Contrastive Graph Transformer Network for Personality Detection

Yangfu Zhu[1†], **Linmei Hu**[1†*], **Xinkai Ge**[1], **Wanrong Peng**[2], **Bin Wu**[1*]
[1]Beijing Key Laboratory of Intelligence Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, China
[2]Medical Psychological Center, the Second Xiangya Hospital, Central South University, Changsha, China
{zhuyangfu, hulinmei,gexinkai2021,wubin}@bupt.edu.cn, wanrongpeng@csu.edu.cn

Abstract

Personality detection is to identify the personality traits underlying social media posts. Most of the existing work is mainly devoted to learning the representations of posts based on labeled data. Yet the ground-truth personality traits are collected through time-consuming questionnaires. Thus, one of the biggest limitations lies in the lack of training data for this data-hungry task. In addition, the correlations among traits should be considered since they are important psychological cues that could help collectively identify the traits. In this paper, we construct a fully-connected post graph for each user and develop a novel Contrastive Graph Transformer Network model (CGTN) which distills potential labels of the graphs based on both labeled and unlabeled data. Specifically, our model first explores a self-supervised Graph Neural Network (GNN) to learn the post embeddings. We design two types of post graph augmentations to incorporate different priors based on psycholinguistic knowledge of Linguistic Inquiry and Word Count (LIWC) and post semantics. Then, upon the post embeddings of the graph, a Transformer-based decoder equipped with post-to-trait attention is exploited to generate traits sequentially. Experiments on two standard datasets demonstrate that our CGTN outperforms the state-of-the-art methods for personality detection.

1 Introduction

Personality refers to the characteristic pattern in a person's thinking, feeling, and decision-making [Kaushal and Patwardhan, 2018]. Personality detection is an emerging topic in user profile research, which aims to identify one's personality traits from online texts he/she creates and has expanded to massive applications such as recommendation system [Shen *et al.*, 2020], dialogue system [Yang *et al.*, 2021b; Wen *et al.*, 2021] and computer game design [Lang *et al.*, 2019].

With the blossoming of social media, users yield considerable posts containing their mental activities every day, offering new possibilities for automatically inferring personality traits [Štajner and Yenikent, 2020]. Earlier researchers mainly used two sources of lexical features, Linguistic Inquiry and Word Count (LIWC) [Tausczik and Pennebaker, 2010] and Medical Research Council (MRC) [Coltheart, 1981] to identify personality from user-generated posts [Mairesse *et al.*, 2007]. To overcome manual feature engineering, deep neural networks (DNNs) were applied in the personality detection task to obtain the representations of posts. However, understanding the hidden personality traits behind the posts is non-trivial. Most recent works have been devoted to refining post representations from the perspective of the post structure including [Lynn *et al.*, 2020], [Yang *et al.*, 2021c],and [Yang *et al.*, 2021a]. Despite the considerable improvements achieved in personality detection, the existing models are likely to suffer from the scarcity of personality tags as the ground-truth personality traits are usually collected from professional questionnaires, which are often resource-intensive and time-consuming. Hence, such precious personality tags are hard to collect, which becomes a limitation for training deep neural networks and makes it difficult to infer personality from posts.

In addition, personality is defined in terms of different dimensions (traits) and these traits often co-occur with a non-negligible correlation, which has been confirmed in empirical psychological researches[John *et al.*, 2008; Sharpe *et al.*, 2011]. For example, neurotic people are more likely to be extroverted, like Trump. However, such implicit trait correlations are rarely exploited, which should have been the key psychological cues to be considered for personality detection.

Taking both the problem of data scarcity and the correlations among traits into consideration, we model the user-generated posts as a fully-connected post graph, and propose a novel Contrastive Graph Transformer Network model (CGTN) for personality detection, which distills potential labels of the graphs based on both labeled and unlabeled data. Specifically, CGTN consists of a contrastive post graph encoder and a trait sequence decoder. **In post graph encoder**, two types of graph augmentations are designed to incorporate different priors based on psycholinguistic knowledge of LIWC and post semantics. To be precise, LIWC can be utilized to extract psycholinguistic features while post semantics

[†]Equal contribution.
[*]Corresponding authors.

are able to capture the semantic relations among the posts. A self-supervised paradigm is defined to maximize the agreement over the representations of the augmented graphs that come from the same user. This contrastive strategy allows us to learn the post embeddings without using any labeled data. **In trait sequence decoder**, we view the multi-trait detection task as a trait sequence generation problem and apply a transformer-based decoder to model the correlations among traits. In addition, we use the post-to-trait attention to ensure that crucial posts are selected for trait generation.

