first collects the representative posts for each user by leveraging the elaborately designed depression symptom descriptions according to the PHQ9, which is a questionnaire widely used by clinicians in the depression screening process. Then DeCapsNet extracts symptom capsules from representative posts with the attention mechanism, and integrates these symptom capsules into depression capsules with dynamic routing for the subsequent depression detection. In addition, by integrating with contrastive learning, DeCapsNet can obtain more class-indicative embeddings, thus boosting the performance greatly. The overall framework of DeCapsNet is in an explicit hierarchical reasoning manner, and employs

*Corresponding author.

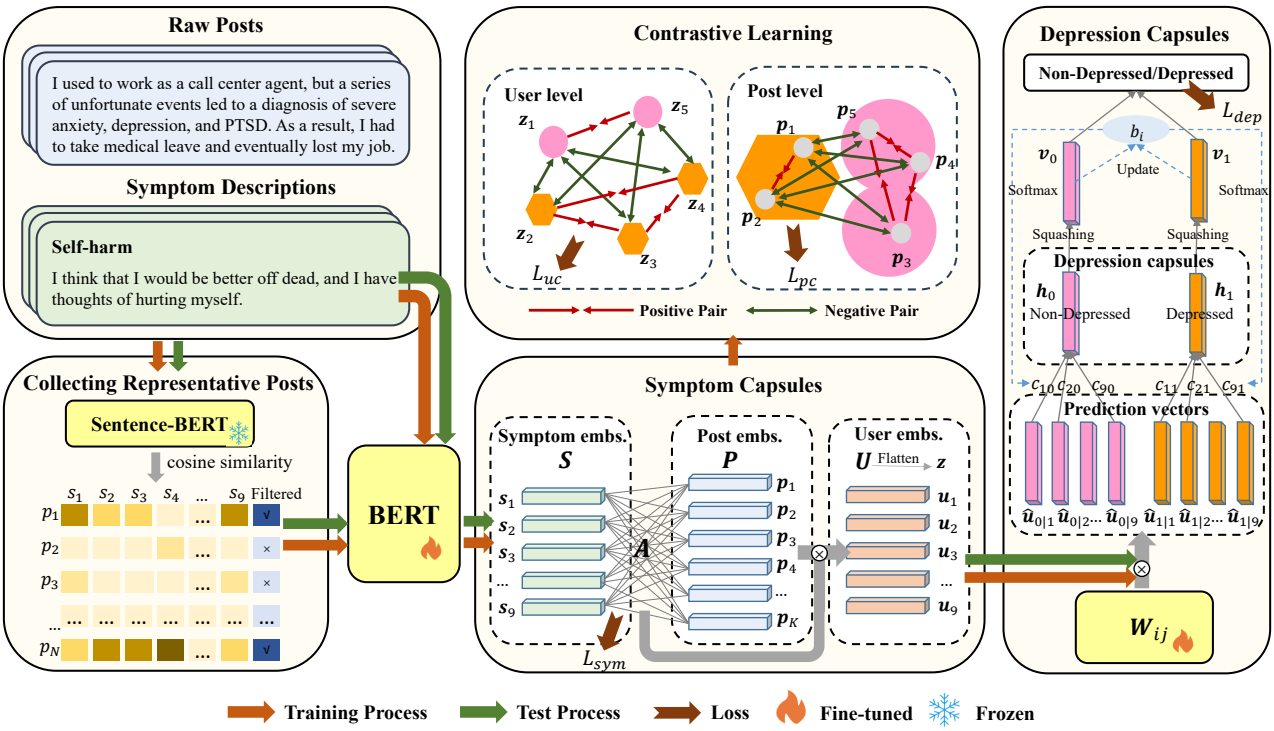


Figure 1: Illustration of our proposed framework.

an end-to-end training strategy. Extensive experiments show that DeCapsNet can achieve state-of-the-art performance in comparison with other strong baselines[1].

Related Work

Depression Detection Depression detection has become a hot topic in recent years, and several methods have been proposed to solve this problem based on online user posts. Yates, Cohan, and Goharian (2017) use the convolutional neural networks to extract post features and then feed them into the multilayer perceptron for classification. Wolohan et al. (2018) attempt to utilize the n-gram features and LIWC features (Tausczik and Pennebaker 2010) to fulfill the task. However, these methods lack of interpretability due to the black-box models and cannot generalize well over different datasets. Nguyen et al. (2022) propose an approach based on the PHQ9 questionnaire to address the interpretability and generalization issues. It consists of a questionnaire model which relies on additional annotated data to train symptom classifiers and a depression model which can predict whether a user is depressed based on the output of the questionnaire model. However, it is a two-stage method, which means the performance of the former questionnaire model will affect the effectiveness of the latter depression model. In addition, training the questionnaire model requires annotating numerous data, which is expensive and laborious.

[1]The source code is publicly available at the following link https://github.com/DeCapsNet/DeCapsNet-demo.

