Information Processing and Management 60 (2023) 103422

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

DesPrompt: Personality-descriptive prompt tuning for few-shot personality recognition

Zhiyuan Wen, Jiannong Cao, Yu Yang [*], Haoli Wang, Ruosong Yang, Shuaiqi Liu

Department of Computing, The Hong Kong Polytechnic University, Yuk Choi Road 11, Hung Hom, Kowloon, Hong Kong Special Administrative Region

ARTICLE INFO

Keywords:
Personality recognition
Prompt-tuning
Text classification

ABSTRACT

Personality recognition in text is a critical problem in classifying personality traits from the input content of users. Recent studies address this issue by fine-tuning pre-trained language models (PLMs) with additional classification heads. However, the classification heads are often insufficiently trained when annotated data is scarce, resulting in poor recognition performance. To this end, we propose DesPrompt to tune PLM through personality-descriptive prompts for few-shot personality recognition, without introducing additional parameters. DesPrompt is based on the lexical hypothesis of personality, which suggests that personalities are revealed by descriptive adjectives. Specifically, DesPrompt models personality recognition as a word-filling task. The input content is first encapsulated with personality-descriptive prompts. Then, the PLM is supervised to fill in the prompts with label words describing personality traits. The label words are selected from trait-descriptive adjectives from psychology findings and lexical knowledge. Finally, the label words filled in by PLM are mapped into the personality labels for recognition. Our approach aligns with the Masked Language Modeling (MLM) task in pre-training PLMs. So, it efficiently utilizes pre-trained parameters to reduce dependence on annotated data. Experiments on four public datasets show that DesPrompt outperforms conventional fine-tuning and other prompt-based methods, especially in zero-shot and few-shot settings.

1. Introduction

Personality comprises a set of relatively stable traits stemming from individuals' values, attitudes, memories, social relationships, and habits (Mischel, Shoda, & Ayduk, 2007). These traits are widely implied in posts on social media, product comments, self-report essays, and even dialog content. Recognizing personality from these text materials enhances plenty of web applications: *e.g.*, personalized product recommendations, customer service, and community detection in social networks.

Personality recognition in text is solved as a text classification task, where the input is the text content, and the output is the personality category label. Many attempts have been made in existing research. Early studies extracted distinct linguistic patterns in psycholinguistics as classification features (Pennebaker, Francis, & Booth, 2001; Tausczik & Pennebaker, 2010). But traditional feature engineering disregards the semantic comprehension of the input, hence limiting the accuracy of classification. In recent studies, models based on neural networks (Moreno, Gomez, Almanza-Ojeda, & Ibarra-Manzano, 2019; Rissola, Bahrainian, & Crestani, 2019) relieve the problem of semantic comprehension. But these methods require a substantial quantity of training data

[*] Corresponding author.
E-mail addresses: cszwen@comp.polyu.edu.hk (Z. Wen), csjcao@comp.polyu.edu.hk (J. Cao), cs-yu.yang@polyu.edu.hk (Y. Yang), 22042854G@connect.polyu.hk (H. Wang), rsong.yang@polyu.edu.hk (R. Yang), shuaiqi.liu@connect.polyu.hk (S. Liu).

https://doi.org/10.1016/j.ipm.2023.103422
Received 6 February 2023; Received in revised form 31 May 2023; Accepted 31 May 2023
Available online 20 June 2023

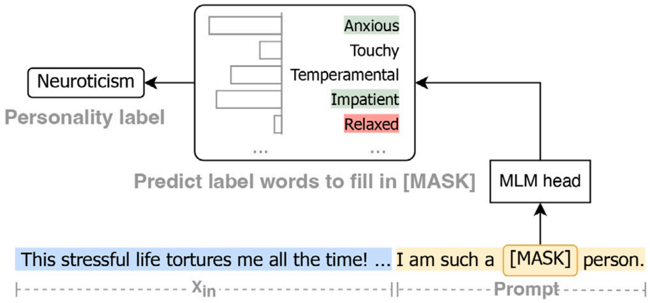
Neuroticism
Personality label
Anxious
Touchy
Temperamental
Impatient
Relaxed
...
...
Predict label words to fill in [MASK]
MLM head
This stressful life tortures me all the time! ... I am such a [MASK] person.
x_{in}
Prompt

Fig. 1. A toy example of DesPrompt. To recognize the personality in the input content x_{in}, DesPrompt utilizes a prompt describing the speaker to encapsulate x_{in}. The PLM predicts the most probable label words to fill in the masked position through a Masked Language Model (MLM) head. Then, the probability distribution of the label words is projected to the personality label.

with annotations. Due to the distinct benefits of pre-trained language models (PLMs) in natural language understanding, the current best practice is to fine-tune PLMs by incorporating particular modules for personality recognition (Jain, Kumar, & Beniwal, 2022; Jiang, Zhang & Choi, 2020; Jun, Peng, Changhui, Pengzheng, Shenke, & Kejia, 2021; Keh, Cheng, et al., 2019). While typically task-agnostic in architecture, this method requires task-specific fine-tuning with thousands or tens of thousands of examples (Brown et al., 2020).

Despite the abundance of text for personality recognition on social media, data collection for this purpose is often limited due to privacy concerns. Besides, due to the professional nature of personality analysis, obtaining accurate personality annotations typically relies on questionnaires, long-term observation, or specialized experiments. The shortage of annotated data hinders the performance of existing classification approaches that map semantic distributions to personality labels. Consequently, the question of how to accurately recognize personality with limited labeled data remains unresolved.

In this work, we propose DesPrompt to encapsulate the input text with personality-descriptive prompts and tune PLM for few-shot personality recognition, as shown in Fig. 1. DesPrompt is inspired by the lexical hypothesis (Galton, 1884) of the personality, which indicates that the major dimensions of personality are unraveled by the descriptive lexicons of human languages (Allport & Odbert, 1936; Cattell, 1943). Therefore, it is natural to abstract the personality implied in the input content with descriptive words and further map them to personality labels. Moreover, employing prompts to encapsulate the input text reformulates personality classification as a cloze-style word-filling task (Liu et al., 2023; Schick & Schütze, 2021a), which brings two strengths: (1) This word-filling task aligns with the pre-training Masked Language Modeling (MLM) task in most PLMs, hence enhancing the utilization of the pre-trained parameters for downstream personality recognition. (2) Besides, using prompts to fine-tune PLMs does not introduce extra parameters to learn; even with a limited amount of data, significant performance can be achieved.

Implementing our vision requires resolving two challenging issues that are not well solved by existing prompt-based studies: (1) finding precise and commonly used label words describing personality and (2) generating informative and general prompt content. First, a quantity of trait-descriptive adjectives for personality are terms from psychological analysis, which are inappropriate as label words for PLMs trained on the common corpus. Then, on the one hand, the prompt content should be informative, with specific words for each input to stimulate the PLM to generate personality-describing label words to fill in the prompt; on the other hand, the prompt content should be commonly suitable for various inputs. Existing prompt-based methods fail to solve these two challenging issues. Some approaches (Schick & Schütze, 2021a, 2021b) utilize hand-craft prompt content and verbalizer, which require expert knowledge and are laborious to find the optimal. Other methods automatically generate the verbalizer (Hu et al., 2022) and the prompt content (Gao, Fisch, & Chen, 2021) either overlook the prevalence of the label words or the informativeness in the prompt content.

To tackle the above challenges, we propose DesPrompt to automatically generate personality-descriptive label words and prompt contents to fine-tune PLM for few-shot personality recognition. First, DesPrompt expands trait-descriptive adjectives from psychology findings with a commonsense knowledge graph. Then, it weights each label word based on its relevance to the personality and its probability of being generated by the PLM. To generate the prompt content, DesPrompt learns the most probable context for all the label words by pre-finetuning (Gururangan et al., 2020) a T5 model, where the T5 model complements the sentences containing the label words by re-generating the context surrounding the label words in a self-supervised manner. Finally, the T5 model generates multiple prompts for each input and ensemble their results according to their coherence with the input.

We carried out comprehensive experiments on four datasets for personality recognition, spanning different scenarios such as daily conversations, self-report essays, Twitter posts, and Youtube comments. The results demonstrate that the DesPrompt approach exhibits substantial improvements compared to traditional fine-tuning techniques and competitive baseline methods, particularly in low-data regimes like zero-shot and few-shot scenarios. Our key contributions are summarized as follows:

- We introduce DesPrompt, a method for generating personality-descriptive prompts and fine-tuning PLM for efficient personality recognition with limited data.
- DesPrompt tackles two technical challenges: (1) finding precise and commonly used label words and (2) generating informative and general prompt content.

- Our experiments on four personality recognition datasets demonstrate DesPrompt's superiority over conventional fine-tuning and state-of-the-art prompt-based methods, particularly in zero-shot and few-shot scenarios.