In summary, our main contributions are as follows:

- To our best knowledge, this is the first effort to explore contrastive self-supervised learning to distill auxiliary signals for personality detection, providing a new perspective for alleviating the data scarcity for personality detection.
- We propose a novel Contrasive Graph Transformer Network (CGTN) model, for which we design two types of graph augmentations to incorporate priors based on LIWC, and post semantic knowledge, and explicitly introduce trait correlations by exploiting a sequence generation model.
- The experimental results demonstrate the outperformance of our model over the baselines including the state-of-the-art methods, which shows the effectiveness of our model.

2 Related Work

As an emerging interdisciplinary study, personality detection has attracted the attention of both computer scientists and psychologists [Xue *et al.*, 2018; Mehta *et al.*, 2020; Yang *et al.*, 2021c].

Earlier studies mainly exploit psychologically statistical features to detect personality [Mairesse *et al.*, 2007], such as LIWC [Tausczik and Pennebaker, 2010] and MRC [Coltheart, 1981]. Nonetheless, the statistical analysis cannot effectively represent the original semantics of the posts. With the rapid development of deep learning, a series of Deep Neural Networks (DNNs) are applied to personality detection task and have achieved great success, including CNN [Xue *et al.*, 2018], LSTM [Tandera *et al.*, 2017], etc. Recently, personality detection has benefited from large-scale pre-trained language models, such as BERT [Devlin *et al.*, 2018], and thus get improved[Mehta *et al.*, 2020; Ren *et al.*, 2021]. Based on these pre-trained models, latest works focus on refining post representations from the perspective of post structure. [Lynn *et al.*, 2020] designed SN+Attn which introduces a hierarchical attention network to obtain user document representations in a bottom-up manner from the word-grained level to the post-grained level, arguing that not every post contributes equally. In order to avoid introducing post-order bias, Transformer-MD [Yang *et al.*, 2021a] considers different posts to be unrelated to each other. TrigNet [Yang *et al.*, 2021c], however, holds a different view that there is a psycholinguistic structure between posts and constructs a heterogeneous graph for each user and aggregates post information from a psychological perspective.

However, the above methods mainly focus on obtaining the representations of the user's posts by the supervised paradigms. For personality detection task, human-provided labels are hard to collect, thus the model tends to overfit the training data and performs poorly on test data. In this work, to address the issue, we develop a novel contrastive graph transformer model for personality detection, which fully exploits both labeled and unlabeled data through contrastive self-supervised learning.

3 Preliminaries

Personality detection can be phrased as a multi-document multi-label classification task [Lynn *et al.*, 2020; Yang *et al.*, 2021a]. Formally, given a set $P = \{p_1, p_2...p_n\}$ of N posts from a user, where $p_i = \{w_i^1, w_i^2...w_i^k\}$ is i-th post with k tokens, our goal is to predict t-dimensional personality traits from the trait-specific label space $\mathrm{Y} = \{y_1, y_2, \cdots y_t\}$, e.g., t = 4 in the MBTI taxonomy, t = 5 in the Big-Five taxonomy. In this paper, we model a user-generated document as graph over posts. For each user with n posts, we construct a fully-connected original graph $G = (V, E)$, where V consists of n post nodes and the edges E capture the correlations among the posts. The BERT is employed to obtain the initial embeddings of each post node. And then based on the post graph, we propose a Contrastive Graph Transformer Network model (CGTN) for personality detection.

4 Contrastive Graph Transformer Network

Figure 1 presents the overall architecture of the proposed CGTN, which consists of a contrastive post graph encoder and a trait sequence decoder. The encoder aims to learn rich post representations via self-discrimination on post graph while the decoder is to uncover psychological cues contained in the personality correlations. In the following subsections, we detail the contrastive post graph encoder and trait sequence decoder.

4.1 Contrastive Post Graph Encoder

In contrastive post graph encoder, we design two types of graph augmentations based on psycholinguistic knowledge of LIWC and post semantics. Thereafter, contrastive self-supervised learning is exploited on augmentations graph to learn post representation by judging whether two augmented graphs are from the same user.

Post Graph Augmentation

The core of personality detection is to understand a collection of user-generated posts. Previous works demonstrated that digging the inherent patterns in the structure of post is helpful for representation. Self-supervised learning allows us to exploit the "unlabeled" data via making disturbs on the input data. Naturally, we can construct the "unlabeled" data by generating multi-view post graphs for each user. Specifically, LIWC is used to construct **psycholinguistic view graphs** G^{α} [Yang *et al.*, 2021c]. The LIWC dictionary divides words into psychology-related categories $C = \{c_1, c_2 \cdots, c_n\}$ which can be taken as bridges to connect different post nodes. Two post nodes are connected if they contain the words of the same

