Leveraging the Dual Capabilities of LLM: LLM-Enhanced Text Mapping Model for Personality Detection

Weihong Bi[1,2*], Feifei Kou[1,2*†], Lei Shi[3,4], Yawen Li[5], Haisheng Li[6], Jinpeng Chen[1,7], Mingying Xu[8]

[1]School of Computer Science (National Pilot School of Software Engineering), BUPT, Beijing, 100876, China
[2] Key Laboratory of Trustworthy Distributed Computing and Service, BUPT, Ministry of Education, Beijing, 100876, China
[3]State Key Laboratory of Media Convergence and Communication, CUC, Beijing, 100024, China
[4]State Key Laboratory of Intelligent Game, Yangtze River Delta Research Institute of NPU, Taicang 215400, China
[5]School of Economics and Management, BUPT, Beijing, 100876, China
[6]Beijing Technology and Business University, Beijing, 100048, China
[7]Xiangjiang Laboratory, Changsha, 410205, China
[8]North China University of Technology, Beijing, 100144, China
bwh2023140721@bupt.edu.cn; koufeifei000@bupt.edu.cn; leiky_shi@cuc.edu.cn; warmly0716@126.com;
lihsh@btbu.edu.cn; jpchen@bupt.edu.cn; xumingying@ncut.edu.cn

Abstract

Personality detection aims to deduce a user's personality from their published posts. The goal of this task is to map posts to specific personality types. Existing methods encode post information to obtain user vectors, which are then mapped to personality labels. However, existing methods face two main issues: first, only using small models makes it hard to accurately extract semantic features from multiple long documents. Second, the relationship between user vectors and personality labels is not fully considered. To address the issue of poor user representation, we utilize the text embedding capabilities of LLM. To solve the problem of insufficient consideration of the relationship between user vectors and personality labels, we leverage the text generation capabilities of LLM. Therefore, we propose the LLM-**E**nhanced **T**ext **M**apping Model (**ETM**) for Personality Detection. The model applies LLM's text embedding capability to enhance user vector representations. Additionally, it uses LLM's text generation capability to create multi-perspective interpretations of the labels, which are then used within a contrastive learning framework to strengthen the mapping of these vectors to personality labels. Experimental results show that our model achieves state-of-the-art performance on benchmark datasets.

Code — https://github.com/BUPT-SN/ETM

Introduction

Personality plays a crucial role in understanding the relationship between individual behaviors and mental activities (Kernberg 2016). It can also serve as a valuable tool for guiding personal growth and career choices. The MBTI is a widely used system for classifying personality, dividing individuals into sixteen categories based on four dimensions (Myers 1987). User posts on social media can be used to detect their MBTI personality type. Therefore, the goal of this task is to map posts to specific personality types, as shown in Figure 1(a). Using deep learning methods to solve personality detection task is generally effective. This approach uses a data-driven method to extract text features from posts, fuses them into a user vector, and then maps the user vector to personality labels, as depicted in Figure 1(b). The previous methods (Keh, Cheng et al. 2019; Jiang, Zhang, and Choi 2020) use BERT (Devlin et al. 2018) to encode each individual post or to encode a concatenated sequence of multiple posts from a user. Some alternative methods (Yang et al. 2021b, 2023a; Zhu et al. 2022) use BERT to obtain semantic features and then apply graph neural networks to enhance the representation of individual posts, followed by average pooling to create a user vector. Other methods (Yang et al. 2023a; Zhu et al. 2022; Zhang et al. 2023) manually extract psychological and statistical features, combine them with text features into a user vector. Previous methods face two main issues. On one hand, these methods use BERT to extract text features, but this small model encoder cannot effectively represent multiple long posts. On the other hand, they fail to capture the relationship between user representation vectors and personality labels. These issues collectively lead to poor performance.

Recently, large language models (LLM) have demonstrated powerful text understanding and generation capabilities, surpassing small models in tasks like translation and question answering (Wang et al. 2023), but perform poorly on classification tasks. Some research (Yang et al. 2023b) has designed prompts based on psychological questionnaires to leverage the capabilities of LLM for personality detection. This approach still underperforms compared to fine-tuning small models, showing that using only LLM with prompts might not be an effective method for personality detection. Some studies (Hu et al. 2024) have used LLM for data augmentation from an emotional perspective, but the results show that this approach does not significantly im-

*Equal contribution.
†Corresponding author.

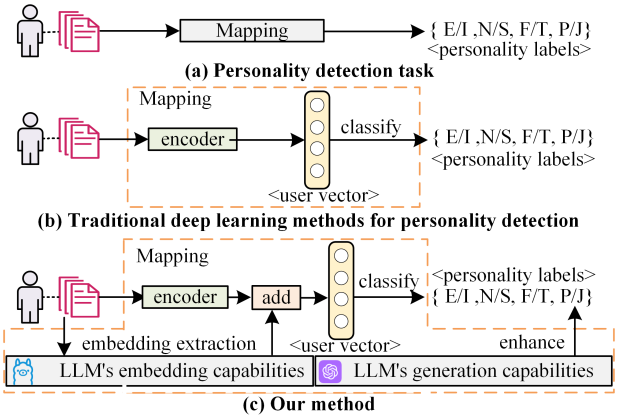

Figure 1: An overview of our method's innovations.

prove performance, indicating that the potential of LLM in personality detection tasks has not been fully realized. Previous research (Freestone and Santu 2024) shows that its distinct pre-training tasks enable it to cluster semantically related words more effectively than traditional encoding models. Therefore, we considered using LLM's distinct text embedding capabilities, along with its widely adopted text generation abilities, to enhance the mapping process from user posts to personality labels, as illustrated in figure 1(c).