The rest of this paper is organized as follows: We present a comprehensive literature review of related studies in Section 2. The studied problem and our DesPrompt model are presented in Section 3. We introduce our experiment settings, including the datasets, baseline models, and evaluation metrics, in Section 4. Then, we analyze the experiment results in Section 5. Finally, Section 6 concludes this paper and discusses our future work.

2. Related works

In this section, we review existing studies in text-based personality analysis and the prompt learning in Natural Language Processing.

2.1. Personality analysis

The study of personality recognition encourages a lot of applications personalized product recommendations and social media analysis (Chen, Yin, Li, Wang, Chen, & Chen, 2017; Roshchina, Cardiff, & Rosso, 2011; Tkalcic & Chen, 2015), partner matching on dating websites (Donnellan, Conger, & Bryant, 2004). Existing research in text-based personality analysis mainly focuses on self-reported essays (Pennebaker & King, 1999; Tighe, Ureta, Pollo, Cheng, & de Dios Bulos, 2016), behaviors in social media (Golbeck, Robles, & Turner, 2011; Schwartz et al., 2013; Yin, Zhang, & Liu, 2020), and daily conversations (Fang, Chen, Long, Xu, & Xiao, 2022; Jiang, Zhang et al., 2020; Mehl, Gosling, & Pennebaker, 2006; Rissola et al., 2019; Wen, Cao, Yang, Liu, & Shen, 2021). The scope of related research encompasses early-stage linguistic analysis through to the current classification method based on deep-learning models.

Research in the early stage focuses on finding statistical features for recognizing personality. Statistical word usage patterns and social behavior habits are highly correlated to personality traits. Early study (Mairesse & Walker, 2006) first used the RankBoost algorithm with non-linear statistical models to rank utterances with linguistic features for personality traits recognition. Then, Schwartz et al. (2013) statistically analyzed 700 million words, phrases, and topic instances collected from the Facebook messages of 75,000 volunteers and found striking variations in language with personality, gender, and age. Tighe et al. (2016) performed the Principal Component Analysis (PCA) on linguistic features from essays to classify the author's personality traits. Moreno et al. (2019) extract TF-IDF statistical features from Twitter blogs to identify the personality of Twitter users with PCA, Latent Dirichlet Allocation (LDA) model, and Non-negative matrix factorization models. However, although the shallow features are efficient in providing statistical differences for personality recognition, they fail in scenarios where the personality are identified by a deep understanding of the text content.

With the development of deep learning, neural network models are widely applied to recognizing personality through understanding text content. For microblog analysis, Yin et al. (2020) contribute to revealing the predictors of reposting negative information (RNI) on microblogs and by investigating the contingency role of personality. The Facebook posts are also studied in Lynn, Balasubramanian, and Schwartz (2020). They hierarchically encode all posts from one user with attention-based GRU (Cho et al., 2014) to produce the whole contextual representation for personality identification. Moreover, some researchers propose to combine emotional and semantic features for personality recognition (Ren, Shen, Diao, & Xu, 2021). Specifically, they leverage BERT to generate sentence-level embedding for text semantic extraction. Although deep neural networks have improved the performance of personality recognition in text, most models still require a large amount of data for training. Therefore, how to efficiently recognize personality with limited data remains an open problem.

2.2. Prompt-based learning

To parameter-efficiently utilize pre-trained language models, existing prompt-based learning studies work on generating better templates and verbalizers.

Firstly, Pattern-Exploiting Training (PET) is introduced as a semi-supervised training procedure that reformulates input examples as cloze-style phrases to help language models understand a given task (Schick & Schütze, 2021a, 2021b). However, the patterns (also called templates) that help to form the cloze-style phrases rely on human knowledge and are usually unstable to even slight changes. Then, researchers investigate automatically discovering better prompts. Several mining-based and paraphrasing-based methods are proposed to systematically generate diverse prompts to query specific pieces of relational knowledge in the LM Prompt and Query Archive (Jiang, Xu, Araki & Neubig, 2020; Wang, Xia, Wang, & Philip, 2022). Besides, AutoPrompt (Shin, Razeghi, Logan IV, Wallace, & Singh, 2020) creates prompts containing some trigger tokens for multiple tasks based on a gradient-guided search. Besides generating better templates, how to select appropriate words to fill in the cloze-style templates is also studied. It requires knowledge (*i.e.*, the verbalizer) of a task's labels and how they can best be expressed in natural language using single words (Schick, Schmid, & Schütze, 2020). AdaPrompt (Chen et al., 2022) makes use of knowledge in Natural Language Inference models for deriving adaptive verbalizers from limited label-related words. Besides, researchers also investigate incorporating external knowledge into the verbalizer (Hu et al., 2022). It is worth noting that LM-BFF (better few-shot fine-tuning of language models), a suite of simple and complementary techniques for fine-tuning language models on a small number of annotated examples, is proposed as a pipeline to automatic generate templates and verbalizers (Gao et al., 2021).

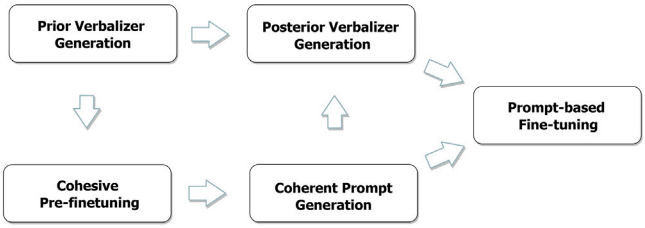
Prior Verbalizer Generation
Posterior Verbalizer Generation
Prompt-based Fine-tuning
Cohesive Pre-finetuning
Coherent Prompt Generation

Fig. 2. The overall workflow of DesPrompt.

3. Method

3.1. Problem statement and modeling

The problem under study is to classify the personality traits of a person based on their text inputs, such as conversations, self-report essays, or social media posts. Given an input text x_{in}, the objective is to determine the binary label y indicating the presence of each of the big-five personality traits (*i.e.*, Agreeableness, Conscientiousness, Extraversion, Openness, and Neuroticism, respectively). However, the problem is under the constraint where annotation data for training is limited, typically only tens or hundreds of instances.

We solve this problem by generating a personality-descriptive prompt to wrap x_{in}:

$$\mathcal{T}(x_{in}) = x_{in} \_\_ \ \texttt{[MASK]} \ \_\_.$$

as the input of a pre-trained masked language model $\mathcal{M}$. This prompting method requires two components: a prompt content $\mathcal{T}$ and a verbalizer f^y. The prompt content $\mathcal{T}$ includes the personality-descriptive contexts (represented by the blanks) and a [MASK] position for the label words to be filled in by $\mathcal{M}$. The verbalizer f^y that maps personality classification label y to a set of trait-descriptive label words $\mathcal{V}^y = \{v_1, v_2, \ldots\}$. It is worth noting that DesPrompt determines the big-five personality traits with five binary classification tasks sharing the same personality-descriptive prompt. In each binary personality classification task, we construct the verbalizer f^y including positive label words and negative label words with their relevance scores to the corresponding trait label y. To evaluate the quality of the label words, we also introduce a weight w_i to each label word v_i indicating both its relevance to the class label y and the familiarity to the PLM $\mathcal{M}$.

Formally, our problem is to maximize the probability $p(y|x_{in})$ by modifying the parameters θ in $\mathcal{M}$ and be further represented as maximizing the weighted sum of probabilities of all the label words $v_i \in \mathcal{V}^y$ filling in [MASK] when feeding $\mathcal{T}(x_{in})$ into $\mathcal{M}$:

$$\begin{aligned} &\arg\max_{\theta} \ p(y|x_{in}) \\ =&\arg\max_{\theta} \ p(\mathcal{V}^y|\mathcal{M}_\theta(\mathcal{T}(x_{in}))) \\ =&\arg\max_{\theta} \ \sum_{i=1}^{|\mathcal{V}^y|} p(\texttt{[MASK]} = v_i|\mathcal{M}_\theta(\mathcal{T}(x_{in}))) * w_i \end{aligned} \tag{1}$$

3.2. Overview of DesPrompt

To implement the above idea, we propose DesPrompt by automatically constructing personality-descriptive prompts and the verbalizer for fine-tuning the PLM. As shown in Fig. 2, DesPrompt has five main modules: prior verbalizer generation, cohesive pre-finetuning, coherent prompt generation, posterior verbalizer generation, and prompt-based fine-tuning.