Capsule Networks Capsule networks aim to address the limitations of traditional convolutional neural networks in computer vision, which are initially proposed by Sabour, Frosst, and Hinton (2017) and further improved in (Hinton, Sabour, and Frosst 2018). Capsule networks consist of multiple capsule layers, where each capsule layer is divided into many small groups of neurons called capsules. Information transmission between capsule layers includes two stages: voting and routing. During voting, low-level capsules cast votes for high-level ones, and in the routing stage, the dynamic routing algorithm automatically determines the coefficients to facilitate this process. Recently, capsule networks have also been explored in natural language processing. Xia et al. (2018) and Liu et al. (2019) use capsule networks to deal with the zero-shot intent detection problem. Cho et al. (2019) propose a multi-document summarization method with the improved similarity measure inspired by capsule networks for determining sentence redundancy. Zhao et al. (2019) introduce an adaptive optimizer into capsule networks to adjust the number of iterations for each sample, which has obtained satisfactory performance in low-resource scenarios.

Contrastive Learning Contrastive learning applied to self-supervised representation learning has seen a resurgence in recent years, leading to impressive performance in computer vision (He et al. 2020; Chen et al. 2020) and natural language processing (Yan et al. 2021; Gao, Yao, and Chen 2021). Khosla et al. (2020) propose the supervised contrastive learning, which extends the contrastive learning to the fully supervised setting. Specifically, it employs

Symbol	Explanation
x	A user with multiple posts
y	The class label of user x
K	The number of risky posts
S	The symptom description set
$\boldsymbol{P}$	The embedding matrix of risky posts
$\boldsymbol{S}$	The embedding matrix of S
$\boldsymbol{A}$	The attention weight matrix
$\boldsymbol{U}$	The user symptom representation matrix
$\boldsymbol{W}_{ij}$	The transformation matrix which is learnable
$\boldsymbol{h}_j$	The depression capsule of j-th class
$\boldsymbol{v}_j$	The output of $\boldsymbol{h}_j$ after squashing function

Table 1: Symbol explanation.

an improved loss function, which can leverage the label information effectively and allow samples of the same distribution to lie close together in the latent space, while samples belonging to disparate classes are repelled in the latent space. In this paper, we attempt to use this technique to improve the embedding learning for depression detection.

The Proposed Method

Problem Formulation. Depression detection aims to identify whether a user suffers from the depression issue based on his/her history posts on social media (Nguyen et al. 2022). To ease understanding, we formalize this task as follows. Given a user x with N posts, x can be represented by $x = \{p_1, p_2, ..., p_N\}$, and p_j is the j-th post written by user x. The depression detection task is to predict a binary label $y \in \{0, 1\}$, which indicates whether user x is depressed. $y = 0$ and $y = 1$ mean that user x is non-depressed (control) or depressed respectively. Table 1 summarizes some symbol explanation in detail.

Collecting Representative Posts

In general, each user usually has numerous posts on social media, thereby inevitably containing some noisy posts which are meaningless for depression detection. To alleviate the above issue, we propose to collect the representative posts (i.e., risky posts) for each user by leveraging the depression symptom descriptions. In this paper, we utilize the Patient Health Questionnaire-9 (PHQ-9) (Kroenke, Spitzer, and Williams 2001) to generate the depression symptom descriptions, where PHQ-9 is one of the most widely adopted questionnaires for depression diagnosis. We follow PHQ-9 to divide the depression into 9 symptoms, and the designed depression symptom descriptions are shown in Table 2.

During the collecting representative posts procedure, given a user x with N posts $x = \{p_1, p_2, ..., p_N\}$ and the symptom description set $S = \{s_1, s_2, ..., s_9\}$, we first encode the posts and the symptom descriptions by utilizing the pre-trained Sentence-BERT (Reimers and Gurevych 2019), thus obtaining their corresponding embeddings. We represent them as $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N\}$ and $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_9\}$, where the letters in bold mean the embeddings. Then for each post p_i. we can calculate its risky score

Symptom	Description
Anhedonia	I have little interest or pleasure in dong things.
Mood	I always feel down, depressed and hopeless.
Sleep	Sometimes I have trouble falling asleep, sometimes I sleep too much.
Fatigue	I feel tired and have little energy.
Eating	Sometimes my appetite is poor, sometimes I cannot stop overeating.
Self-esteem	I feel bad about myself, think myself a failure. And I have let other people down.
Concentration	I have trouble concentrating on things.
Psychomotor	I move and speak much slower than before, but sometimes I have been moving around a lot more than usual.
Self-harm	I think that I would be better off dead, and I have thoughts of hurting myself.

Table 2: The detailed depression symptom descriptions.

r_i as follows:

$$r_i = \sum_{j=1}^{9} \frac{\boldsymbol{p}_i^T \boldsymbol{s}_j}{||\boldsymbol{p}_i||\ ||\boldsymbol{s}_j||}, \tag{1}$$

where $\boldsymbol{p}_i^T$ is the transpose of $\boldsymbol{p}_i$. $\frac{\boldsymbol{p}_i^T \boldsymbol{s}_j}{||\boldsymbol{p}_i||\ ||\boldsymbol{s}_j||}$ is the cosine similarity between $\boldsymbol{p}_i$ and $\boldsymbol{s}_j$. For user x, when obtaining the risky score for each post, we select its top K posts with the highest risk scores as the representative posts for the downstream depression detection task. Then for a user x, it can be rewritten as $x = \{p_1, p_2, ..., p_K\}$.