In this paper, we introduce a LLM-Enhanced Text Mapping Model for Personality Detection. We use a small model to encode individual posts and apply average pooling to create an initial user representation. These posts are then merged into one long post, and a lightweight language model generates its embedded representation. To better integrate the semantics of the two models, we add a cross-attention mechanism to integrate the user and long post representations. We also use a powerful language model's text generation capabilities to analyze personality labels from three aspects: personality definition, thematic tendency, and text expression. Additionally, we employ contrastive learning to enhance the correlation between documents and labels. The contributions can be summarized as follows:

- We propose an LLM-Enhanced Text Mapping Model for Personality Detection that effectively achieves the goal of accurately mapping posts to specific personality types.
- Our model effectively uses the text embedding capability of LLM to enhance the mapping process of multiple long posts to user vectors. The text generation capability of LLM is used to enhance the mapping process of user vectors to personality labels.
- According to the test results on the benchmark dataset, our model outperforms other existing personality detection models.

Related Work

Traditional and Deep Learning Methods in Personality Detection

For the personality detection task, both traditional methods and deep learning methods aim to better represent the user vector, and then map the user vector to the MBTI labels. Traditional methods extract static features from texts. Machine learning models such as Support Vector Machines (Cui and Qi 2017) and XGBoost (Tadesse et al. 2018) were then used to fit the mapping of posts to MBTI personalities. Traditional machine learning methods rely too much on hand-extracted features, leading to subpar performance. Deep learning methods initially use some feature extraction models, such as LSTM, hierarchical DNNs with AttRCNN, and GRUs with attention mechanisms (Tandera et al. 2017; Xue et al. 2018; Lynn, Balasubramanian, and Schwartz 2020) used a data-driven approach to model the mapping of posts to MBTI personality. Some pre-trained models such as BERT (Devlin et al. 2018) show unique advantages in text feature extraction tasks. Some studies (Keh, Cheng et al. 2019; Jiang, Zhang, and Choi 2020)use text pre-training models to extract features from posts and map the user vectors to MBTI labels. Other studies aim to enhance the representation of individual posts by constructing relationships between posts, thereby improving the overall user vector representation. Transformer MD (Yang et al. 2021a) leverages Transformer XL's memory to store posts and uses attention mechanisms to capture and fuse relationships between them. Some studies (Yang et al. 2021b; Zhu et al. 2022; Yang et al. 2023a) use psychological statistical feature similarity or post embedding similarity to measure whether there is correlation between posts, and then use graph neural networks to build the topology of post sets and carry out feature fusion between posts. Deep learning methods that rely only on small model semantic encoders to extract semantic features from multiple long texts often produce low-quality user vectors, and the relationship between these vectors and MBTI labels is not fully established.

Large Language Models in Personality Detection

Large Language Models (LLM) are used to solve some natural language tasks due to their powerful in-context learning(ICL) ability and extensive knowledge reserve (Brown et al. 2020; Rae et al. 2021; Thoppilan et al. 2022; Chowdhery et al. 2023; Achiam et al. 2023). Some new methods use LLM to help solve personality detection tasks. One way to use LLM to solve complex problems is to break down a complex problem into several small problems and then guide LLM to solve these problems in turn (Wei et al. 2022). Wei et al. (Yang et al. 2023b) adapted a psychological questionnaire into multiple questions, directing language models to answer based on post content and then drawing conclusions from the answers obtained in each round. The method using language models for personality detection relies heavily on prompt construction, leading to suboptimal results. Using LLM for data augmentation from an emotional perspective is also a way to leverage LLM. Hu et al. (Hu et al. 2024)used LLM to enhance posts from three perspectives, semantic, semantic and semantic, and then used comparative learning to establish the relationship between enhanced posts and initial posts. This method only uses the text generation capability of LLM, resulting in little improvement of the mapping process of user posts to personality labels in the process of this method.

Approach

In this section, we first define the problem of personality detection, then conduct preliminary explorations using the method of fine-tuning lightweight LLM, and finally provide detailed descriptions of the two enhancement modules in our ETM model.

Problem Definition

The personality detection task involves mapping posts to the personality types. Each posts, denoted as $P = \{p_1, p_2, \ldots, p_n\}$, consists of M tokens per post, represented as $p_i = [w_{i1}, w_{i2}, \ldots, w_{iM}]$. The personality types is formalized as $Y = (\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \boldsymbol{y_4})$, with each component $\boldsymbol{y_i}$ taking a value of 0 or 1. This task aims to establish a mapping from P to Y.

Exploring Lightweight Large Language Models for Personality Detection

Lightweight Large Language Models retain the essential capabilities of traditional language models while being optimized for task-specific training (Yang et al. 2023c). In our experiments, we initially explored the use of lightweight LLM for personality detection through zero-shot learning, binary classification fine-tuning, and sixteen-class fine-tuning. first, we establish a baseline by using a lightweight LLM with zero-shot learning. Then, we explore two approaches: fine-tuning for sixteen-class classification and fine-tuning for binary classification. Although these methods outperform direct zero-shot learning, they still do not reach the baseline performance achieved with fine-tuned small models. The experimental results can be seen in the "Performance of Lightweight LLM" section of the Experiments. The extensive pre-training of lightweight large language models on varied datasets has often reduced their effectiveness when fine-tuning for specific text classification tasks. In response, we explored the use of LLMs to address the challenges of poor user vector representation and the inadequate relationship between user vectors and personality labels typically encountered with small models in personality detection. This strategic application of LLM allowed our approach to ultimately achieve state-of-the-art performance.