In the Prior Verbalizer Generation, we first adopt trait-descriptive adjectives in psychology studies as the initial positive and negative label words. We further find the synonyms and antonyms for each label word and combine them together as the prior verbalizer. After we obtain the prior verbalizer, we conduct the Cohesive Pre-finetuning for a pre-trained T5 model to learn the appropriate context of the label words. Next, the pre-finetuned T5 model is used to generate prompt content coherent with the input x_{in} in Coherent Prompt Generation. The generated prompt content and the prior verbalizer are utilized to obtain the weight w_i for each label word facilitated by $\mathcal{M}$ in Posterior Verbalizer Generation. Finally, we conduct the Prompt-based Fine-tuning with the generated prompt content and the verbalizer for personality recognition with limited annotation data. In the following subsections, we will introduce each module in detail.

3.3. Prior verbalizer construction

On the basis of the lexical hypothesis in personality, we first construct a prior verbalizer f^y that maps personality classification label y to descriptive words $\mathcal{V}$ from psychology expand with lexical knowledge, as shown in Fig. 3.

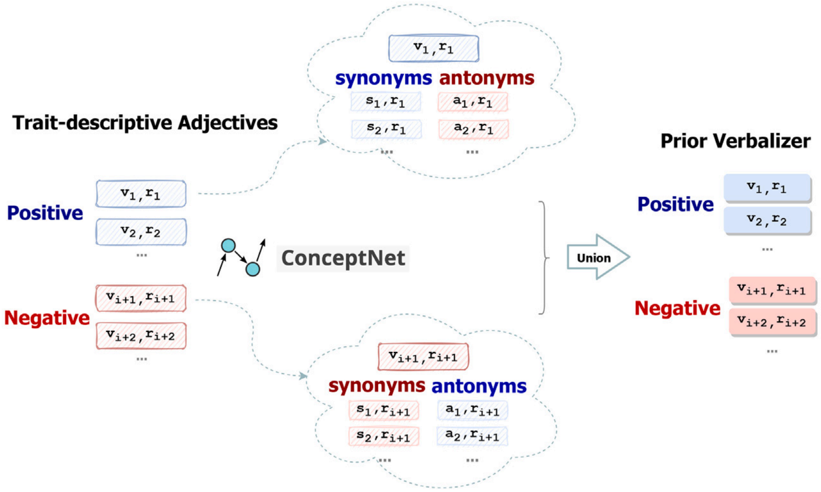

Fig. 3. Prior Verbalizer Generation. For each v_i, r_i is the relevance score to each personality trait. $s_1, s_2, \ldots$ are its synonyms, while $a_1, a_2, \ldots$ are its antonyms.

Table 1
The big-five personality traits and description (Costa & McCrae, 1992).

Factor	Description
Openness	Open-minded, flexible, nondogmatic
Conscientiousness	Scrupulous, well-organized.
Extraversion	The tendency to experience positive emotions.
Agreeableness	Trusting, sympathetic, and cooperative.
Neuroticism	The tendency to experience psychological distress.

The lexical hypothesis states that (1) the most distinctive, significant, and widespread phenotypic attributes tend to become encoded as single words in the conceptual reservoir of language, and (2) the degree of representation of an attribute in languages tends to correspond to the relative importance of the attribute (Goldberg, 1995). Therefore, it is natural to abstract the personality implied in the input content with descriptive words and further map them to the personality labels.

We use the binary values on the big-five personality traits as the classification labels (*i.e.*, positive and negative). The big-five trait theory presents a discrete taxonomy of personality as shown in Table 1. It is developed from the trait theory and the lexical hypothesis in psychology. It is also widely applied as personality classification labels in social media content (Iacobelli, Gill, Nowson, & Oberlander, 2011; Souri, Hosseinpour, & Rahmani, 2018) and conversations (Mairesse & Walker, 2006; Mairesse, Walker, Mehl, & Moore, 2007).

As for the label words, it is suggested that a representative sampling of personality-descriptive terms, especially adjectives, might yield a representative sampling of personality attributes (Saucier & Goldberg, 1996). Therefore, we collect 435 adjectives $\{v_1, v_2, \ldots\}$ from psychology analysis (Saucier & Goldberg, 1996), where each adjective has relevance scores r_i ranging from −1 to 1 to the big-five personality traits. In this research, eight men and 17 women psychology undergraduate students in a US university rated the frequencies of the adjectives and select 435 adjectives often used to describe people. Then, these adjectives are used by the students to rate both themselves and other peers to obtain the correlation between the adjectives and the big-five personality traits. For each trait, the adjectives with positive relevance scores consist of the label words of the positive class and vice versa.

Considering some of the adjectives may accurately describe the personality but are rare in daily usage (*e.g.*, introverted, exhibitionistic), these words are not prevalent in the general corpus pre-training the PLM. To increase the probability of the label words being predicted by the PLM, we find top-n synonyms and antonyms for each label word from the ConceptNet (Speer, Chin, & Havasi, 2017) for expansion, n is a hyperparameter. ConceptNet (Speer et al., 2017) is a freely-available semantic network that connects words and phrases of natural language (terms) with labeled, weighted edges (assertions). It is also the knowledge graph version of the Open Mind Common Sense project (Singh et al., 2002), a common sense knowledge base of the most basic things a person knows. We choose n to be 10 and show the numbers of label words before and after the expansion in Table 2. It is worth noting that we reduce the duplication of the synonyms and antonyms among different adjectives. So, the numbers of expanded label words are not necessarily 10 times of the numbers before expansion.

We also integrate the relevance scores for the expanding label words. For example, synonyms for label word v_i keep the same scores, while the antonyms for v_i have negative relevance scores:

$$\begin{aligned} \mathcal{V}^{pos} &= \{(v_1^{pos}, r_1^{pos}), (v_2^{pos}, r_2^{pos}), \ldots\}; \\ \mathcal{V}^{neg} &= \{(v_1^{neg}, r_1^{neg}), (v_2^{neg}, r_2^{neg}), \ldots\} \end{aligned} \tag{2}$$

Table 2
The numbers of label words before and after expansion.

Personality Traits	Before Expansion			After Expansion		
	Positive	Negative	Total	Positive	Negative	Total
AGR	225	204	429	765	775	1540
CON	209	216	425	788	742	1530
EXT	213	208	421	791	718	1509
OPN	249	174	423	724	790	1514
NEU	199	224	423	868	649	1517

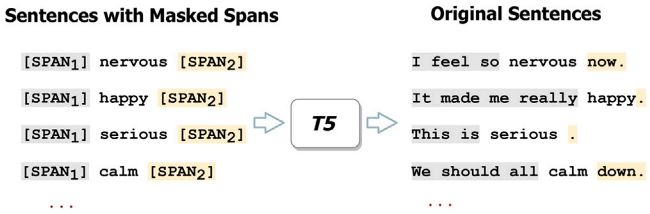

Fig. 4. Cohesive Pre-finetuning. The T5 model is pre-finetuned to fill in the masked spans and reconstruct the original sentences to learn the words surrounding the label words in daily usage.

3.4. Cohesive and coherent prompt generation

After obtaining the prior verbalizer, we generate the prompt content $\mathcal{T}$ using the label words. To ensure interpretability and align with the pre-training MLM task, we propose two linguistic characteristics for the prompts:

- **Cohesive**: The prompt must be grammatically cohesive with the given label words, allowing for similarity to the pre-training MLM task on natural input samples.
- **Coherent**: The prompt must be semantically coherent with the input x_{in}, enabling the PLM to recognize the input and prompt content as a complete sentence and use the input as context information when filling in the masked label words.

Manually generating prompts satisfying the characteristics above is labor extensive. Besides, it is difficult to find the optimal because the performance of different manual-designed prompts varies a lot. Inspired by LM-BFF (Gao et al., 2021), we quantitively model the two characteristics by automatically generating prompts with a pre-trained T5 model adapted to the personality-related corpus.

3.4.1. Cohesive pre-finetuning

The T5 model (Raffel et al., 2020) (*i.e.*, a large pre-trained text-to-text Transformer) is pre-trained by filling the masked spans given partial words in the original input sentences. For example, in pre-training, if we have an original sentence as *Thank you for inviting me to your party last week.*; the input of T5 is *Thank you [X] me to your party [Y] week.*; T5 is trained to generate *for inviting* to fill in *[X]* and *last* for *[Y]*.[1] Similar, the prompt content generation is to predict appropriate context words surrounding the given label words. For example, if the input is *This stressful life tortures me all the time!* $[\mathtt{SPAN}_1]$ *anxious* $[\mathtt{SPAN}_2]$, the prompt content generation is expected to generate *I am such a* for $[\mathtt{SPAN}_1]$ and *person.* for $[\mathtt{SPAN}_2]$.

So, it is natural to predict appropriate context words surrounding the given label words in the prompt content. However, directly using T5 for prompt content generation (Gao et al., 2021) merely uses the pre-trained language modeling ability on the general corpus, which may be insufficient for personality analysis on specific content. Thus, we conduct a pre-finetuning (Aghajanyan et al., 2021; Gururangan et al., 2020) on the T5 with the sentences containing the label words from the personality-related corpus, as shown in Fig. 4. By doing so, we aim to supervise the T5 to generate prompt content, including the context **cohesive** to the label words.