Depression Capsule Networks

Capsule networks have shown to be effective in various tasks, such as image classification (Sabour, Frosst, and Hinton 2017), intent identification (Xia et al. 2018) and so on. In this paper, we propose to modify the structure of capsule networks for depression detection. The proposed depression capsule networks contain two types of capsules: symptom capsules and depression capsules. Symptom capsules aim to extract interpretable symptom features from the user posts. Depression capsules are geared to aggregate symptom features to form a higher-level representation which can be directly used for depression detection.

Symptom Capsules Given a user $x = \{p_1, p_2, ..., p_K\}$ with K representative posts and the symptom description set $S = \{s_1, s_2, ..., s_9\}$, we can use any pre-trained language model to encode these posts and symptom descriptions, and then obtain the embedding matrix of the risky posts from user x which is represented as $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_K]^T \in \mathbb{R}^{K \times d}$, and the embedding matrix of the symptom descriptions which is represented by $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_9]^T \in \mathbb{R}^{9 \times d}$.

Based on the PHQ-9, we can assume that each user can be represented with 9 symptom features (capsules), and different posts usually contribute distinctively to each symptom capsule. To extract the symptom features for each user accurately, the key point is to assign appropriate importance

Algorithm 1: Dynamic routing algorithm

procedure ROUTING($\hat{u}_{j|i}$,$iter$)
  for symptom capsule i and depression capsule j: $b_{ij} \leftarrow 0$
  for $iter$ iterations **do**
    for all symptom capsule i: $c_i \leftarrow \text{softmax}(b_i)$
    for all depression capsule j: $\boldsymbol{h}_j \leftarrow \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i}$
    for all depression capsule j: $\boldsymbol{v}_j \leftarrow squash(\boldsymbol{h}_j)$
    for all symptom capsule i and depression capsule j:
    $b_{ij} \leftarrow b_{ij} + \hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$
  end for
  return $\boldsymbol{v}_j$
end procedure

weight for each post. In this paper, we exploit the attention mechanism (Vaswani et al. 2017) to automatically learn different attention scores for different posts. Specifically, we first map $\boldsymbol{S}$ to the query matrix $\boldsymbol{Q}$ by using the linear projection matrix $\boldsymbol{W}_q$, and map $\boldsymbol{P}$ to the key and value matrices $\boldsymbol{K}$ and $\boldsymbol{V}$ by using the linear projection $\boldsymbol{W}_k$ and $\boldsymbol{W}_v$, where $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{d\times d}$ are trainable parameter matrices. Then we can calculate the attention weight matrix $\boldsymbol{A} \in \mathbb{R}^{9\times K}$ by:

$$\boldsymbol{A} = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}), \tag{2}$$

where $\boldsymbol{A}(i,j)$ (the element of $\boldsymbol{A}$ in the i-th row and j-th column) means the importance weight of the j-th post to i-th symptom, and softmax denotes the row-wise softmax operation. After obtaining $\boldsymbol{A}$, the symptom capsules of user x can be computed as:

$$\boldsymbol{U} = \boldsymbol{A}\boldsymbol{V}, \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{9\times d}$ is the user symptom representation matrix, and each row of $\boldsymbol{U}$ represents one of the symptom features of the user.

In order to guarantee that different posts are attentive to different symptom features, we add the orthogonal constraint term to the attention matrix $\boldsymbol{A}$. Specifically,

$$\mathcal{L}_{sym} = ||\boldsymbol{A}\boldsymbol{A}^T - \boldsymbol{I}||_F^2, \tag{4}$$

where $\boldsymbol{I}$ is an identity matrix, $||\cdot||_F$ is the Frobenius norm of a matrix. By minimizing $\mathcal{L}_{sym}$, we can ensure the diversity of the symptom capsules, i.e., different symptom capsules tend to be generated by different posts of each user.

Depression Capsules After extracting symptom features for each user, we can distill these features to obtain the higher-level representations, which can be used to infer whether the user is depressed. Specifically, the distilling procedure exploits an unsupervised routing-by-agreement mechanism to automatically select the appropriate symptom features to construct the final output capsules. Given the user x with the user symptom representation matrix $\boldsymbol{U} = [\boldsymbol{u}_1; \boldsymbol{u}_2; ...; \boldsymbol{u}_9] \in \mathbb{R}^{9\times d}$, we first transform each symptom feature $\boldsymbol{u}_i \in \mathbb{R}^{1\times d}$ (the i-th row of $\boldsymbol{U}$) to a prediction vector associated with each class:

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{u}_i \boldsymbol{W}_{ij}, \tag{5}$$

where $\hat{\boldsymbol{u}}_{j|i} \in \mathbb{R}^{1\times d'}$ is the prediction vector of the i-th symptom feature pertaining to the j-th class. $\boldsymbol{W}_{ij} \in \mathbb{R}^{d\times d'}$ is the corresponding transformation matrix, where $i \in \{1, 2, ..., 9\}$ represents 9 different symptom features, $j \in \{0, 1\}$ represents 2 classes (non-depressed and depressed users), and d' is the dimension of the prediction vector.