An Overview of Our ETM Model Architecture

In this paper, we propose an LLM-Enhanced Text Mapping Model for Personality Detection. As the overall architecture shows in Figure 2. We first use BERT to encode each post individually, then apply average pooling to generate the initial user vectors. These posts are then combined into a single text. To enhance its text embedding, we apply a lightweight large language model, capturing the text representation from the model's unique perspective. This vector is subsequently refined using a cross-attention mechanism to enhance its representational accuracy. Then, we leverage the text generation capabilities of powerful LLM to interpret MBTI labels from three dimensions: personality definition, thematic tendency, and text expression. This interpretation is encoded using a small model. Finally, a contrastive learning framework strengthens the mapping from user vectors to MBTI labels.

Lightweight LLM Enhance User Posts to Vector Representation

We followed the previous approach (Keh, Cheng et al. 2019; Hu et al. 2024) to get the initial representation of the user vector, and then used the lightweight LLM to get a new perspective on the user representation. Then, using the cross-attention mechanism, the user representation from a new perspective is used to strengthen the initial representation, so as to obtain an enhanced user vector representation.

For a post p_i in a post set P, we use the BERT model as a text encoder to encode p_i, and then use the token at the `[CLS]` position as the feature representation of the p_i post to denote $\boldsymbol{h_i}$.

$$\boldsymbol{h_i} = \text{Encoder}(p_i) \tag{1}$$

Multiple long documents generated by a user, totaling N, are encoded separately, and then an initial user vector u is obtained by means of average pooling:

$$\boldsymbol{u} = \text{mean}([\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_N}]) \tag{2}$$

To enhance the semantic representation beyond BERT, we leverage lightweight LLM to process concatenated user posts as extended contexts. Using the Llama3 model (Zhang et al. 2024), we extract embeddings from the concatenated texts:

$$P_{long} = [p_1|p_2|\ldots|p_N] \tag{3}$$

where | denotes the concatenation operator.

Based on previous findings (Geva et al. 2020), which indicate that deeper insights are derived from upper layers of transformer-based models, we utilize the embeddings from the last d_m layer. To synthesize a comprehensive user vector $\boldsymbol{U_{llama}}$, we perform average pooling across and within these layers, capturing richer semantic details from the lightweight LLM.

$$[\boldsymbol{h_0}, \boldsymbol{h_1}, \ldots, \boldsymbol{h_L}]_n = \text{Llama}(P_{long}, n) \tag{4}$$

$$\boldsymbol{U_n} = \text{mean}([\boldsymbol{h_0}, \boldsymbol{h_1}, \ldots, \boldsymbol{h_L}]_n) \tag{5}$$

$$\boldsymbol{U_{llama}} = \text{mean}([\boldsymbol{U_{33-d_m}}, \boldsymbol{U_{34-d_m}}, \ldots, \boldsymbol{U_{32}}]) \tag{6}$$

where n denotes the n-th layer and $\boldsymbol{L}$ represents the number of tokens after text embedding.

To enhance the initial user vector with additional insights from the lightweight LLM, we employed a cross-attention mechanism for effective vector fusion. To match the differing token dimensions of the BERT and Llama3 models, we utilized transformation matrices $\boldsymbol{W_{o1}}$ and $\boldsymbol{W_{o2}}$. The vectors were then mapped to the $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ spaces of the attention mechanism using three fully connected layers: $\boldsymbol{W_Q}$, $\boldsymbol{W_K}$, and $\boldsymbol{W_V}$:

$$\boldsymbol{Q} = \boldsymbol{u}\,\boldsymbol{W_Q} \tag{7}$$

$$\boldsymbol{K} = (\boldsymbol{U}_{\text{llama}}\,\boldsymbol{W_{o1}})\,\boldsymbol{W_K} \tag{8}$$

$$\boldsymbol{V} = (\boldsymbol{U}_{\text{llama}}\,\boldsymbol{W_{o2}})\,\boldsymbol{W_V} \tag{9}$$

The cross-attention mechanism generates an $\boldsymbol{H_{llama}}$ vector that incorporates insights from a new perspective. This vector is then fused with the initial user vector, resulting in