We first show how to construct the samples to support the cohesive pre-finetuning. For the two classification labels $\{pos, neg\} \in Y$, we union the sets of label words of each class and obtain the overall label word set:

$$\mathcal{V}^{all} = \mathcal{V}^{pos} \cup \mathcal{V}^{neg}$$

Then, for each $v_i \in \mathcal{V}^{all}$, we retrieve all the sentences $\{sent_1, sent_2, \dots\}$ containing v_i from a personality-related corpus (*i.e.*, the training datasets and other personality-descriptive online posts) and construct the self-supervise training sample $sample_j$ from each $sent_j$ by:

$$(sample_j, label_j) = ([\mathtt{SPAN}_1]\ v_i\ [\mathtt{SPAN}_2], sent_j)$$

[1] This example is shown in the original paper (Raffel et al., 2020) introducing T5.

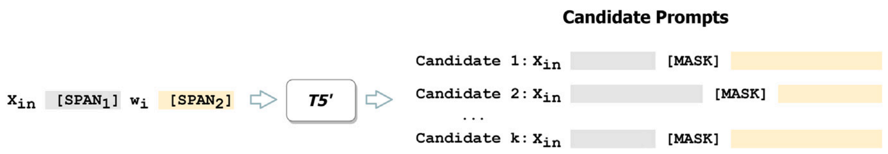

Fig. 5. Coherent Prompt Generation. The $T5'$ is the pre-finetuned T5 model from Cohesive Pre-finetuning, and the w_i is each label words corresponding to X_{in}.

where we mask the context before and after v_i in the original $sent_j$ with [SPAN$_1$] and [SPAN$_2$], respectively. The training set S for pre-finetuning is then unionized from the samples for all the v_i in $\mathcal{V}^{all}$.

Next, we conduct the pre-finetuning on a T5 model by feeding all the $sample_j \in S$ into the encoder and supervising the model to auto-regressively generate the content to fill in [SPAN$_1$] and [SPAN$_2$] with the decoder. This process can be represented as minimizing the cross-entropy loss of predicting every token in the original $sent_j$:

$$\arg\min_{\theta} \sum_{j=1}^{|S|}\sum_{k=1}^{|T|} CE(T5_{\theta}[\mathcal{D}(t_1,\dots,t_{k-1});\mathcal{E}(sample_j)],t'_k) \tag{3}$$

where θ is the parameter in T5 and $T5_{\theta}[]$ means the logit of T5 model predicting the kth token t_k, t'_k is the ground-truth token in $sent_j$. CE means the cross-entropy loss.

After the process above, we obtain the pre-finetuned T5 model for prompt content generation.

3.4.2. Coherent prompt generation

To generate informative and general prompts that stimulate the PLM to express the implied personality in the input, we feed all training samples x_{in} together with their corresponding label words to the pre-finetuned T5 model, as shown in Fig. 5. The generated prompt content is designed to be **coherent** with x_{in}.

Formally, given a input (x_{in}, y) and one of its corresponding label words $w_i \in \mathcal{V}^y$, we construct the input

$$\mathcal{T}(x_{in}, w_i) = x_{in}\ \texttt{[SPAN}_1\texttt{]}\ w_i\ \texttt{[SPAN}_2\texttt{]}$$

for the T5-encoder, and let T5-decoder to generate the content in the masked [SPAN$_1$] and [SPAN$_2$]. The log likelihood of the prompts $\mathcal{T}$ is calculated by:

$$P_{\mathcal{T}}(x_{in}, w_i) = \sum_{j=1}^{|\mathcal{T}|} log[P_{T5}(t_j|\mathcal{D}(t_1,\dots,t_{j-1});\mathcal{E}(\mathcal{T}(x_{in}, w_i)))] \tag{4}$$

where P_{T5} is the output probability distribution of pre-trained T5 model, $\mathcal{D}()$ means $t_1,\dots,t_{j-1}$ are the input of the auto-regressive T5-decoder, while $\mathcal{E}()$ means the $\mathcal{T}(x_{in}, w_i)$ are for the T5-encoder.

Since we have multiple label words for each input x_{in}, and the prompt content should be suitable for all the input samples, the prompt generation is to maximize the overall log-likelihood of the prompt content as:

$$P_{\mathcal{T}} = \frac{1}{|\mathcal{V}^y|}\sum_{(x_{in},y)\in\mathcal{D}_{tr}}\sum_{i=1}^{|\mathcal{V}^y|} P_{\mathcal{T}}(x_{in}, w_i) \tag{5}$$

$\mathcal{D}_{tr}$ is the training set.

As there is no ground truth for what prompt content is the most suitable for the input, generating the prompt content here is an unsupervised process where the parameters in T5 are fixed. The ability of the T5 model to generate the prompt content is ensured by the previous cohesive pre-finetuning.

To ensure the generality of the prompt content (*i.e.*, the prompt content is suitable for various inputs), we use beam search to decode c prompt candidate set $T = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$, and ensemble their results following the settings in previous works (Gao et al., 2021; Schick & Schütze, 2021a).

3.5. Posterior verbalizer generation

The label words and their relevance to the personality labels in the prior verbalizer are only based on psychology and lexical knowledge. We still do not know if the label words are suitable for being filled in the prompts with the input x_{in} by the PLM. Therefore, we conduct a posterior verbalizer generation by weighting and selecting the label words by the sum of the relevance score and the probability to be filled in by the PLM, as shown in Fig. 6.

Specifically, for each label word v_j and a candidate prompt $\mathcal{T}_i \in T$, we obtain the logit of v_j by filling it into the [MASK] in $\mathcal{T}_i$ with $\mathcal{M}$. Then, we normalize the average of the logits from all the prompts in T and obtain the probability p_j of v_j through a

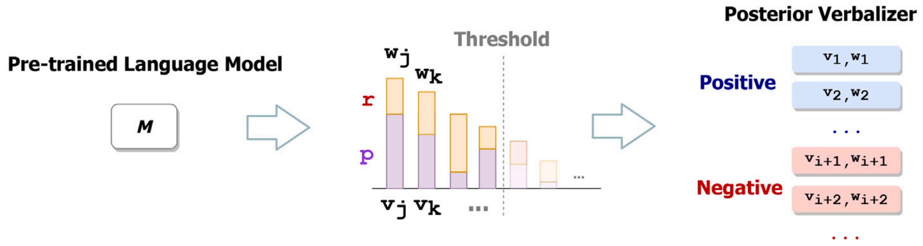

Fig. 6. Posterior Verbalizer Generation. p is the probability of generating each word v by M.

softmax function:

$$logit_j = \frac{1}{|T|}\sum_{i=1}^{|T|}\mathcal{M}(\mathcal{T}_i, v_j)$$
$$p_j = \frac{e^{logit_j}}{\sum_{i=0}^{|\mathcal{V}|} e^{logit_i}} \tag{6}$$

After we got the probability p_j for v_j, we integrate p_j and the prior relevance score r_j together with a normalized sum:

$$w_j = \frac{(p_j + r_j)}{\sum_{i=0}^{|\mathcal{V}|}(p_i + r_i)} \tag{7}$$

The weight measures both the relevance to the personality label and the familiarity with the PLM, which are the two aspects we consider good label words should have. Therefore, we also remove the label words with weights lower than a manually set threshold to refine the posterior verbalizer to eliminate possible biased prediction.

3.6. Prompt-based fine-tuning

After we obtain the prompt candidate set $T = \{\mathcal{T}_1, \mathcal{T}_2, \ldots\}$, and the posterior verbalizer $\mathcal{V}$. We ensemble the results of all the prompt content for each input x_{in}. The prompt-based fine-tuning process is to adjusting the parameters θ in $\mathcal{M}$ to maximize the probability of predicting the label y of x_{in}:

$$P(y|x_{in}) = \frac{1}{|T|}\sum_{j=1}^{|T|}\sum_{i=1}^{|\mathcal{V}^y|}\mathcal{M}_\theta(\mathcal{T}_j(x_{in}, v_i^y)) * w_i \tag{8}$$

In DesPrompt, we reformulate the classification task as a word-filling task, which aligns with the MLM task in pre-training PLMs. RoBERTa is only pre-trained by the MLM task, and the performance of RoBERTa is widely validated in many Natural Language Understanding tasks. So, we choose it as our subsequent PLM for the prompt-based fine-tuning.

4. Experiment settings

To evaluate the personality recognition performance of DesPrompt, we design extensive experiments on four various personality analysis datasets. We will introduce the datasets, baseline approaches, and evaluation settings as below.

4.1. Datasets

FriendsPersona (Jiang, Zhang et al., 2020) is dialog script dataset developed upon the public Friends TV Show.[2] It contains 711 conversations including 8,157 utterances. Each conversation is annotated by the binary Big Five personality traits of a specified speaker.