In terms of the depression detection task, only two output (depression) capsules are required, which are corresponding to the labels of depressed and non-depressed users. For the j-th depression capsule $\boldsymbol{h}_j$, it can be calculated by the weighted sum over all prediction vectors:

$$\boldsymbol{h}_j = \sum_{i=1}^{9} c_{ij}\hat{\boldsymbol{u}}_{j|i}, \tag{6}$$

where c_{ij} is the coupling coefficient which means the contribution degrees of the i-th symptom feature to the j-th class. And it can be determined by the unsupervised dynamic routing algorithm (Sabour, Frosst, and Hinton 2017). More details can refer to Algorithm 1, and $\hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$ denotes the inner product of the two vectors

To compress the length of the depression capsule between 0 and 1, we apply the squashing function on $\boldsymbol{h}_j$, and then obtain the final output $\boldsymbol{v}_j$ associated with the j-th class. Formally,

$$\boldsymbol{v}_j = \text{squash}(\boldsymbol{h}_j) = \frac{\|\boldsymbol{h}_j\|^2}{1 + \|\boldsymbol{h}_j\|^2} \frac{\boldsymbol{h}_j}{\|\boldsymbol{h}_j\|}, \tag{7}$$

where the length of the depression capsule $\boldsymbol{v}_j$ means the probability of the existence of the j-th class.

Motivated by (Sabour, Frosst, and Hinton 2017), We use the max-margin loss function to conduct the training procedure. The loss function $\mathcal{L}_{dep}$ can be written as:

$$\begin{aligned}\mathcal{L}_{dep} = \sum_{j=0}^{1} \{&\mathbb{I}(y = y_j)\max(0, m^+ - \|\boldsymbol{v}_j\|)^2 \\ &+ \lambda\mathbb{I}(y \neq y_j)\max(0, \|\boldsymbol{v}_j\| - m^-)^2\},\end{aligned} \tag{8}$$

where $\mathbb{I}(\cdot)$ is an indicator function. If the condition $(\cdot)$ is satisfied, the return value is 1, otherwise 0. y is the ground truth class label of x. m^+ and m^- are the margins. λ is a down-weighting coefficient.

Contrastive Learning

Contrastive learning has achieved promising success in computer vision (Chen et al. 2021; Wang et al. 2021; Chen et al. 2022), which aims to maximize similarities between instances from the same class and minimize similarities between instances from different classes. Here we integrate two kinds of contrastive learning, i.e., user-level and post-level contrastive learning, to acquire enhanced intermediate representations.

User-Level Contrastive Learning We construct the users from the same category as positive samples, and the users from different categories as negative samples. Assume that there are B users in a batch, given a user x_i with the user symptom representation matrix $\boldsymbol{U}_i \in \mathbb{R}^{9\times d}$, we first flatten

the matrix into a vector and normalize it to obtain $\boldsymbol{z}_i \in \mathbb{R}^{9d}$. Then, we define the set $I = \{1, 2, ..., B\}$, and $I \backslash i$ means the elements in I with i excluded. Considering $\boldsymbol{z}_i$ as an anchor, we can obtain the positive sample number set $\Gamma(i) = \{j \in I \backslash i | y_i = y_j\}$, and the negative sample numbers are the remaining ones in $I \backslash i$. Furthermore, for one batch, we can have the user-level contrastive learning loss function as follows:

$$\mathcal{L}_{uc} = \frac{1}{B} \sum_{i \in I} -\frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \log \frac{\exp(\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_j}{\tau_u})}{\sum_{k \in I \backslash i} \exp(\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_k}{\tau_u})}, \quad (9)$$

where $\boldsymbol{z}_i \cdot \boldsymbol{z}_j$ denotes the inner product of the two vectors, $|\Gamma(i)|$ is the number of samples in $\Gamma(i)$, and τ_u is a temperature hyperparameter.

To analyze Eq. (9), we do some simple formula manipulation as below.

$$\begin{aligned} \mathcal{L}_{uc} &= \frac{1}{B} \sum_{i \in I} -\frac{1}{|\Gamma(i)|} \mathcal{L}', \\ \mathcal{L}' &= \sum_{j \in \Gamma(i)} \log \frac{\exp(\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_j}{\tau_u})}{\sum_{k \in I \backslash i} \exp(\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_k}{\tau_u})} \\ &= \sum_{j \in \Gamma(i)} (\underbrace{\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_j}{\tau_u}}_{positive} - \log \underbrace{\sum_{k \in I \backslash i} \exp(\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_k}{\tau_u})}_{positive+negative}). \end{aligned} \quad (10)$$

According to the above formula, It is easy to discover that if we want to minimize $\mathcal{L}_{uc}$, we must maximize $\mathcal{L}'$, which requires us to maximize the positive term and minimize the sum of the positive and negative terms, thereby decreasing the negative term.