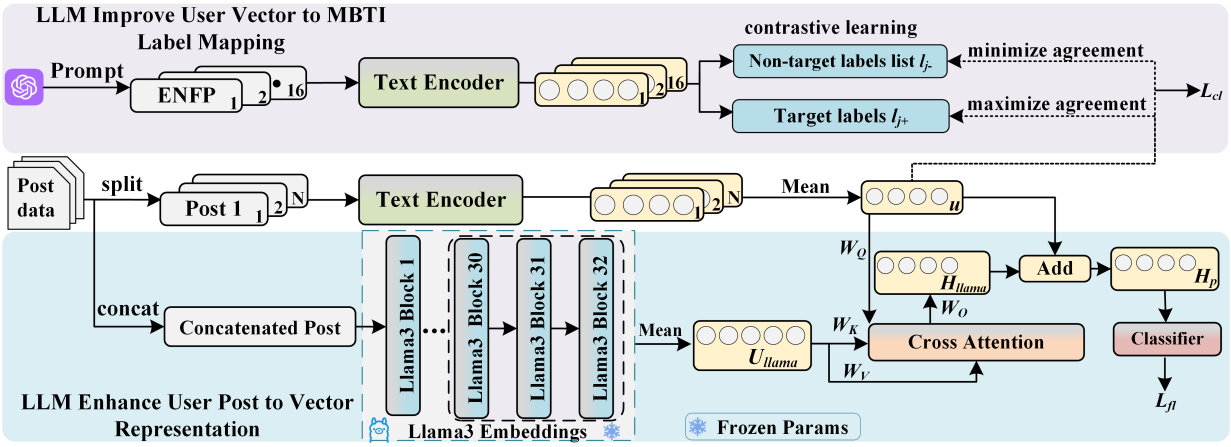

Figure 2: The overall pipeline of the ETM.

the final enhanced user vector $\boldsymbol{H_p}$. Here, d_k denotes the dimension of the $\boldsymbol{K}$ space.

$$\boldsymbol{H}_{llama} = \text{softmax}\left(\frac{\boldsymbol{QK}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{10}$$

$$\boldsymbol{H}_P = \boldsymbol{H}_{llama} + \boldsymbol{u} \tag{11}$$

Powerful LLM Improve User Vector to MBTI Label Mapping

MBTI labels represent specific personality traits, each with its inherent meaning (Myers 1987). By positioning the semantic encoding of user posts closer to their corresponding labels and farther from unrelated labels, we can improve the accuracy of subsequent classification tasks. Therefore, we plan to use the advanced model GPT-4 (Achiam et al. 2023) to interpret MBTI personality labels from three dimensions: type definition, thematic inclination, and mode of expression. Following this, we will employ the contrastive learning framework proposed by Chen (Chen et al. 2020) to establish the relationship between the semantics of the post collection and the label meaning.

To reduce potential hallucinations (Huang et al. 2023) by GPT-4, we provide it with official interpretations of the 16 MBTI personality labels as part of the prompt. Due to space constraints, we replace the full MBTI personality interpretations with {personality authoritative interpretations} and the specific social network context with {description of the social media context} in the prompt. The prompt is as follows:

The official interpretations of the 16 personality types is: {personality authoritative interpretations}. Based on the following social media context: {description of the social media context}, generate discussion content interpretations for the 16 MBTI personality types on a social platform. Each interpretation should include the type's definition, thematic tendencies, and typical modes of expression.

First, obtain detailed interpretations of the MBTI personality types, then encode these interpretations using a text encoder, and finally apply the max pooling method to maximize the retention of MBTI personality interpretations:

$$[\boldsymbol{h}_{\boldsymbol{j}_0}, \boldsymbol{h}_{\boldsymbol{j}_1}, \ldots, \boldsymbol{h}_{\boldsymbol{j}_L}] = \text{Encoder}(l_j) \tag{12}$$

$$\boldsymbol{l}_{\boldsymbol{j}_+} = \text{MaxPooling}([\boldsymbol{h}_{\boldsymbol{j}_{+0}}, \boldsymbol{h}_{\boldsymbol{j}_{+1}} \ldots, \boldsymbol{h}_{\boldsymbol{j}_{+L}}]) \tag{13}$$

$$\boldsymbol{l}_{\boldsymbol{j}_-} = \text{MaxPooling}([\boldsymbol{h}_{\boldsymbol{j}_{-0}}, \boldsymbol{h}_{\boldsymbol{j}_{-1}} \ldots, \boldsymbol{h}_{\boldsymbol{j}_{-L}}]) \tag{14}$$

where $\boldsymbol{l}_{\boldsymbol{j}_+}$ corresponds to the label associated with the post collection, and $\boldsymbol{l}_{\boldsymbol{j}_-}$ corresponds to the remaining labels that do not pair with the post collection. Here, L represents the number of tokens after text embedding.

The contrastive loss L_{cl} is defined as:

$$L_{cl} = -\log \frac{e^{\text{sim}(\boldsymbol{u}, \boldsymbol{l}_{j_+})/\tau}}{e^{\text{sim}(\boldsymbol{u}, \boldsymbol{l}_{j_+})/\tau} + \sum_{j=1}^{15} e^{\text{sim}(\boldsymbol{u}, \boldsymbol{l}_{j_-})/\tau}} \tag{15}$$

where τ is a temperature parameter that adjusts the sensitivity of similarity scores.

Joint Learning

We employ the focal loss function (Lin et al. 2017) to better target hard-to-classify samples and lessen focus on simpler ones. A linear layer maps the user vector to match the dimensions of MBTI labels Y, and softmax converts these to probabilities:

$$y = \text{softmax}(\boldsymbol{H_P W_u} + \boldsymbol{b_u}) \tag{16}$$

where $\boldsymbol{W_u}$ represents the weight matrix, and $\boldsymbol{b_u}$ is the bias term used in the softmax calculation.