Essays (Pennebaker & King, 1999) is the benchmark dataset for text-based personality recognition with 2,468 self-report essays from more than 1,200 students with binary Big Five personality annotations.

myPersonality[3] is an open-source dataset that collects 50 statuses for each of the 250 Facebook users. The personality annotations for each user are obtained by a binary classification model for each personality trait trained on a dataset from the myPersonality project.[4]

[2] https://en.wikipedia.org/wiki/Friends

[3] https://github.com/jcl132/personality-prediction-from-text/tree/master/data/myPersonality

[4] https://sites.google.com/michalkosinski.com/mypersonality

Table 3
Basic statistics and label distributions (positive : negative) of the four datasets.

	FriendsPersona	Essays	myPersonality	PAN-AP-2015
Type	Conversation	Self-report essays	Facebook posts	Twitter posts
#Samples	711	2,467	425	658
Avg. length	48.30	662.40	321.48	464.05
AGR	0.43:0.57	0.47:0.53	0.47:0.53	0.46:0.54
CON	0.46:0.54	0.49:0.51	0.47:0.53	0.48:0.52
EXT	0.44:0.56	0.51:0.49	0.41:0.59	0.49:0.51
OPN	0.35:0.65	0.49:0.51	0.29:0.71	0.32:0.68
NEU	0.47:0.53	0.50:0.50	0.39:0.61	0.42:0.58

PAN-AP-2015 (Rangel et al., 2015) contains personality annotations for 294 twitter users and their twitter content. Personality traits were selfassessed with the BFI-10 online test (Rammstedt & John, 2007) and reported as scores normalized between −0.5 and +0.5. We set thresholds for five traits to obtain the labels of positive and negative classes, as introduced below.

To obtain the binary labels for each trait in PAN-AP-2015, we need a threshold to split the samples by their personality scores to get positive and negative samples. We first enumerated the split points between $[-0.5, +0.5]$ with the step of 0.1 as the threshold. For each threshold, we manually check by keeping a relatively balanced positive:negative rate for all the traits. This is for balanced classification. More importantly, the statistics from the other three datasets show that a balanced positive: negative rate is the common sample distribution in each trait.

After we select the appropriate threshold for each personality trait, we manually adjust the inaccurate labels of the samples with personality scores near the selected threshold. To be specific, for all the samples with personality scores higher and lower than the threshold by 0.2, we manually check the labels of the samples and adjust them if they have the wrong labels. The adjusting results were obtained by voting from five computer science students and final verification by another student in psychology.

We preprocess the datasets by anonymizing the user (or speaker) names and locations, removing the weblinks and constant repeat content. Especially, each data sample in **myPersonality** and **PAN-AP-2015** is a post list containing multiple posts for one user. To avoid the sample length being too long due to concatenating all the posts in each sample, we split the post list of one user into several sub-lists as multiple samples with the same personality annotation so that the total length of each sample is less than 512. After preprocessing the input content, we calculate the label distributions for all the datasets, the overall dataset information is shown in Table 3.

As we can see, the sample number among the four datasets ranges from 425 to 2,467. Such data amounts are insufficient to train neural network models, not to mention collecting and annotating these data are laborious and time-consuming. The longest average length of samples is in the **Essays**, while the shortest one is for the dialog content in **FriendsPersona**. Except for the obvious imbalance in **OPN** among **FriendsPersona**, **myPersonality**, and **PAN-AP-2015**, other personality trait labels are around equally distributed in the datasets with a slight bias of having more negative samples.

4.2. Baseline methods

We compare DesPrompt with the following state-of-the-art models in the personality recognition task. The baselines we choose are in two categories: traditionally fine-tuning PLM (**Fine-tune**) and prompt-based methods (**PET** Schick & Schütze, 2021a, **LM-BFF** Gao et al., 2021, and **KPT** Hu et al., 2022), we will introduce the baseline methods as below:

Fine-tune: Fine-tune refers to the conventional fine-tuning approaches by fine-tuning the pre-trained language model with an additional classification head to classify x_{in}. A lot of existing approaches follow this setting but with different back-bone PLM (*e.g.*, fine-tuning BERT Jun et al., 2021, fine-tuning RoBERTa Jiang, Zhang et al., 2020). We use this baseline to represent them and evaluate whether DesPrompt is better than the traditional fine-tuning.

PET: The pioneer prompt-tuning method uses the class name as the only label word for each class and manually defined prompt template. However, the class names for big-five personality traits are the nouns (in Table 1) without positive and negative label words, so we use the adjectives describing the personality traits (Roccas, Sagiv, Schwartz, & Knafo, 2002) of the big five personality traits as the label words.

LM-BFF: LM-BFF is a pipeline for prompt-based fine-tuning with automating the prompt generation and a refined strategy for dynamically and selectively incorporating demonstrations into each context.

KPT: KPT stands for Knowledgeable prompt-tuning, which incorporates external knowledge into the verbalizer to improve and stabilize prompt-based fine-tuning. Here, we follow the procedure of KPT to construct its verbalizer with the label words from the prior verbalizer described in Section 3.2.

All above baseline models except the Vanilla Transformer use the RoBERTa-large as the backbone model in our DesPrompt.

4.3. Evaluation metrics

We use the F-score of binary classification on each trait for evaluation. To further validate DesPrompt in limited data, which is more often in the real application, we conduct experiments in full data, few-shots, and zero-shot scenarios. For full data scenarios, we split each dataset in around 8:1:1 for train, validation, and test sets. For few-shot scenarios, we sample 1, 5, 10, and 20 instances respectively for each class from the original train sets. For zero-shot scenarios, we only use the test sets for evaluation.

To ensure the reliability of results, we run each experiment 10 times with different random seeds and report the average performances and the standard deviations, where the random seeds are used to split the datasets and initialize the model parameters.

4.4. Implementation details

During implementation, we used the T5-large model as the underlying model for prompt content generation and RoBERTa-large as our PLM M. Both models were obtained from Hugging Face.[5]

To pre-finetune T5, we collected 41,741 unique utterances from the training sets of four datasets and other online posts to generate training samples. In this process, all utterances were padded to 20 tokens and grouped into batches of 16. The T5 model was trained for 10 epochs with a fixed learning rate of 0.0001.

In the prompt-based fine-tuning process for RoBERTa, we set the learning rate to 0.0001 and trained for 3 epochs. The number of tokens for utterances in the FriendsPersona, Essay, MyPersonality, and PAN-AP-2015 datasets was padded to 123, 512, 512, and 512, respectively. For full data training, the utterances were batched in eights, while in few-shot settings, the utterances were fed to DesPrompt one by one during training.

5. Results and analysis

In this section, we report and analyze the experimental results of personality recognition by answering the following **R**esearch **Q**uestions:

RQ1: Can DesPrompt relieve the limitation of annotated data in personality recognition?
RQ2: What is the performance of DesPrompt recognizing specific personality traits?
RQ3: Does DesPrompt generate better verbalizer and prompt content?

5.1. Can DesPrompt relieve the limitation of annotated data in personality recognition?

We answer this research question by conducting zero-shot and few-shot experiments on personality recognition. The results suggest two statements: (1) When training data is limited, DesPrompt considerably outperforms other methods, especially training neural networks and traditional fine-tuning; the less data required, the more obvious the strength is. (2) 5–10 annotated samples in each class are adequate for DesPrompt to fine-tune the PLM to be comparable to other methods with full data. We will explain the two statements in detail.

We first report zero-shot and few-shot results (F-scores) of personality recognition the Table 4. In general, our approach (*i.e.*, DesPrompt) achieves the best performances in 16 of 20 experiment settings among the four datasets. It is worth mentioning that our approach largely outperforms all the baseline models in zero-shot (**34.9%**) and one-shot (**21.5%**) scenarios. It indicates that DesPrompt effectively introduces prior knowledge to eliminate the annotation shortage in fine-tuning.

We also draw the line plots of the results in Table 4, including the full data results in Fig. 7 to show the variations of all the methods as the data amount increase. We can see that basically, all the methods will perform better with more training data, but the prompt-based approaches (Our method, PET, LM-BFF, and KPT) increase decelerates when the data samples are more than 5. Moreover, their performance with 5 or 10-shot performance is comparable with the Fine-tune with full data.

Besides, in myPersonality, the performance of Fine-tune continues to improve as the number of training samples exceeds 20. However, different degrees of decline is observed in PET, LM-BFF, and KPT. The reason might be the conflict between prior modeling ability in PLM and the supervision of the data. However, DesPrompt keeps relatively stable in this situation. We postulate that the superiority may be attributed to the cohesive pre-finetuning process and the posterior verbalizer generation in our proposed method, which helps the PLM process more familiar prompt content and label words.