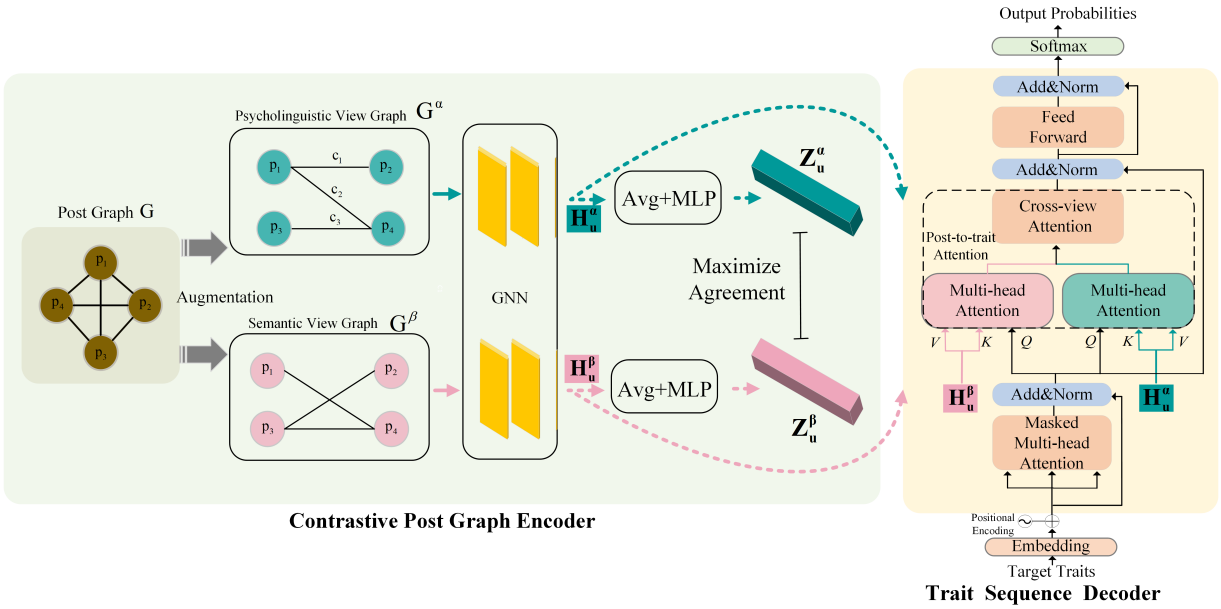


Figure 1: An overview of our CGTN, which consists of a contrastive post graph encoder and a trait sequence decoder.

categories. For the **semantic view graphs** G^{β}, We build the edges between the posts if their semantic similarity is larger than a given threshold. The semantic similarity is computed as the cosine similarity based on the initial post embeddings.

Graph Contrastive Self-supervised Learning

Contrastive self-supervised learning offers a simple way to learn invariant representations by local disturbs in the input data without using any labeled data. In our task, we randomly sample a batch of U users and any pair of augmented graphs (G^{α}, G^{β}) that comes from the same user is considered as a positive pair. Otherwise, they are labeled as negative. We learn to predict whether two augmented graphs originate from the same user or not. In the following, we first introduce how we learn the representation of a graph and then illustrate the contrastive loss.

Specifically, to obtain the graph representation, we first use GNN to capture the structural information within nodes' neighborhoods [Xu *et al.*, 2018]. The L-th layer GNN updates the post node embeddings h_p as:

$$\boldsymbol{h}_{\boldsymbol{p}}^{\boldsymbol{L}} = \mathrm{GNN}(\boldsymbol{h}_{\boldsymbol{p'}}^{\boldsymbol{L-1}}), \tag{1}$$

where p' is the neighbour node of post node p on the given augmented graph. where h_p^L is the embedding of the node p at the L layer. After obtaining the post node embeddings with fused neighbor information, we pass them through an average pooling layer and a two-layer MLP to obtain the entire graph representation. Formlly,

$$\boldsymbol{z}_{\boldsymbol{u}} = \mathrm{MLP}(Avg(\boldsymbol{h}_{\boldsymbol{p}}^{\boldsymbol{L}})), u \in U. \tag{2}$$

Based on the above graph embedding, the psycholinguistic augmentation graph and the semantic augmentation graph of user u are represented as $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}$ and $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\beta}}$, respectively. Given a positive pair $(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\beta}})$ and a negative pair $(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{v}}^{\boldsymbol{\beta}})$, which is sampled from the augmented graphs of other users v within the same batch. The contrastive loss L_{cl} is defined to maximize the consistency between positive pairs compared with negative pairs:

$$L_{cl} = \sum_{u \in U} -\log \frac{\exp(sim(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\beta}})/\tau)}{\sum_{v \in U} \exp(sim(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}, \boldsymbol{z}_{\boldsymbol{v}}^{\boldsymbol{\beta}})/\tau)}, \tag{3}$$

where $sim()$ denotes the cosine similarity and τ is a temperature hyperparameter.