Post-Level Contrastive Learning For post-level contrastive learning, we treat the posts from the same types of users as the positive samples, and the posts from the different types of users as the negative samples. Assume that there are B users in a batch size, then $B \times K$ risky posts are in this batch. We define the set $M = \{1, 2, ..., B \times K\}$, and $M \backslash i$ means the elements in M with i excluded. Considering the embedding of p_i associated with the user type y_i, we can get the positive sample number set $\Phi(i) = \{j \in M \backslash i | y_i = y_j\}$, and the negative sample numbers are the remaining ones in $M \backslash i$. for one batch, we can have the post-level contrastive learning loss function as follows:

$$\mathcal{L}_{pc} = \frac{1}{B} \sum_{i \in M} -\frac{1}{|\Phi(i)|} \sum_{j \in \Phi(i)} \log \frac{\exp(\frac{\boldsymbol{p}_i \cdot \boldsymbol{p}_j}{\tau_p})}{\sum_{l \in M \backslash i} \exp(\frac{\boldsymbol{p}_i \cdot \boldsymbol{p}_l}{\tau_p})}, \quad (11)$$

where $\boldsymbol{p}_i \cdot \boldsymbol{p}_j$ denotes the inner product of the two vectors, $\boldsymbol{p}_i$ is the post embeddings and τ_p is a hyperparameter.

The Overall Loss Function By combining Eqs. (4), (8), (9) and (11), we have the overall loss function of the proposed method.

$$\mathcal{L}_{total} = \mathcal{L}_{dep} + \alpha \mathcal{L}_{sym} + \beta \mathcal{L}_{uc} + \gamma \mathcal{L}_{pc}, \quad (12)$$

where α, β, γ are the hyperparameters. By minimizing the loss $\mathcal{L}_{total}$ with the gradient descent method, all trainable parameters can be learned.

Dataset	eRisk2018	RSDD	TRT
No. of users	1707	117203	12447
No. of non-depressed users	1493	107995	6873
No. of depressed users	214	9208	5574
Avg. posts of per user	600.8	942.3	1220.9
Avg. words of per post	40.0	33.2	33.1

Table 3: Dataset statistics.

Dataset	K	α	β	γ	λ	m^+	m^-
eRisk2018	16	0.1	0.3	0.5			
TRT	32	0.2	0.6	0.5	0.5	0.95	0.05
RSDD	64	0.2	0.7	0.5			

Table 4: Hyperparameters of our proposed method.

Experiments

Datasets

We follow (Nguyen et al. 2022) to conduct experiments on three widely used depression detection datasets: eRisk2018 (Losada and Crestani 2016), RSDD (Yates, Cohan, and Goharian 2017) and TRT (Wolohan et al. 2018). As the TRT dataset used in (Nguyen et al. 2022) is not released publicly, we follow (Wolohan et al. 2018) to reconstruct this dataset. For data split, we also follow (Nguyen et al. 2022) to separate the datasets eRisk2018 and TRT into the training, validation, test set with the ratio 8:1:1, and separate the dataset RSDD with the ratio 1:1:1. The detailed dataset statistics are shown in Table 3.

Baselines

We compare our method with the following strong baselines.

- **BERT-CNN** only uses a BERT (Devlin et al. 2019) encoder and a CNN classifier to fulfill the task.
- **Pattern** (Nguyen et al. 2022) designs patterns for each symptom and matches them with posts to create a matching matrix. **Pattern (threshold)** regards users with the number of matches in the matrix exceeding a certain threshold as depressed users. **Pattern (CNN)** utilizes this matrix with a CNN classifier for classification.
- **PHQ9** (Nguyen et al. 2022) uses the weakly-labeled data to train a classifier for each symptom. **PHQ9 (scores)** and **PHQ9 (vectors)** uses the scores and embeddings obtained previously as the input respectively, and then classify them through a CNN classifier. **PHQ9plus** extends the PHQ9 method by appending an additional symptom to the PHQ9 symptoms.

Implementation Details

Evaluation Metrics We follow (Nguyen et al. 2022) to use the F1 score and Area Under Curve (AUC) to evaluate the performance.