As the most data-require module in DesPrompt (*i.e.*, the cohesive pre-finetuning) is in a self-supervised manner, the limitation of data annotations does not constrain its performance. Therefore, DesPrompts can robustly fine-tune the PLM with only 5 to 10 annotated samples for each class.

5.2. What is the performance of DesPrompt recognizing specific personality traits?

After verifying the superiority of DesPrompt in few-shot personality recognition, we focus on its performance on specific personality traits.

To comprehensively illustrate the performance, we conduct experiments under two settings: zero-shot and full-data training. Besides, to show the significance of the outperformance of DesPrompt, we conduct Welch's t-test (Welch, 1947) between DesPrompt and all the baseline models on the results with 10 random seeds. The Welch's t-test, also known as Welch's unequal variances t-test, is a statistical test used to compare the means of two independent groups when the assumption of equal variances is violated. It is a

[5] https://huggingface.co/

Table 4
Personality recognition results (F-scores) on few-shot settings.

Dataset	Method	0	1	5	10	20
FriendsPersona	Fine-tune	0.451	0.392	0.598	0.616	0.670
	PET	0.368	0.594	0.660	**0.676**	0.673
	LM-BFF	0.438	0.636	0.654	0.668	0.671
	KPT	0.466	0.522	0.669	0.671	0.673
	DesPrompt	**0.576**	**0.656**	**0.680**	0.665	**0.678**
Eassy	Fine-tune	0.429	0.351	0.583	0.677	0.678
	PET	0.374	0.605	0.663	0.673	0.677
	LM-BFF	0.476	0.526	0.670	0.676	0.677
	KPT	0.533	0.570	0.647	**0.678**	0.677
	DesPrompt	**0.566**	**0.632**	**0.668**	0.658	**0.678**
myPersonality	Fine-tune	0.428	0.503	0.577	0.590	0.622
	PET	0.384	0.626	0.659	0.657	0.663
	LM-BFF	0.471	0.428	0.646	0.655	0.664
	KPT	0.505	0.541	0.656	0.657	**0.666**
	DesPrompt	**0.625**	**0.628**	**0.665**	**0.670**	0.659
PAN-AP-2015	Fine-tune	0.440	0.562	0.654	0.650	0.707
	PET	0.423	0.613	0.685	0.704	0.701
	LM-BFF	0.521	0.455	**0.706**	0.704	0.707
	KPT	0.523	0.597	0.694	0.704	0.708
	DesPrompt	**0.672**	**0.672**	0.682	**0.706**	**0.711**

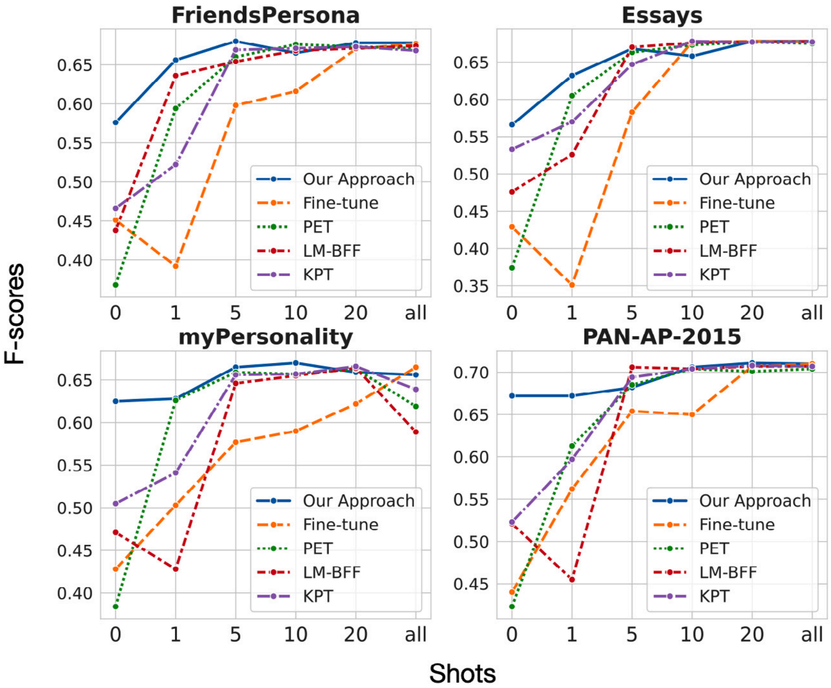

Fig. 7. Personality recognition performance variations along with the amount of training data.

modification of the traditional Student's t-test that allows for unequal variances between the groups being compared. We employed Welch's t-test to assess the statistical significance of DesPrompt's superior performance over other baseline models. Since these results were derived from different models, we cannot guarantee that their variances are equal or similar. Therefore, we opted for Welch's t-test instead of the standard t-test. The results are shown in Table 5 and Table 6, respectively. In both tables, each F-score is averaged from the results with 10 random seeds. The P-value is obtained by calculating Welch's t-test between each baseline and DesPrompt on the 10 results. The green results indicate that DesPrompt significantly outperforms the corresponding baselines with $p < 0.05$.

We answer the research question in two aspects: (1) Directly applying our method to new datasets, DesPrompt significantly outperforms other baseline methods in recognizing most personality traits. (2) Even with adequate training data, DesPrompt

Table 5
Zero-shot personality recognition results (F-scores and P values) on the big five personality traits.

Dataset	Method	AGR	CON	EXT	OPN	NEU	Average
FriendsPersona	Fine-tune	0.487 0.03	0.421 0.93	0.415 0.03	0.497 0.03	0.436 0.00	0.451 0.04
	PET	0.643 0.00	0.404 0.04	0.000 0.00	0.163 0.00	0.629 0.04	0.368 0.00
	LM-BFF	0.473 0.00	**0.455** 0.76	0.404 0.00	0.591 0.00	0.420 0.00	0.471 0.00
	KPT	0.509 0.00	0.443 0.32	0.369 0.00	0.517 0.05	0.494 0.00	0.466 0.00
	DesPrompt	**0.703**	0.412	**0.535**	**0.599**	**0.631**	**0.576**
Essay	Fine-tune	0.436 0.05	0.428 0.04	0.432 0.05	0.423 0.01	0.426 0.05	0.429 0.03
	PET	0.605 0.00	0.453 0.56	0.052 0.00	0.247 0.00	0.562 0.00	0.384 0.00
	LM-BFF	0.486 0.00	**0.516** 0.01	0.435 0.00	0.482 0.80	0.463 0.00	0.476 0.00
	KPT	0.511 0.00	0.480 0.42	0.481 0.05	**0.582** 0.00	0.469 0.00	0.505 0.00
	DesPrompt	**0.638**	0.473	**0.594**	0.478	**0.649**	**0.566**
MyPersonality	Fine-tune	0.452 0.05	0.402 0.02	0.385 0.05	0.535 0.05	0.367 0.03	0.428 0.03
	PET	0.605 0.00	0.453 0.00	0.052 0.00	0.247 0.00	**0.562** 0.01	0.384 0.00
	LM-BFF	0.473 0.00	0.455 0.00	0.404 0.00	0.601 0.00	0.42 0.00	0.471 0.00
	KPT	0.511 0.00	0.48 0.00	0.481 0.00	0.582 0.00	0.469 0.18	0.505 0.00
	DesPrompt	**0.695**	**0.594**	**0.587**	**0.735**	0.512	**0.625**
PAN-AP-2015	Fine-tune	0.381 0.03	0.434 0.05	0.432 0.04	0.484 0.04	0.467 0.03	0.440 0.04
	PET	0.544 0.00	0.503 0.00	0.0 0.00	0.380 0.00	**0.689** 0.00	0.423 0.00
	LM-BFF	0.532 0.00	0.507 0.00	0.424 0.00	0.570 0.00	0.571 0.04	0.521 0.00
	KPT	0.387 0.00	0.505 0.00	0.584 0.00	0.546 0.00	0.593 0.48	0.523 0.00
	DesPrompt	**0.626**	**0.663**	**0.678**	**0.781**	0.612	**0.672**

maintains its superiority over other prompt-based methods in recognizing most traits and occasionally even outperforms the traditional fine-tuning approach. We will explain the answers in detail.

We first report the zero-shot personality recognition results (with F-scores and P values) on the big five personality traits in Table 5. In average results, DesPrompt significantly outperforms all other baselines on all four datasets. More specifically, DesPrompt achieves the best results and significantly outperforms all other baselines on 15 of 20 single trait recognition. These results show the superiority of DesPrompt in data limitation, where our method could efficiently utilize the prior knowledge to recognize specific personality traits. We further investigate the reasons for the outperformance. As there is no supervision in the zero-shot scenario, compared with Fine-tune, DesPrompt reformulates the classification task into the word-filling task, so that efficiently utilizes the pre-trained parameters. Compared with other prompt-based methods, DesPrompt better utilizes prior knowledge to construct the verbalizer and the prompt content. This is also analyzed in detail in Section 5.3.