The focal loss is defined as:

$$L_{fl} = \frac{1}{V}\sum_{i=1}^{V}\sum_{j=1}^{T}[-\alpha(1-p(\hat{y}_{ji}|\theta))^{\gamma} y_{ji} \log p(\hat{y}_{ji}|\theta)] \tag{17}$$

where α is a weighting factor to balance the importance of different classes, and γ adjusts the rate at which easy examples are down-weighted. The variable V denotes the total number of training samples, and y corresponds to the actual label for each dimension of personality traits. $p(\hat{y}|\theta)$ indicates the predicted probability for each dimension, computed based on model parameters θ. Here, T represents the number of personality dimensions, typically $T = 4$.

Furthermore, we combine the focal loss with the contrastive loss:

$$L = L_{fl} + \lambda L_{cl} \tag{18}$$

where λ is a hyperparameter adjusting the relative importance of each loss in the objective function.

Experiments

Datasets

Considering the datasets employed in prior research (Hu et al. 2024; Yang et al. 2023a, 2021a,b), we also choose to use the Kaggle[1] and Pandora[2] MBTI personality datasets for our experiments. The Kaggle dataset comes from PersonalityCafe[3] and includes over 8,600 entries. Each entry lists an individual's four-letter MBTI type and excerpts from their 50 most recent posts. The Pandora dataset is from Reddit[4] and features posts from 9,067 users with self-reported MBTI types. The number of posts varies, with each user having from dozens to hundreds of posts. To prevent information leaks, words related to personality label are replaced with ⟨mask⟩ (Yang et al. 2023a). We also adopt their data partitioning strategy, splitting the datasets into training, validation, and testing sets using a 60-20-20 ratio. Performance is evaluated using the Macro-F1 metric.

Table 1 presents the distribution of MBTI types and the number of analyzed posts for each dataset. While the Kaggle dataset retains its original label distribution, the Pandora dataset, due to its more pronounced class imbalance, employs undersampling strategies during the training, validation, and testing phases to ensure balanced labels along each MBTI dimension. With these adjustments, our model achieves exceptional performance on the Pandora dataset while maintaining stable, state-of-the-art results on the Kaggle dataset. These findings confirm that our model design consistently demonstrates robust generalization and high-level performance under varying distributional conditions.

Dataset	Types	Train	Validation	Test
Kaggle	I/E	4032/1173	1330/405	1314/421
	S/N	724/4481	230/1505	243/1492
	T/F	2388/2817	802/933	791/944
	P/J	3160/2045	1007/728	1074/661
Pandora	I/E	4314/1126	1425/388	1403/411
	S/N	621/4819	202/1611	205/1609
	T/F	3527/1913	1160/653	1164/650
	P/J	3211/2229	1064/749	1035/779

Table 1: Statistics of the Kaggle and Pandora datasets.

Baselines

SVM (Cui and Qi 2017) and **XGBoost (Tadesse et al. 2018)**: This method combines all user posts into one long document, extracts features using a bag-of-words model, and processes the data with classification algorithms like SVM or XGBoost.

BiLSTM (Tandera et al. 2017): This method uses a Bi-LSTM architecture with average pooling to merge post embeddings into a single representation for personality prediction.

BERTconcat (Jiang, Zhang, and Choi 2020): This method concatenates a user's posts into one long post, extracts features using the BERT model, and maps these features to personality labels through fully connected layers.

BERTmean (Keh, Cheng et al. 2019): This approach encodes posts using the BERT model, applies average pooling to create user feature representation, and maps these features to personality labels via fully connected layers.

AttRCNN (Xue et al. 2018): This method uses a hierarchical deep neural network that combines an AttRCNN structure with an Inception variant to extract deep semantic features from social network texts. These features are then combined with statistical linguistic features and fed into regression algorithms.

AttnSeq (Lynn, Balasubramanian, and Schwartz 2020): This method uses a hierarchical attention mechanism to process posts, applying word-level and message-level attentions for personality prediction.

Transformer-MD (Yang et al. 2021a): This method uses a Multi-Document Transformer architecture to encode posts without order bias, utilizing memory tokens with shared position embeddings. This allows dynamic access to information across posts, creating a coherent personality profile across multiple documents.

TrigNet (Yang et al. 2021b): This method uses a psycholinguistic tripartite graph network, which combines a BERT-based initializer with a graph attention mechanism to integrate psycholinguistic knowledge for text-based personality detection.

D-DGCN (Yang et al. 2023a): This method employs a Dynamic Deep Graph Convolutional Network to detect personality traits from social media posts. It constructs graphs dynamically, with posts as nodes using multi-hop connectivity and deep graph convolutional layers, reducing biases from post order.

TAE (Hu et al. 2024): This method combines LLM-based text augmentation with a small model to improve personality detection. It uses LLM to generate augmented posts focusing on semantic, sentiment, and linguistic aspects, enhancing data and personality label representations.

Implementation Details

Our deep learning models are developed using PyTorch (Paszke et al. 2017), utilizing AdamW (Loshchilov and Hutter 2017) as the optimizer. The learning rate is set to 3×10^{-5}. We conducted our experiments on a setup with an NVIDIA A6000 GPU. We use `BERT-base-uncased` as the text encoder and `Meta-Llama-3-8B-Instruct` as the lightweight LLM text embedding extraction tool. The model setup specifies a batch size of 4, with the temperature parameter (τ) maintained at 0.07 and the trade-off parameter (λ) at 1. Each post is limited to 128 tokens. Dataset limits are set to 50 posts for Kaggle and 100 posts for Pandora.