4.2 Trait Sequence Decoder

Unlike single-trait classification where only one label is assigned to each sample, a decoder with Transformer [Vaswani *et al.*, 2017] backbones is designed to capture the correlations of traits by the sequence generation architecture. In addition we design post-to-trait attention to select the key posts for trait generation. Formally, the trait generation can be modeled as finding an optimal trait sequence y^* that maximizes the conditional probability:

$$P(y|\boldsymbol{H}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}, \boldsymbol{H}_{\boldsymbol{u}}^{\boldsymbol{\beta}}) = \prod_{t=1}^{T} p(y_t|y_1, y_2, \cdots, y_{t-1}; \boldsymbol{H}_{\boldsymbol{u}}^{\boldsymbol{\alpha}}, \boldsymbol{H}_{\boldsymbol{u}}^{\boldsymbol{\beta}}), \tag{4}$$

where $\boldsymbol{H}_{\boldsymbol{u}}^{\boldsymbol{\alpha}} = [\boldsymbol{h}_{\boldsymbol{p_1}}^{\boldsymbol{\alpha}}, \boldsymbol{h}_{\boldsymbol{p_2}}^{\boldsymbol{\alpha}}, \cdots, \boldsymbol{h}_{\boldsymbol{p_n}}^{\boldsymbol{\alpha}}]$ is post sequence based on psycholinguistic view graph G^{α}, similarly, $\boldsymbol{H}_{\boldsymbol{u}}^{\boldsymbol{\beta}}$ is post sequence based on semantic view graph G^{β}.

The decoder as shown in the right part of Figure 1 is composed of M identical blocks, where each block contains a multi-head self-attention layer, a post-to-trait attention layer and a feed-forward layer. Formally, the output of the first sub-layer $\boldsymbol{C}^{\boldsymbol{m}}$, the second sub-layer $\boldsymbol{D}^{\boldsymbol{m}}$, and the third sub-layer $\boldsymbol{E}^{\boldsymbol{m}}$ at m-th decoding block are sequentially calculated as:

$$\boldsymbol{C}^{\boldsymbol{m}} = \mathrm{LN}(\mathrm{SATT}(\boldsymbol{E}^{\boldsymbol{m-1}}) + \boldsymbol{E}^{\boldsymbol{m-1}}), \tag{5}$$

$$\boldsymbol{D^m} = \text{LN}((\text{PTATT}(\boldsymbol{C^m}, \boldsymbol{H_u}) + \boldsymbol{C^m})), \tag{6}$$

$$\boldsymbol{E^m} = \text{LN}(\text{FFN}(\boldsymbol{D^m}) + \boldsymbol{D^m}), \tag{7}$$

where LN($\cdot$) denotes layer normalization, SATT($\cdot$) denotes multi-head self-attention mechanism, PTATT($\cdot$) is post-to-trait attention layer we inserted, and FFN($\cdot$) is feed-forward network, $\mathbf{H_u} = \{\mathbf{H}_u^\alpha, \mathbf{H}_u^\beta\}$ denotes two post sequences, respectively.

Post-to-trait Attention

We design post-to-trait attention sub-layer to select crucial posts from the two augmented views for generating traits. This inserted sub-layer includes two steps: first, two view-specific post sequences $\mathbf{H}_u^\alpha$ and $\mathbf{H}_u^\beta$ are fed into the decoding module simultaneously. For each decoding step, the decoder processes each view independently and obtains two contextual sequences $(\mathbf{C}^{\mathbf{m}}_{\mathbf{p}\to\mathbf{t}})^\alpha$ and $(\mathbf{C}^{\mathbf{m}}_{\mathbf{p}\to\mathbf{t}})^\beta$. Formally:

$$\mathbf{C}^{\mathbf{m}}_{\mathbf{p}\to\mathbf{t}} = \text{ATT}(\mathbf{C}^{\mathbf{m}}, \mathbf{H_u}), \tag{8}$$

Subsequently, we leverage cross-view self-attention over two sequences to control different contributions of different views at each step. Formally:

$$\text{PTATT}(\cdot) = \text{SATT}((\mathbf{C}^{\mathbf{m}}_{\mathbf{p}\to\mathbf{t}})^\alpha, (\mathbf{C}^{\mathbf{m}}_{\mathbf{p}\to\mathbf{t}})^\beta), \tag{9}$$

Trait Generation

Finally, the output of the last layer of the decoder $\boldsymbol{E^m}$ is used to detect the personality via linear and softmax layer. The generation of the t-th trait by the decoder can be formalized as

$$\widehat{y}_t = \text{softmax}(W\boldsymbol{E^m} + I_t), \tag{10}$$

where I_t is the mask vector at decoding step t that is used to prevent the decoder from detecting the repeated trait. In inferring stage, $\widehat{y}_t$ is further used as input token of the next generation step to detect the $(t+1)$-th trait:

$$(I_t)_{t'} = \begin{cases} -\infty & \text{if the } t'\text{-th traits has been detected} \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