Training	Method	Test: eRisk2018		Test: RSDD		Test: TRT	
		F1	AUC	F1	AUC	F1	AUC
eRisk2018	Pattern (threshold)	0.40 ± 0.00	-	0.32 ± 0.00	-	0.43 ± 0.00	-
	Pattern (CNN)	0.43 ± 0.01	0.80 ± 0.00	0.31 ± 0.01	0.73 ± 0.01	0.42 ± 0.01	0.66 ± 0.00
	PHQ9 (scores)	0.54 ± 0.02	0.87 ± 0.00	0.38 ± 0.00	0.81 ± 0.01	0.45 ± 0.01	0.68 ± 0.00
	PHQ9 (vectors)	0.55 ± 0.00	0.88 ± 0.00	$\mathbf{0.39 \pm 0.01}$	$\mathbf{0.82 \pm 0.00}$	0.46 ± 0.01	0.67 ± 0.01
	PHQ9plus	$\mathbf{0.73 \pm 0.03}$	0.94 ± 0.00	0.35 ± 0.01	0.79 ± 0.01	$\mathbf{0.47 \pm 0.02}$	0.69 ± 0.00
	BERT-CNN	0.71 ± 0.03	$\mathbf{0.95 \pm 0.01}$	0.36 ± 0.02	0.81 ± 0.02	0.46 ± 0.00	$\mathbf{0.70 \pm 0.02}$
	DeCapsNet	$\mathbf{0.79 \pm 0.02}$	$\mathbf{0.95 \pm 0.01}$	$\mathbf{0.40 \pm 0.03}$	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.49 \pm 0.02}$	$\mathbf{0.73 \pm 0.01}$
RSDD	Pattern (threshold)	0.38 ± 0.00	-	0.35 ± 0.00	-	0.45 ± 0.00	-
	Pattern (CNN)	0.47 ± 0.00	0.79 ± 0.01	0.36 ± 0.02	0.74 ± 0.01	0.44 ± 0.02	0.68 ± 0.01
	PHQ9 (scores)	0.43 ± 0.01	0.80 ± 0.01	0.47 ± 0.01	0.85 ± 0.00	0.47 ± 0.00	0.69 ± 0.00
	PHQ9 (vectors)	0.46 ± 0.01	0.81 ± 0.01	0.49 ± 0.01	0.85 ± 0.00	0.52 ± 0.01	0.69 ± 0.02
	PHQ9plus	$\mathbf{0.49 \pm 0.00}$	0.81 ± 0.03	$\mathbf{0.55 \pm 0.00}$	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.55 \pm 0.00}$	$\mathbf{0.70 \pm 0.00}$
	BERT-CNN	0.44 ± 0.02	$\mathbf{0.84 \pm 0.01}$	0.53 ± 0.01	$\mathbf{0.86 \pm 0.00}$	0.52 ± 0.01	0.69 ± 0.01
	DeCapsNet	$\mathbf{0.57 \pm 0.02}$	$\mathbf{0.92 \pm 0.01}$	$\mathbf{0.64 \pm 0.01}$	$\mathbf{0.92 \pm 0.00}$	$\mathbf{0.59 \pm 0.01}$	$\mathbf{0.71 \pm 0.01}$
TRT	Pattern (threshold)	0.39 ± 0.00	-	0.31 ± 0.00	-	0.51 ± 0.00	-
	Pattern (CNN)	0.40 ± 0.00	0.79 ± 0.00	0.34 ± 0.01	0.72 ± 0.00	0.58 ± 0.00	0.70 ± 0.00
	PHQ9 (scores)	0.41 ± 0.02	0.81 ± 0.03	0.37 ± 0.00	0.78 ± 0.01	0.62 ± 0.01	0.74 ± 0.01
	PHQ9 (vectors)	0.39 ± 0.01	$\mathbf{0.82 \pm 0.01}$	$\mathbf{0.38 \pm 0.01}$	$\mathbf{0.80 \pm 0.01}$	0.68 ± 0.00	0.78 ± 0.00
	PHQ9plus	$\mathbf{0.42 \pm 0.01}$	0.80 ± 0.01	0.36 ± 0.01	0.79 ± 0.01	$\mathbf{0.72 \pm 0.02}$	0.80 ± 0.01
	BERT-CNN	0.34 ± 0.00	$\mathbf{0.82 \pm 0.01}$	0.27 ± 0.01	0.78 ± 0.01	0.70 ± 0.01	$\mathbf{0.82 \pm 0.00}$
	DeCapsNet	$\mathbf{0.45 \pm 0.01}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.39 \pm 0.04}$	$\mathbf{0.80 \pm 0.02}$	$\mathbf{0.77 \pm 0.01}$	$\mathbf{0.87 \pm 0.00}$

Table 5: Experimental results on eRisk2018, RSDD and TRT three datasets.

Method	eRisk2018		RSDD		TRT	
	F1	AUC	F1	AUC	F1	AUC
DeCapsNet	0.79 ± 0.02	0.95 ± 0.01	0.64 ± 0.01	0.92 ± 0.00	0.77 ± 0.01	0.87 ± 0.00
DeCapsNet w/o $\mathcal{L}_{sym}$	0.78 ± 0.07	0.94 ± 0.02	0.62 ± 0.04	0.92 ± 0.00	0.75 ± 0.01	0.86 ± 0.00
DeCapsNet w/o $\mathcal{L}_{uc}$	0.74 ± 0.02	0.93 ± 0.02	0.58 ± 0.01	0.92 ± 0.02	0.72 ± 0.00	0.80 ± 0.01
DeCapsNet w/o $\mathcal{L}_{pc}$	0.76 ± 0.02	0.93 ± 0.01	0.59 ± 0.01	0.92 ± 0.01	0.73 ± 0.00	0.82 ± 0.01

Table 6: Ablation study on eRisk2018, RSDD and TRT three datasets.

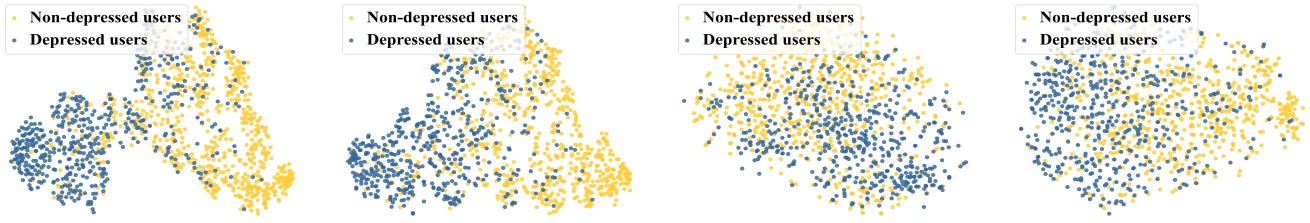
Parameter Settings The dimension of each symptom capsule $d = 768$, and the dimension of each depression capsule $d' = 100$. The total iteration number of dynamic routing is 3. We use AdamW optimizer with the initialized learning rate 2e-5. For the contrastive learning, we set τ_u and τ_p as 0.01 consistently. All the hyperparameters are selected based on the performance of the validation set. More parameter details are listed in Table 4.