We use `GPT-4`[5] for interpreting MBTI labels and `Meta-Llama-3-8B-Instruct` for evaluating the performance of fine-tuned lightweight models on personality

[1]https://www.kaggle.com/datasnaek/mbti-type

[2]https://psy.takelab.fer.hr/datasets/all

[3]http://personalitycafe.com/forum

[4]https://www.reddit.com

[5]https://chat.openai.com

Posts: I am all about the sacrament of reconciliation (confession)x85...	
Prompt	Please determine the author's MBTI type based on the following text. Please answer the MBTI type directly with out any other explanation. MBTI has four letters, the first letter is E or I, the second letter is N or S, the third letter is T or F, and the fourth letter is J or P. The above is the content of the posts. Your response must be one of the sixteen MBTI types. Here are the sixteen types enumerated: ISTJ, ISFJ, INFJ, INTJ, ISTP, ISFP, INFP, INTP, ESTP, ESFP, ENFP, ENTP, ESTJ, ESFJ, ENFJ, ENTJ.
Output	INFP

(a) Prompt Template for Llama3 MBTI 16-Type Classification

Prompt: Based on the text paragraph, would you infer the author is more extroverted (E) or introverted (I)?	**Output**: I
Prompt: Considering the details in the text paragraph, do you think the author relies more on sensing (S) or intuition (N)?	**Output**: N
Prompt: From the given text paragraph, does it appear that the author makes decisions based on thinking (T) or feeling (F)?	**Output**: F
Prompt: Analyzing the text paragraph, would you say the author prefers judging (J) or perceiving (P)?	**Output**: P

(b) Prompt Template for Llama3 MBTI Binary Classification

Figure 3: Prompts for fine-tuning lightweight LLM.

detection task. The fine-tuning process was conducted under the LoRA framework (Hu et al. 2021). We selected a rank of 16 and conducted the fine-tuning over a duration of 5 epochs. The learning rate was set at 0.0001, and the training was carried out with a batch size of 8.

Performance of Lightweight LLM

In the task of fine-tuning lightweight LLM for personality detection, it can be understood as involving a 16-category classification for MBTI personality types, with the prompt used shown in Figure 3 (a). Alternatively, for binary classification of each MBTI dimension, the prompt used is shown in Figure 3 (b). Zero-shot methods are also used as a baseline experiment to verify whether fine-tuning is effective.

The experimental results, as shown in Table 2, indicate that the fine-tuning performs better than the zero-shot approach but is still significantly lower than the baseline using the small BERT model. The results from experiments using fine-tuned lightweight LLM, alongside insights from previous work (Hu et al. 2024) on employing ChatGPT for personality detection, indicate that relying only on LLM for personality detection can lead to poor performance.

Methods	**Kaggle**				
	I/E	**S/N**	**T/F**	**P/J**	**Avg**
Llama3+ zero-shot	43.63	48.66	47.55	49.27	47.28
Llama3+ FT (2)	48.75	46.25	48.54	48.62	48.04
Llama3+ FT (16)	48.80	49.26	47.55	50.26	48.97
BERT_mean	64.05	57.82	77.06	65.25	66.04
ETM(our)	68.97	71.21	86.19	84.78	77.79

Table 2: Performance comparison on Kaggle dataset.

Overall Results

Table 3 shows that our ETM method surpasses all existing baseline models on the benchmark dataset in terms of Macro-F1 scores. Specifically, our ETM model secures performance gains of 9.78% and 11.51% over TrigNet, and 9.03% and 7.12% over D-DGCN on the benchmark datasets by effectively utilizing concatenated post representations, a feature that previous methods overlooked by ignoring post order. Additionally,Our ETM model outperforms the BERT concat method by 28.35% and 22%, addressing its limitations in context length and truncation, thanks to the lightweight LLM's improved capacity for handling extended context. Our model outperforms TAE by 7.94% and 4.31% on the benchmark datasets. Although TAE uses personality label semantics to produce soft labels, it doesn't fully leverage this information, resulting in only minor improvements in ablation studies. In contrast, our model fully interprets label information through LLM and integrates it into a contrastive learning framework, significantly refining the relationship between user vectors and the meanings of the 16 personality labels. Overall, our model achieves outstanding performance due to two key enhancements. Firstly, the integration of a lightweight LLM significantly boosts the small model encoder's capability to process extended texts and deliver distinct semantic insights. Secondly, we employ a powerful LLM to generate multi-dimensional interpretations of personality labels, which are effectively incorporated within a contrastive learning framework.

Ablation Study

To evaluate the significance of each component within our ETM model, we conducted an ablation study using the Kaggle dataset, as shown in Table 4. By employing a lightweight LLM to enhance user representation, removing this enhancement led to an 8.5% decrease in overall performance, confirming the significance of the diverse embedding semantics of lightweight LLM as a complement to textual features. Furthermore, we utilized the powerful generative model to interpret personality labels and employed a contrastive learning framework to establish the relationship between user vectors and label vectors. The removal of this component resulted in a performance decrease of 6.97%, highlighting the value of using multi-dimensional personality labels in the mapping process. We interpret personality labels from three perspectives: type definitions, thematic inclinations, and modes of expression. Removing these interpretations individually led to performance decreases of 1.95%, 1.63%, and 0.98% respectively, showing that type definitions have the greatest impact on performance among the multi-dimensional interpretations. When both key components are removed, the performance deteriorates by 12.99%, further validating the architecture's efficacy in personality detection. These results demonstrate that our model effectively leverages the text embedding and generation capabilities of LLMs to significantly enhance the mapping process from user posts to personality labels.