Following SGM [Yang *et al.*, 2018], we use beam search to find the top-ranked prediction path at generation time. The final output is trained using the mean binary cross-entropy over all traits. Given true binary label vector y_t and predicted labels $\widehat{y}_t$, the detection loss is:

$$L_{det} = -\sum_{u}^{U}\sum_{t=1}^{Y}(y_t \log(\hat{y}_t) + (1-y_t)\log(1-\hat{y}_t)). \tag{12}$$

4.3 Model Training

We apply two training strategies including pre-training and joint learning. For the **pre-training strategy**, the model is trained in a two-stage paradigm. Given a collection of unlabeled post graphs, a direct contrastive method is to predict whether two augmented graphs are similar. After training, we finetune the pre-trained graph embeddings in the downstream trait generation task. For the **joint learning strategy**, an auxiliary self-supervised task is included to help learn the supervised detection task, and two tasks share the same graph encoder. Our training objective is to minimize the cross-entropy loss and contrastive loss corresponding to the tasks of personality detection and post graph contrastive self-supervised learning, respectively. Formally, the objective function is defined as follows:

$$L = L_{det} + \lambda L_{cl}. \tag{13}$$

where λ is a trade-off parameter to control the strengths of contrastive learning L_{cl}.

5 Experiments

5.1 Dataset

Following previous studies, we conduct experiments on the Kaggle[1] with MBTI taxnomy and Essays datasets with Big-Five taxonomy. The **Kaggle** dataset is collected from PersonalityCafe, where people share their personality types and daily communications, with a total of 8675 users and 45-50 posts for each user. The traits for Kaggle dataset, namely, MBTI taxonomy, include Introversion / Extroversion, Sensing / Ntuition, Think / Feeling, and Perception / Judging. The **Essays** [Pennebaker and King, 1999] is a well-known dataset of stream-of-consciousness texts which contains 2468 anonymous users with approximately 50 sentences recorded for each user. Each user is tagged with a binary label of the Big Five taxonomy, including Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism. Two datasets are randomly divided into 6:2:2 for training, validation, and testing, respectively. The F1 metric is adopted to evaluate in the Essays dataset. The Macro-F1 is adopted to evaluate the performance in each personality trait since the Kaggle dataset is imbalanced. Note that, due to the privacy and high expenses for data collection, available personality datasets with standard labels are rare. In 2018, the MyPersonality dataset[2] stopped sharing as the world's largest personality dataset due to privacy breach.

5.2 Baselines

We compare our model with several baselines, which can be categorized as follows.

- **BiLSTM** [Tandera *et al.*, 2017] is a sequence model firstly employed to encode each post, and then the averaged post representation is used for user representation.
- **AttRCNN** [Xue *et al.*, 2018] is a hierarchical structure, in which CNN-based aggregator is employed to obtain the user representations.
- **BERT** is a pre-trained language model, [Mehta *et al.*, 2020; Ren *et al.*, 2021] perform extensive experiments to arrive at the optimal configuration for personality detection.
- **SN+Attn** [Lynn *et al.*, 2020] is a hierarchical network, in which the GRU with attention is used to encode both sequences of words and posts for user representations.

[1] kaggle.com/datasnaek/mbti-type

[2] http://mypersonality.org./

Methods	Kaggle						Essays						
	I/E	S/N	T/F	P/J	Average	∇	OPN	CON	EXT	AGR	NEU	Average	∇
BiLSTM	57.82	57.87	69.97	57.01	60.67	-	63.32	62.47	63.54	65.97	56.30	62.32	-
$\text{BERT}_{\text{finetune}}$	63.57	62.15	76.41	63.04	66.29	-	65.13	64.55	67.12	68.14	60.51	65.09	-
AttRCNN	59.74	64.08	78.77	66.44	67.25	-	67.84	63.46	71.50	71.92	62.36	67.42	-
SN+Attn	65.43	62.15	78.05	63.92	67.39	-	68.50	64.19	72.25	70.82	68.10	68.77	-
Transformer-MD	66.08	69.10	79.19	67.50	70.47	-	70.47	68.50	72.79	71.07	69.76	69.51	-
TrigNet	69.54	67.17	79.06	67.69	70.86	6.81	69.52	68.27	70.01	73.12	69.34	70.05	11.26
$\text{CGTN}_{\text{pretrain}}$	**71.66**	69.43	80.14	69.90	72.78	2.95	**72.28**	74.75	76.21	76.01	**73.77**	74.60	6.46
$\text{CGTN}_{\text{joint}}$	71.12	**70.44**	**80.22**	**72.64**	**73.61**	2.52	72.17	**76.21**	**78.78**	**77.12**	70.87	**75.03**	3.25

Table 1: Overall results of CGTN family and baselines in Macro-F1(%) score of kaggle dataset and F1(%) score of Essays dataset, where ∇ denotes difference between training score and testing score.