Then, we report full-data training personality recognition results in Table 6. DesPrompt achieves the best performances on all 4 datasets. More specifically, DesPrompt achieves the best performances in 14 over 20 personality trait recognition results. Across all trait recognition results, DesPrompt either significantly outperforms the traditional fine-tuning method or performs comparably to it. In six of the single trait results, DesPrompt outperforms all prompt-based baselines with significant differences. Moreover, in the majority (18/20) of the results, DesPrompt performs significantly better than at least one baseline model; in the remaining two results (Recognizing EXT in Essay and PAN-AP-2015), DesPrompt is also competitive with the best-performing results.

We then analyze the results of the full data training. In general, traditional fine-tuning introduces additional modules for classification, which results in underfitting in few-shot scenarios but works better than prompt-based methods when data is adequate to train the parameters. However, we found it even without additional classification modules, DesPrompt can also be competitive with the traditional fine-tuning method and maintains its superiority over other prompt-based methods in recognizing most traits.

Table 6
Personality recognition results (F-scores and P values) on the big five personality traits with full-data training.

Dataset	Method	AGR	CON	EXT	OPN	NEU	Average
FriendsPersona	Fine-tune	0.724 0.01	0.624 0.00	0.611 0.36	0.790 0.03	0.636 0.00	0.677 0.05
	PET	0.712 0.02	0.625 0.39	0.595 0.04	0.787 0.03	0.636 0.00	0.671 0.03
	LM-BFF	0.723 0.01	**0.628** 0.00	0.589 0.02	0.790 0.99	0.637 0.29	0.673 0.03
	KPT	0.723 0.03	0.620 0.04	0.573 0.04	0.790 0.99	0.633 0.04	0.668 0.01
	DesPrompt	**0.728**	0.622	**0.613**	**0.790**	**0.642**	**0.679**
Essay	Fine-tune	0.693 0.98	0.671 0.07	0.683 0.47	0.678 0.20	0.661 0.77	0.677 0.04
	PET	0.693 0.92	0.672 0.58	0.683 0.53	0.670 0.01	0.657 0.02	0.675 0.00
	LM-BFF	0.693 0.31	0.671 0.03	**0.684** 0.17	0.679 0.33	**0.665** 0.22	0.678 0.97
	KPT	0.690 0.02	0.670 0.04	0.682 0.60	0.679 0.03	0.663 0.63	0.677 0.04
	DesPrompt	**0.693**	**0.673**	0.682	**0.682**	0.662	**0.678**
MyPersonality	Fine-tune	0.690 0.01	0.633 0.44	0.590 0.01	0.817 0.00	0.567 0.04	0.659 0.02
	PET	0.697 0.11	0.603 0.05	0.506 0.00	0.838 0.98	0.453 0.00	0.619 0.00
	LM-BFF	0.675 0.03	0.631 0.04	0.469 0.01	0.838 0.98	0.331 0.03	0.589 0.00
	KPT	0.698 0.01	0.630 0.02	0.560 0.02	0.838 0.98	0.469 0.01	0.639 0.05
	DesPrompt	**0.705**	**0.637**	**0.592**	**0.838**	**0.571**	**0.669**
PAN-AP-2015	Fine-tune	**0.626** 0.57	**0.694** 0.97	0.673 0.12	**0.812** 0.34	0.739 0.46	0.709 0.44
	PET	0.617 0.05	0.682 0.02	0.677 0.40	0.806 0.03	0.740 0.34	0.704 0.03
	LM-BFF	0.618 0.05	0.692 0.82	0.672 0.31	0.811 0.00	0.740 0.34	0.707 0.01
	KPT	0.623 0.98	0.692 0.82	0.669 0.17	0.808 0.05	0.742 0.00	0.707 0.02
	DesPrompt	0.623	0.693	**0.682**	0.811	**0.743**	**0.710**

It suggests that the task form of DesPrompt can match the advantages of additional trainable parameters in Fine-tune. Besides, compared to other prompt-based methods, DesPrompt introduces richer prior knowledge than other methods, so it can achieve better performance in full data training.

5.3. Does DesPrompt generate better verbalizer and prompt content?

After evaluating the performance of DesPrompt, we investigate the reasons why DesPrompt works better than other prompt-based methods both quantitatively and qualitatively. We conduct an ablation study and conclude with some intuitive explanations to answer this research question: (1) The verbalizer of DesPrompt contains distinguishable and commonly used label words to contribute to personality recognition. (2) DesPrompt generates more informative and diverse prompts for the ensemble in personality recognition.

We first introduce the setting of our ablation study. To evaluate the effectiveness of the verbalizer of DesPrompt, we respectively use (1) the original 435 adjectives in psychology knowledge (Psychology Knowledge), (2) the label words from the prior verbalizer expanded with the ConceptNet (Prior Verbalizer), and (3) the label words from the posterior verbalizer refined in DesPrompt (DesPrompt) as the verbalizer. To evaluate the quality of the prompt content, we compare the performance between the prompts generated by the Vanilla T5 model (Vanilla Prompts) and the pre-finetuned T5 in DesPrompt. In order to maintain fairness in our comparisons, we ensure consistency by employing identical prompt content when evaluating different verbalizers and utilizing the same verbalizers when comparing different prompt content. The numeric results are shown in Table 7.

By comparing Psychology Knowledge, Prior Verbalizer, and DesPrompt, we found that directly using the adjectives in psychology knowledge obtains the lowest results, while the Prior Verbalizer enhanced by synonyms and antonyms significantly improves the

Table 7
Personality recognition results (F-scores) of the ablation study on few-shot settings.

Dataset	Method	0	1	5	10	20
FriendsPersona	Psychology Knowledge	0.514	0.525	0.580	0.580	0.609
	Prior Verbalizer	0.532	0.571	0.669	0.651	0.674
	Vanilla Prompts	0.562	0.636	0.640	0.618	0.642
	DesPrompt	**0.576**	**0.656**	**0.680**	**0.665**	**0.678**
Eassy	Psychology Knowledge	0.505	0.544	0.611	0.638	0.638
	Prior Verbalizer	0.565	0.579	0.647	0.655	0.675
	Vanilla Prompts	0.494	0.535	0.648	0.644	0.669
	DesPrompt	**0.566**	**0.632**	**0.668**	**0.658**	**0.678**
myPersonality	Psychology Knowledge	0.508	0.505	0.587	0.609	0.616
	Prior Verbalizer	0.508	0.525	0.637	0.645	0.645
	Vanilla Prompts	0.539	0.537	0.618	0.635	0.645
	DesPrompt	**0.625**	**0.628**	**0.665**	**0.670**	**0.659**
PAN-AP-2015	Psychology Knowledge	0.535	0.584	0.665	0.665	0.680
	Prior Verbalizer	0.549	0.601	0.665	0.694	0.699
	Vanilla Prompts	0.630	0.648	0.643	0.704	0.694
	DesPrompt	**0.672**	**0.672**	**0.682**	**0.706**	**0.711**