Impact of Llama3 Layer Selection

The Llama3 model is used as a crucial component to obtain representations of user posts from a lightweight LLM perspective, thereby enhancing the user vector representations. Given that the deeper layers of the transformer architecture can capture richer semantic information, we set up a comparative experiment by selecting the last d_m embedding layers of Llama3 for average pooling between layers. As shown in

Methods	Kaggle					Pandora				
	I/E	S/N	T/F	P/J	Avg	I/E	S/N	T/F	P/J	Avg
SVM (Cui and Qi 2017)	53.34	47.75	76.72	63.03	60.21	44.74	46.92	64.62	56.32	53.15
XGBoost (Tadesse et al. 2018)	56.67	52.85	75.42	65.94	62.72	45.99	48.93	63.51	55.55	53.50
BiLSTM (Tandera et al. 2017)	57.82	57.87	69.97	57.01	60.67	48.01	52.01	63.48	56.21	54.93
BERT_concat (Jiang, Zhang, and Choi 2020)	58.33	53.88	69.36	60.88	60.61	54.22	49.15	58.31	53.14	53.91
BERT_mean (Keh, Cheng et al. 2019)	64.05	57.82	77.06	65.25	66.04	56.60	48.71	64.70	56.07	56.52
AttRCNN (Xue et al. 2018)	59.74	64.08	78.77	66.44	67.25	48.55	56.19	64.39	57.26	56.60
AttnSeq (Lynn, Balasubramanian, and Schwartz 2020)	65.43	62.15	78.05	63.92	67.39	56.98	54.78	60.95	54.81	56.88
Transformer-MD (Yang et al. 2021a)	66.08	69.10	79.19	67.50	70.47	55.26	58.77	69.26	60.90	61.05
TrigNet (Yang et al. 2021b)	69.54	67.17	79.06	67.69	70.86	56.69	55.57	66.38	57.27	58.98
D-DGCN (Yang et al. 2023a)	68.41	65.66	79.56	67.22	70.21	61.55	55.46	**71.07**	59.96	62.01
D-DGCN+ℓ_0 (Yang et al. 2023a)	69.52	67.19	80.53	68.16	71.35	59.98	55.52	70.53	59.56	61.40
TAE (Hu et al. 2024)	**70.90**	66.21	81.17	70.20	72.07	62.57	61.01	69.28	59.34	63.05
ETM (our)	68.97	**71.21**	**86.19**	**84.78**	**77.79**	**68.57**	**64.91**	66.07	**63.53**	**65.77**

Table 3: Performance comparison on Kaggle and Pandora datasets.

Methods	Kaggle				
	I/E	S/N	T/F	P/J	Avg
$\text{ETM}_{\text{w/o llama3-boost}}$	61.10	56.98	84.12	82.51	71.18
$\text{ETM}_{\text{w/o gpt4-cl}}$	57.50	65.31	83.00	83.68	72.37
$\text{ETM}_{\text{w/o definition}}$	67.95	67.54	85.17	84.42	76.27
$\text{ETM}_{\text{w/o tendencies}}$	69.48	68.31	85.74	82.54	76.52
$\text{ETM}_{\text{w/o expression}}$	68.28	69.58	85.24	85.03	77.03
$\text{ETM}_{\text{w/o all}}$	55.31	49.00	82.92	83.50	67.68
ETM (our)	68.97	71.21	86.19	84.78	77.79

Table 4: Results of ablation study on Macro-F1 on the Kaggle dataset.

Table 5, choosing the last five embedding layers of Llama3 is more effective in helping the small model enhance the representation of user vectors.

Methods	Kaggle				
	I/E	S/N	T/F	P/J	Avg
$\phi^{(\text{Llama3},d_m)}_{d_m=1}$	68.19	68.82	82.65	85.09	76.19
$\phi^{(\text{Llama3},d_m)}_{d_m=1,2}$	69.40	68.07	83.48	85.16	76.53
$\phi^{(\text{Llama3},d_m)}_{d_m=1,2,3}$	67.97	67.52	85.65	80.51	75.41
$\phi^{(\text{Llama3},d_m)}_{d_m=1,2,3,4}$	68.51	68.71	85.73	82.45	76.35
$\phi^{(\text{Llama3},d_m)}_{d_m=1,2,3,4,5}$	68.97	71.21	86.19	84.78	77.79

Table 5: Performance of Selecting Llama's Last d_m Layer Embeddings.

Effect of Trade-Off Parameter

We tested various λ in the ETM, from $\{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5\}$. Figure 4 demonstrates that the model's performance is optimal at $\lambda = 1$ across the benchmark datasets, with a decline in performance as λ exceeds this value. This indicates that a moderate increase in λ improves the mapping from user vectors to personality labels by via contrastive learning. However, excessively high λ values cause the focus loss to become increasingly insignificant in the optimization objective, resulting in poorer performance. Therefore, $\lambda = 1$ is identified as the optimal setting.