Methods	Kaggle					Essays					
	I/E	S/N	T/F	P/J	Average	OPN	CON	EXT	AGR	NEU	Average
$\text{CGTN}_{\text{w/o CL}}$	67.34	68.37	77.29	69.27	70.56	71.42	72.13	72.51	74.92	71.70	72.53
$\text{CGTN}_{\text{w/o TC}}$	69.83	70.42	79.55	71.21	72.74	71.59	72.84	74.63	74.20	71.13	72.96
$\text{CGTN}_{\text{joint}}$	**71.12**	**70.44**	**80.22**	**72.64**	**73.61**	**72.17**	**76.21**	**78.78**	**77.12**	**70.87**	**75.03**

Table 2: Results of ablation study in Macro-F1 (%) score on the Kaggle dataset and F1 (%) score on the Essays dataset, where "w/o" means removal of a component from the original CGTN.

- **TrigNet** [Yang *et al.*, 2021c] is a novel flow tripartite graph attention network, which aggregates different posts of each user from a psychological perspective.
- **Transformer-MD** [Yang *et al.*, 2021a] is a novel multi-document Transformer, which aggregates different posts to depict a personality profile for each user without introducing post orders.

5.3 Implementation Details

Following previous works [Yang *et al.*, 2021c; Yang *et al.*, 2021a], we set the max number of posts as 50 for each user and the max length as 70 for each post. For pretraining, the initial learning rate is searched in $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$ and to optimize the contrastive loss on different datasets. The mini-batch size is set as 64. The temperature τ is set as 0.15. We adopt early stopping when the validation loss stops decreasing by 10 epochs. For joint learning, we search the trade-off parameter λ in $\{1, 0.1, 0.01, 0.001, 0.0001\}$ for different datasets. The initial learning rate is also searched in $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$. The settings of batch size, patience for early stopping, and temperature are the same as the pretraining strategy[3].

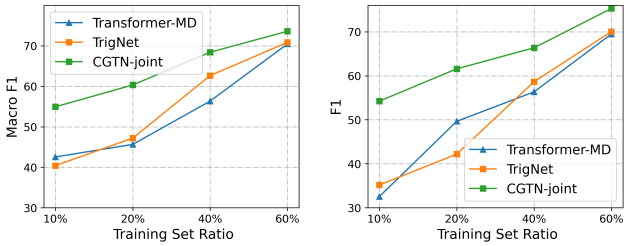
5.4 Overall Results

The overall results are presented in Table 1. The major findings can be summarized as follows, **First**, we can observe that our final model $\text{CGTN}_{\text{joint}}$ achieves the highest scores for both datasets, significantly outperforming the current state-of-the-art model (TrigNet) by 2.75 with t-test $p<0.01$ in Kaggle dataset and 4.98 with t-test $p<0.01$ in Essays dataset. What's more, with pre-training strategy, our model $\text{CGTN}_{\text{pretrain}}$ also achieves significant breakthrough compared to the current SOTA model TrigNet. The results verify the effectiveness of our model in personality detection. We believe the reasons are two fold: (1) Our model CGTN uses contrastive self-supervised learning to learn better post representations which reduces the risk of overfitting on a small training set. (2) Trait correlations are well captured, which injected some psychological clues into for personality detection. **Second**, $\text{CGTN}_{\text{pretrain}}$ and $\text{CGTN}_{\text{joint}}$ performs better on Essays dataset compared with baseline, we measure the difference between the corresponding scores of the training and testing sets compared to that of the supervised paradigm approach TrigNet on each datasets and find that the train-test difference under $\text{CGTN}_{\text{pretrain, joint}}$ is smaller than that under TrigNet. This is even more obvious on Essay dataset which indicates that our method can better mitigate overfitting under small datasets. **Third**, $\text{CGTN}_{\text{joint}}$ generally performs better than $\text{CGTN}_{\text{pretrain}}$. As shown in Table 1, the train-test difference under $\text{CGTN}_{\text{joint}}$ is smaller than $\text{CGTN}_{\text{pretrain}}$ on two datasets, which indicates that the fine-tuned representations are still at risk of bias towards overfitting. Joint learning strategy is probably the better option since the representations in the main and auxiliary tasks are mutually enhanced with each other. **Fourth**, TrigNet and Transformer-MD achieve greater performance compared to the DNNs models, which further implies that making full use of post structure information is essential for personality understanding.