Experimental Results

We follow (Nguyen et al. 2022) to conduct experiments in two scenarios: the within-dataset and cross-dataset scenarios. In the within-dataset setting, we evaluate the method in the test set of the original dataset. In the cross-dataset setting, we evaluate the method in the test set of the other dataset, which can verify the generalization ability of different methods. All reported results are averaged over 5 different runs. Some baseline results are taken from (Nguyen et al. 2022), and the top 2 results are highlighted in bold.

Comparison in Within-Dataset Setting From Table 5, it can be seen that in the within-dataset scenario, our method performs much better than other baselines. Specifically, in terms of F1 score, our method improves upon the most competitive baseline PHQ9plus by 6%, 9% and 5% on eRisk2018, RSDD, and TRT respectively. In terms of AUC, our method improves upon the most competitive baseline BERT-CNN by 6% and 5% on RSDD and TRT respectively. The reason is that our method utilizes an end-to-end training strategy, thus leveraging the supervised information sufficiently. In addition, integrating the contrastive learning can lead the learned embeddings to be more suitable for the classification task.

Comparison in Cross-Dataset Setting From Table 5, we can observe that in the cross-dataset scenario, our method outperforms other baselines significantly. From RSDD to eRisk2018, compared with the second best method, our method can achieve up to 8% improvement in both F1 score and AUC metrics. This indicates that our method has a strong generalization ability, and can be extended to various



(a) DeCapsNet (b) DeCapsNet w/o $\mathcal{L}_{sym}$ (c) DeCapsNet w/o $\mathcal{L}_{uc}$ (d) DeCapsNet w/o $\mathcal{L}_{pc}$

Figure 2: Visualization of user embeddings in the test set of the TRT dataset.

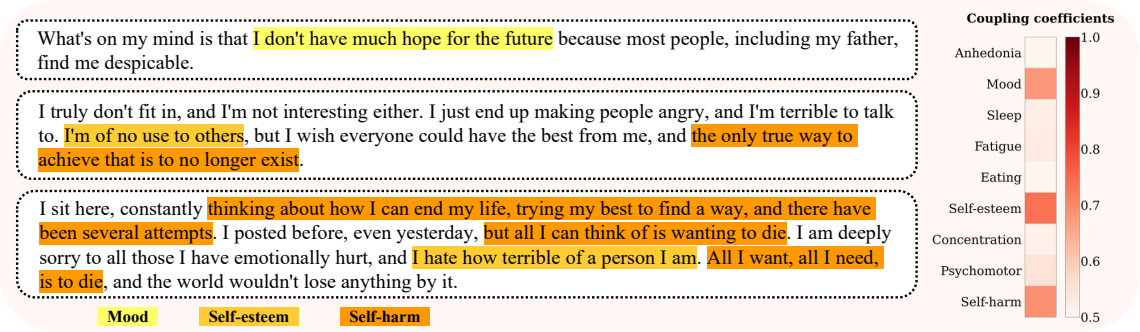


Figure 3: A concrete depressed user example from the eRisk2018 dataset.

depression detection applications.

Ablation Study

To verify the effectiveness of different parts of the loss function, we make the ablation study. The results are shown in Table 6. DeCapsNet w/o $\mathcal{L}_{sym}$, DeCapsNet w/o $\mathcal{L}_{uc}$ and DeCapsNet w/o $\mathcal{L}_{pc}$ mean the DeCapsNet model without the terms $\mathcal{L}_{sym}$, $\mathcal{L}_{uc}$ and $\mathcal{L}_{pc}$ respectively. We can find that DeCapsNet always performs much better than other cases. The reason is that the $\mathcal{L}_{sym}$ term can guarantee the diversity of the symptom capsules, and the $\mathcal{L}_{uc}$ and $\mathcal{L}_{pc}$ terms can push samples in the same class close and pull samples in different classes apart, thus are more conducive to classification.

Figure 2 visualizes the embedding distribution of the users in the test set of the TRT dataset. Specifically, we flatten the symptom capsules as the user embeddings and use t-SNE (Van der Maaten and Hinton 2008) to achieve the visualization. It also can be observed that in Figure 2 (a) the samples belonging to the same class are more compact than other cases, thus further illustrating that different modules all contribute to the model to some extent.

Case Study

To further explain the interpretability of our method, we provide a concrete depressed user example from the eRisk2018 dataset, which is shown in Figure 3. On the left side of the figure, we pick up three representative posts of the user. It is easy to find that these posts contain three obvious symptoms about Mood, Self-esteem and Self-harm. On the right side of the figure, we visualize the coupling coefficients which represent the contribution degrees of each symptom capsule to the predicted class label, where its label is depressed. We can observe that the coefficients associated with Mood, Self-esteem and Self-harm symptom capsules are significantly greater than others, which is consistent with the original posts. Through this example, it can reveal that our proposed method is an effective and interpretable hierarchical reasoning framework for modeling user posts to symptom capsules and distilling appropriate symptom capsules for the depression detection task.