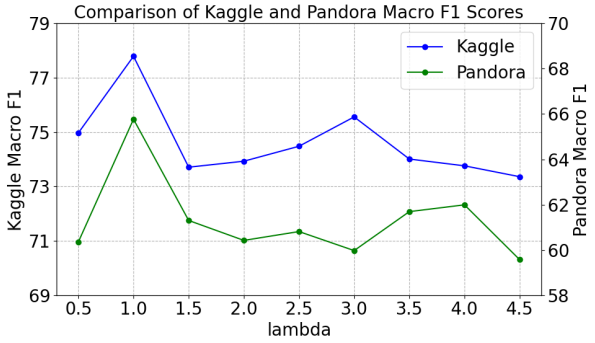

Figure 4: Performance curves for different trade-off parameter.

Conclusion

In this paper, we propose the LLM-Enhanced Text Mapping Model for Personality Detection, which achieves the goal of accurately mapping posts to specific personality types. Our method leverages the text embedding and text generation capabilities of LLM to address the issues of poor user vector representation and the insufficient relationship between user vectors and personality labels in small model-based personality detection. Firstly, we employ lightweight LLM text embeddings for concatenated documents, enhanced by a cross-attention mechanism to improve user vector accuracy. Secondly, we use a powerful LLM to deliver multidimensional explanations of personality labels. This is integrated with a contrastive learning framework that better maps text to labels, enhancing the process. Our model outperforms the best existing baseline methods on benchmark datasets, achieving improvements of 7.94% and 4.31%. In future work, we plan to build a knowledge graph focused on emotion theory and psychology to enhance text-based emotion recognition with LLMs, improving our personality detection model.

Acknowledgments

The work is supported by the National Science and Technology Major Project (2023ZD0121503), by Fundamental Research Funds for the Central Universities (No. 2023RC08, JLU (No.93K172024K17)), by scientific research program of Beijing Municipal Education Commission KZ202110011017, by National Natural Science Foundation of China (U22B2038, 62272058, U23A20319, 62277001), by Open Research Subject of State Key Laboratory of Intelligent Game (No. ZBKF-24-12). by the 8th Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), by Open Project of Xiangjiang Laboratory (No. 23XJ03006).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Cui, B.; and Qi, C. 2017. Survey analysis of machine learning methods for natural language processing for MBTI Personality Type Prediction. *Final Report Stanford University*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Freestone, M.; and Santu, S. K. K. 2024. Word Embeddings Revisited: Do LLMs Offer Something New? *arXiv preprint arXiv:2402.11094*.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hu, L.; He, H.; Wang, D.; Zhao, Z.; Shao, Y.; and Nie, L. 2024. LLM vs Small Model? Large Language Model Based Text Augmentation Enhanced Personality Detection Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18234–18242.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Jiang, H.; Zhang, X.; and Choi, J. D. 2020. Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (student abstract). In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13821–13822.

Keh, S. S.; Cheng, I.; et al. 2019. Myers-Briggs personality classification and personality-specific language generation using pre-trained language models. *arXiv preprint arXiv:1907.06333*.

Kernberg, O. F. 2016. What is personality? *Journal of personality disorders*, 30(2): 145–156.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lynn, V.; Balasubramanian, N.; and Schwartz, H. A. 2020. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 5306–5316.

Myers, I. B. 1987. *Introduction to type: A description of the theory and applications of the Myers-Briggs Type Indicator*. Consulting Psychologists Press.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Tadesse, M. M.; Lin, H.; Xu, B.; and Yang, L. 2018. Personality predictions based on user behavior on the facebook social media platform. *IEEE Access*, 6: 61959–61969.

Tandera, T.; Suhartono, D.; Wongso, R.; Prasetio, Y. L.; et al. 2017. Personality prediction system from facebook users. *Procedia computer science*, 116: 604–611.

Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Wang, S.; Sun, X.; Li, X.; Ouyang, R.; Wu, F.; Zhang, T.; Li, J.; and Wang, G. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xue, D.; Wu, L.; Hong, Z.; Guo, S.; Gao, L.; Wu, Z.; Zhong, X.; and Sun, J. 2018. Deep learning-based personality

recognition from text posts of online social networks. *Applied Intelligence*, 48(11): 4232–4246.

Yang, F.; Quan, X.; Yang, Y.; and Yu, J. 2021a. Multi-document transformer for personality detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 14221–14229.

Yang, T.; Deng, J.; Quan, X.; and Wang, Q. 2023a. Orders are unwanted: dynamic deep graph convolutional network for personality detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 13896–13904.

Yang, T.; Shi, T.; Wan, F.; Quan, X.; Wang, Q.; Wu, B.; and Wu, J. 2023b. Psycot: Psychological questionnaire as powerful chain-of-thought for personality detection. *arXiv preprint arXiv:2310.20256*.

Yang, T.; Yang, F.; Ouyang, H.; and Quan, X. 2021b. Psycholinguistic tripartite graph network for personality detection. *arXiv preprint arXiv:2106.04963*.

Yang, Y.; Sun, H.; Li, J.; Liu, R.; Li, Y.; Liu, Y.; Huang, H.; and Gao, Y. 2023c. Mindllm: Pre-training lightweight large language model from scratch, evaluations and domain applications. *arXiv preprint arXiv:2310.15777*.

Zhang, B.; Huang, Y.; Cui, W.; Huaping, Z.; and Shang, J. 2023. PsyAttention: Psychological Attention Model for Personality Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3398–3411.

Zhang, P.; Zeng, G.; Wang, T.; and Lu, W. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Zhu, Y.; Hu, L.; Ge, X.; Peng, W.; and Wu, B. 2022. Contrastive Graph Transformer Network for Personality Detection. In *IJCAI*, 4559–4565.