5.5 Ablation Study

We conduct an ablation study of our $\text{CGTN}_{\text{joint}}$ model on both datasets by removing trait correlation component, represented by $\text{CGTN}_{\text{w/o TC}}$, and contrasive learning component, represented by $\text{CGTN}_{\text{w/o CL}}$, to investigate their contributions respectively. As shown in Table 2, we observe that $\text{CGTN}_{\text{joint}}$ outperforms $\text{CGTN}_{\text{w/o TC}}$, suggesting the effectiveness of our approach in modeling trait correlations. In particular, the performance improvement on the Essays dataset is higher than

[3]The code available at https://github.com/yangpu06/CGTN



(a) Kaggle (b) Essays

Figure 2: Performance curves for different Training set ratios.

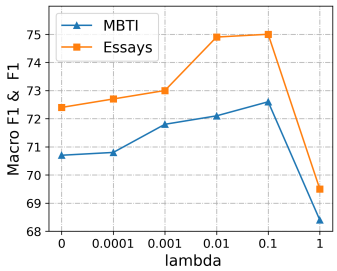


Figure 3: Performance curves for different trade-off parameter.

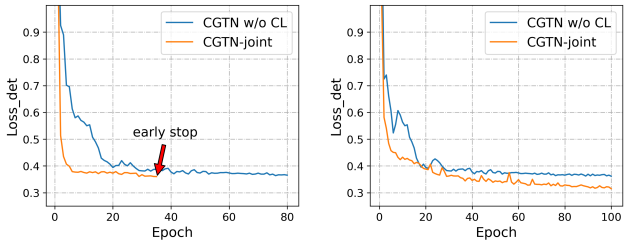
on the Kaggle. We guess that it might be the correlations of Big Five personality traits are slightly higher than that of MBTI indicators. In addition, the performance of $\text{CGTN}_{\text{joint}}$ is also supreior to that of $\text{CGTN}_{\text{w/o CL}}$, especially on Essays dataset, which shows contrasive learning is helpful for increasing the generalization ability of the model, especially on small datasets.

5.6 Impact of Number of Training Samples

We compare our model with two baseline methods with the best performances: Transformer-MD and TrigNet, to study the impact of the ratio of training set. Particularly, we vary the number of training samples and compare their performance on the Kaggle and Essays dataset. We run each method 10 times and report the average performance. As shown in Figure 2, with the increase of training data, all the methods achieve better results in terms of Macro-F1 and F1 on both datasets. Generally, our method outperforms all the other methods consistently. When fewer training data are provided, the baselines exhibit obvious performance drop, while our model still achieves relatively high performance. It demonstrates that our method can more effectively take advantage of the limited labeled data for personality detection. We believe our model benefits from auxiliary signals distilled through contrastive self-supervised learning for personality detection.

5.7 Effect of Trade-off Parameter

Figure 3 demonstrates how Macro-F1 and F1 values change when the trade-off parameter λ in $\text{CGTN}_{\text{joint}}$ increases. from which we can observe that the score first rises as the trade-off parameter λ rises and then begins to drop when λ is larger than 0.1. This is because a bigger value imposes a stronger regularization impact, which helps to reduce overfitting. However, if λ gets too high, the score will drop because excessive regularization impact outweighs the detection loss.



(a) Kaggle (b) Essays

Figure 4: Training curves of $\text{CGTN}_{\text{joint}}$ and $\text{CGTN}_{\text{w/o CL}}$.

5.8 Training Efficiency

We investigate the effect of self-supervised contrastive learning on training efficiency. Figure 4 shows the training curves of $\text{CGTN}_{\text{joint}}$ and $\text{CGTN}_{\text{w/o CL}}$ on Kaggle and Essays datasets. Obviously, $\text{CGTN}_{\text{joint}}$ converges much faster than $\text{CGTN}_{\text{w/o CL}}$ on both datasets. In particular, early stop occurs at the 35-th epochs and arrives at the best performance for $\text{CGTN}_{\text{joint}}$, while it takes more epochs for $\text{CGTN}_{\text{w/o CL}}$ on Kaggle dataset. It demonstrates that contrastive learning task speeds up the detection progress and helps to learn a better model. The Essays dataset shows the same trend, and $\text{CGTN}_{\text{joint}}$ has a lower training loss. The above results verify that the proposed contrasive self-supervised paradigm is effective for such a data-hungry task.

6 Conclusion

In this paper, we proposed a novel Contrastive Graph Transformer Network model (CGTN) for personality detection. CGTN aims to introduce a new learning paradigm to alleviate the data scarcity inherent to personality detection tasks. For this purpose, we designed two types of graph augmentations based on LIWC and post semantics and learned post embeddings from graph self-supervised contrasive learning. Besides, Transformer-based trait generation architecture is designed to exploit correlations among personality traits. Moreover, we used post-to-trait attention to select the vital posts for trait generation. In the end, extensive experimental results on Kaggle and Essays datasets demonstrate the effectiveness and efficiency of our model.