Conclusion

In this paper, we propose a novel hierarchical capsule network integrated with contrastive learning for depression detection (DeCapsNet). By utilizing elaborately designed depression symptom descriptions and the attention mechanism to assign appropriate importance weight for each post, DeCapsNet can extract interpretable symptom features. By leveraging unsupervised dynamic routing to distill symptom capsules to depression capsules, DeCapsNet can fulfill the depression detection task accurately. In addition, DeCapsNet integrates with the contrastive learning to reduce intra-class discrepancy and enlarge the inter-class difference from both user level and post level, thus obtaining more class-indicative representations. Extensive experiments show that in both within-dataset and cross-dataset scenarios DeCapsNet can always achieve impressive performance on three widely used datasets eRisk2018, RSDD and TRT, even in some cases outperform other strong baselines by a large margin. In future work, we plan to deploy DeCapsNet in the online environment and extend the model for other psychological illness diagnosis.

Acknowledgments

The authors are grateful to the reviewers for their valuable comments. This work was supported by National Natural Science Foundation of China (No. 62106035, 62206038, 61972065) and Fundamental Research Funds for the Central Universities (No. DUT20RC(3)040, DUT20RC(3)066), and supported in part by Key Research Project of Zhejiang Lab (No. 2022PI0AC01), National Key Research and Development Program of China (2022YFB4500300). We also would like to thank Dalian Ascend AI Computing Center and Dalian Ascend AI Ecosystem Innovation Center for providing inclusive computing power and technical support.

References

Chen, H.; Wang, Y.; Lagadec, B.; Dantcheva, A.; and Brémond, F. 2021. Joint Generative and Contrastive Learning for Unsupervised Person Re-Identification. In *Computer Vision and Pattern Recognition (CVPR)*, 2004–2013.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, 1597–1607.

Chen, X.; Pan, J.; Jiang, K.; Li, Y.; Huang, Y.; Kong, C.; Dai, L.; and Fan, Z. 2022. Unpaired Deep Image Deraining Using Dual Contrastive Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007–2016.

Cho, S.; Lebanoff, L.; Foroosh, H.; and Liu, F. 2019. Improving the Similarity Measure of Determinantal Point Processes for Extractive Multi-Document Summarization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 1027–1038.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 4171–4186.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6894–6910.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Computer Vision and Pattern Recognition (CVPR)*, 9726–9735.

Hinton, G. E.; Sabour, S.; and Frosst, N. 2018. Matrix capsules with EM routing. In *International Conference on Learning Representations (ICLR)*.

Hu, P.; Lin, C.; Su, H.; Li, S.; Han, X.; Zhang, Y.; and Mei, J. 2020. BlueMemo: Depression Analysis through Twitter Posts. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 5252–5254.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Kroenke, K.; Spitzer, R. L.; and Williams, J. B. 2001. The PHQ-9: validity of a brief depression severity measure. *Journal of General Internal Medicine*.

Liu, H.; Zhang, X.; Fan, L.; Fu, X.; Li, Q.; Wu, X.; and Lam, A. Y. S. 2019. Reconstructing Capsule Networks for Zero-shot Intent Classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4798–4808.

Losada, D. E.; and Crestani, F. 2016. A Test Collection for Research on Depression and Language Use. In *Conference and Labs of the Evaluation Forum (CLEF)*, 28–39.

Nguyen, T.; Yates, A.; Zirikly, A.; Desmet, B.; and Cohan, A. 2022. Improving the Generalizability of Depression Detection by Leveraging Clinical Questionnaires. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 8446–8459.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3980–3990.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic Routing Between Capsules. In *Conference on Neural Information Processing Systems (NeurIPS)*, 3856–3866.

Tausczik, Y. R.; and Pennebaker, J. W. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11): 2579–2605.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 5998–6008.

Wang, P.; Han, K.; Wei, X.; Zhang, L.; and Wang, L. 2021. Contrastive Learning Based Hybrid Networks for Long-Tailed Image Classification. In *Computer Vision and Pattern Recognition (CVPR)*, 943–952.

Wolohan, J.; Hiraga, M.; Mukherjee, A.; Sayyed, Z. A.; and Millard, M. 2018. Detecting linguistic traces of depression in topic-restricted text: Attending to self-stigmatized depression with NLP. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, 11–21.

Xia, C.; Zhang, C.; Yan, X.; Chang, Y.; and Yu, P. S. 2018. Zero-shot User Intent Detection via Capsule Neural Networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3090–3099.

Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; and Xu, W. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 5065–5075.

Yates, A.; Cohan, A.; and Goharian, N. 2017. Depression and Self-Harm Risk Assessment in Online Forums. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2968–2978.

Zhang, T.; Yang, K.; and Ananiadou, S. 2023. Sentiment-guided Transformer with Severity-aware Contrastive Learning for Depression Detection on Social Media. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 114–126.

Zhang, Z.; Chen, S.; Wu, M.; and Zhu, K. Q. 2022. Psychiatric Scale Guided Risky Post Screening for Early Detection of Depression. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 5220–5226.

Zhao, W.; Peng, H.; Eger, S.; Cambria, E.; and Yang, M. 2019. Towards Scalable and Reliable Capsule Networks for Challenging NLP Applications. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 1549–1559.