Acknowledgments

This work is supported by the NSFC-General Technology Basic Research Joint Funds under Grant (U1936220), the National Natural Science Foundation of China under Grant (61972047) and the National Key Research and Development Program of China (2018YFC0831500).

References

[Coltheart, 1981] Max Coltheart. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505, 1981.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[John *et al.*, 2008] Oliver P John, Laura P Naumann, and Christopher J Soto. Paradigm shift to the integrative big five trait taxonomy: History, measurement, and conceptual issues. *Handbook of personality: Theory and research*, pages 114–158, 2008.

[Kaushal and Patwardhan, 2018] Vishal Kaushal and Manasi Patwardhan. Emerging trends in personality identification using online social networks—a literature survey. *ACM Transactions on Knowledge Discovery from Data*, 12(2):1–30, 2018.

[Lang *et al.*, 2019] Yining Lang, Wei Liang, Yujia Wang, and Lap-Fai Yu. 3d face synthesis driven by personality impression. In *AAAI*, volume 33, pages 1707–1714, 2019.

[Lynn *et al.*, 2020] Veronica Lynn, Niranjan Balasubramanian, and H Andrew Schwartz. Hierarchical modeling for user personality prediction: The role of message-level attention. In *ACL*, pages 5306–5316, 2020.

[Mairesse *et al.*, 2007] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500, 2007.

[Mehta *et al.*, 2020] Yash Mehta, Samin Fatehi, Amirmohammad Kazameini, Clemens Stachl, Erik Cambria, and Sauleh Eetemadi. Bottom-up and top-down: Predicting personality with psycholinguistic and language model features. In *ICDM*, pages 1184–1189. IEEE, 2020.

[Pennebaker and King, 1999] James W Pennebaker and Laura A King. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296, 1999.

[Ren *et al.*, 2021] Zhancheng Ren, Qiang Shen, Xiaolei Diao, and Hao Xu. A sentiment-aware deep learning approach for personality detection from text. *Information Processing & Management*, 58(3):102532, 2021.

[Sharpe *et al.*, 2011] J Patrick Sharpe, Nicholas R Martin, and Kelly A Roth. Optimism and the big five factors of personality: Beyond neuroticism and extraversion. *Personality and Individual Differences*, 51(8):946–951, 2011.

[Shen *et al.*, 2020] Tiancheng Shen, Jia Jia, Yan Li, Yihui Ma, Yaohua Bu, Hanjie Wang, Bo Chen, Tat-Seng Chua, and Wendy Hall. Peia: Personality and emotion integrated attentive model for music recommendation on social media platforms. In *AAAI*, volume 34, pages 206–213, 2020.

[Štajner and Yenikent, 2020] Sanja Štajner and Seren Yenikent. A survey of automatic personality detection from texts. In *ACL*, pages 6284–6295, 2020.

[Tandera *et al.*, 2017] Tommy Tandera, Derwin Suhartono, Rini Wongso, Yen Lina Prasetio, et al. Personality prediction system from facebook users. *Procedia computer science*, 116:604–611, 2017.

[Tausczik and Pennebaker, 2010] Yla R Tausczik and James W Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Wen *et al.*, 2021] Zhiyuan Wen, Jiannong Cao, Ruosong Yang, Shuaiqi Liu, and Jiaxing Shen. Automatically select emotion for response via personality-affected emotion transition. In *ACL*, pages 5010–5020, 2021.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2018.

[Xue *et al.*, 2018] Di Xue, Lifa Wu, Zheng Hong, Shize Guo, Liang Gao, Zhiyong Wu, Xiaofeng Zhong, and Jianshan Sun. Deep learning-based personality recognition from text posts of online social networks. *Applied Intelligence*, 48(11):4232–4246, 2018.

[Yang *et al.*, 2018] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. Sgm: sequence generation model for multi-label classification. In *COLING*, pages 3915–3926, 2018.

[Yang *et al.*, 2021a] Feifan Yang, Xiaojun Quan, Yunyi Yang, and Jianxing Yu. Multi-document transformer for personality detection. In *AAAI*, volume 35, pages 14221–14229, 2021.

[Yang *et al.*, 2021b] Runzhe Yang, Jingxiao Chen, and Karthik Narasimhan. Improving dialog systems for negotiation with personality modeling. In *ACL*, pages 681–693, 2021.

[Yang *et al.*, 2021c] Tao Yang, Feifan Yang, Haolan Ouyang, and Xiaojun Quan. Psycholinguistic tripartite graph network for personality detection. In *ACL*, pages 4229–4239, 2021.