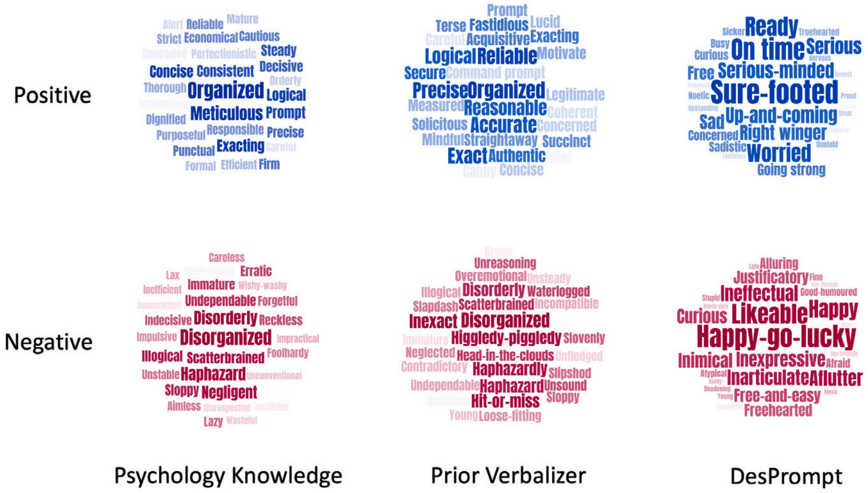
Positive
Negative
Psychology Knowledge
Prior Verbalizer
DesPrompt
Alert Reliable Mature
Strict Economical Cautious
Controlled Perfectionistic Steady
Concise Consistent Decisive
Orderly
Thorough Organized Logical
Industrious
Dignified Meticulous Prompt
Purposeful Responsible Precise
Punctual Exacting Careful
Formal Efficient Firm
Prompt
Terse Fastidious Lucid
Careful Acquisitive Exacting
Logical Reliable Motivate
Secure Command prompt
Precise Organized Legitimate
Measured Reasonable Coherent
Solicitous Accurate Concerned
Mindful Straightaway Succinct
Exact Authentic Brief
Canny Concise
Sicker Ready Truehearted
Busy On time Serious
Curious Nervous
Free Serious-minded Honest
Organized
Noetic Sure-footed Proud
Upstanding
Sad Up-and-coming Great
Concerned Right winger Skeptical
Sadistic Worried Onefold
Fastidious
Going strong
Careless
Lax Absent-minded Erratic
Inefficient Immature Wishy-washy
Inconsistent Undependable Forgetful
Indecisive Disorderly Reckless
Impulsive Disorganized Impractical
Illogical Scatterbrained Foolhardy
Unstable Haphazard Unconventional
Sloppy Negligent
Aimless Disrespectful Unreliable
Lazy Wasteful
Green
Unreasoning
Overemotional Unsteady
Illogical Disorderly Waterlogged
Slapdash Scatterbrained Incompatible
Inexact Disorganized
Immature Higgledy-piggledy Slovenly
Neglected Head-in-the-clouds Unfledged
Contradictory Haphazardly Slipshod
Undependable Haphazard Unsound
Reckless Hit-or-miss Sloppy
Young Loose-fitting
Late Alluring
Justificatory Fine
See through
Stupid Ineffectual Good-humoured
Really dark
Curious Likeable Happy
About
Happy-go-lucky
Inimical Inexpressive Afraid
See through
Atypical Inarticulate Aflutter
Really
Deadening
Young Free-and-easy Meek
Disorganized
Freehearted

Fig. 8. The label words describing Conscientiousness in different verbalizers.

performance, and DesPrompt performs even slightly better than Prior Verbalizer. To investigate the possible reason for verification, we take an example for illustration. We show the word clouds of label words (for both positive and negative labels) to recognize Conscientiousness in Fig. 8. Each word cloud contains the top 30 label words of the corresponding class. The words with higher weights have larger sizes and deeper colors, the vice versa. We can see that the label words with higher weights in Psychology Knowledge are *Organized, Meticulous, Concise, …* and *Disorganized, Negligent, Haphazard, …*. Although they may precisely describe the Conscientiousness trait, some of them are not used daily. As for the word clouds of the Prior Verbalizer including synonyms and antonyms, new words like *Accurate, Higgledy-piggledy, Head-in-the-clouds* occur. Some of them are colloquial compared with words in Psychology Knowledge. After the refinement with the PLM, label words with higher weights in the Posterior verbalizer become *On time, Sure-footed, Happy-go-lucky, Likeable, …*. They describe specific attributes of Conscientiousness, not precisely like the label words above, but more distinguishable and easy to use. It increases the probability of PLM filling these words to the prompts to achieve better performance.

Then, we focus on analyzing the prompt content. Quantitively, we compare the results of Vanilla Prompts and DesPrompt in Table 7. Among all four datasets, DesPrompt outperforms Vanilla Prompts by a large margin, especially when there are fewer data (0 shot and 1 shot). We then qualitatively analyze the reason.

We show the Top 10 prompt content generated for the FriendsPersona dataset as examples in Table 8. These prompts are generated by the Vanilla T5 model and the pre-finetuned T5 in DesPrompt with the same label words. It is worth noting that referring to the design of DesPrompt, the prompt contents in both columns are generated using the label words from the prior verbalizer, rather than the refined posterior label words. We tried to generate the prompt content after we obtained the posterior verbalizer, but the quality of the prompt content did not change much. So, we keep the current setting.

We can see that prompts generated by Vanilla T5 are shorter than the pre-finetuned T5. Besides, their patterns are similar: most of them use the first-person voice and put the [MASK] to the end of the prompt. While the prompts in the second column share

Table 8
Top 10 prompt content generated by T5 before and after the pre-finetuning.

Vanilla T5	Pre-finetuned T5 (DesPrompt)
x_{in} It's [MASK] .	x_{in} and it's always [MASK] .
x_{in} I [MASK] .	x_{in} it's [MASK] .
x_{in} This is [MASK] .	x_{in} i'm [MASK] .
x_{in} I am [MASK] .	x_{in} you're [MASK] .
x_{in} I was [MASK] .	x_{in} she is [MASK] .
x_{in} I'm [MASK] !	x_{in} it is so [MASK] .
x_{in} That's [MASK] .	x_{in} is it [MASK] ?
x_{in} I'm not [MASK] .	x_{in} it's kind of [MASK] .
x_{in} I'm so [MASK] .	x_{in} i'm [MASK] with this.
x_{in} I'm just [MASK] .	x_{in} it's [MASK] and it's not that bad.

relatively diverse voices (*e.g., it's, I'm, you're, she is...*) as well as different collocation words like *always, kind of, with it,* for some specific adjectives. These characteristics link x_{in} and the label words in different scenarios, which enables us to ensemble the results from diverse prompts for better personality recognition performance.

It is worth mention that the prompt contents presented in Table 8 are general prompt content candidates that are suitable for all the samples within the FriendsPersona dataset (one of the four datasets), rather than being specific to a particular x_{in}. In other words, for each specific x_{in} in FriendsPersona, the candidate prompt content would be the same as shown in Table 8.

There are two reasons why we adopt this approach instead of generating prompt content specifically for each input x_{in}:

- As mentioned before, finding the optimal prompt for a specific x_{in} can be challenging, and different prompts can yield significantly different results. Therefore, ensembling multiple prompt contents produce more robust results compared to relying on a single prompt.
- The process of generating prompt content requires the pre-finetuned T5 model to fill in numerous instances with possible label words for each x_{in}. This can be time-consuming in real-world usage scenarios. Thus, we pre-generate all the prompt content candidates using samples from the training set.

We use an instantiated x_{in} in the test set for illustration. If we are recognizing whether x_{in}: *Whoa, hey! What are you doing? Trying to get me drunk?*, is NEU or not, and the label word **counteractive** is highly related to the label. For the prompt content generated by Vanilla T5, the probability of filling in **counteractive** is quite low. However, for the prompt content generated by DesPrompt: x_{in} *i'm [MASK] with this.* is more suitable for filling in **counteractive**. So, in this case, the more diverse prompt content in DesPrompt increases the probability of filling in a highly related label word, thereby improving the performance of personality recognition.

6. Conclusion and future work

In this study, we introduced DesPrompt, a method for generating personality-descriptive prompts to fine-tune language models with limited annotated data. Our method overcomes two major challenges in personality recognition by (1) finding precise and commonly used label words and (2) generating informative and general prompt content. Our experiments on four datasets showed that DesPrompt outperforms existing fine-tuning and prompt-based methods, especially in zero-shot and few-shot scenarios.

DesPrompt can efficiently use descriptive adjectives for personality recognition in text. So, if descriptive adjectives are given, DesPrompt can be easily generalized to other popular personality models *e.g.,* MBTI model in trait theory. In future work, we are interested in finding the common and different adjectives in describing these personality models, to generalize DesPrompt and other personality recognition methods based on the lexical hypothesis.

Besides, with the ability to efficiently recognize personalities, DesPrompt opens the door for integrating personality recognition into conversational agents. We also aim to integrate the DesPrompt model into the open-domain dialog system and design conversational agents that recognize the personalities of users and respond to them appropriately in our future study.

CRediT authorship contribution statement

Zhiyuan Wen: Conceptualization, Methodology, Experiments, Validation, Formal analysis, Investigation, Manuscript writing. **Jiannong Cao:** Conceptualization, Formal analysis, Review, Editing, Supervision. **Yu Yang:** Conceptualization, Methodology, Validation, Manuscript writing, Review, Editing, Investigation. **Haoli Wang:** Conceptualization, Methodology, Experiments, Validation. **Ruosong Yang:** Conceptualization, Validation, Review, Editing, Investigation. **Shuaiqi Liu:** Validation, Review, Editing, Investigation.

Data availability

Data will be made available on request.

Acknowledgments

This work was conducted at the Research Institute for Artificial Intelligence of Things (RIAIoT) at PolyU and supported by the Hong Kong Jockey Club Charities Trust fund (No. 2021-0369), and PolyU Research and Innovation Office (No. BD4A and CD5H).

References

Aghajanyan, A., Gupta, A., Shrivastava, A., Chen, X., Zettlemoyer, L., & Gupta, S. (2021). Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 5799–5811). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.emnlp-main.468, URL https://aclanthology.org/2021.emnlp-main.468.

Allport, G. W., & Odbert, H. S. (1936). Trait-names: A psycho-lexical study. *Psychological Monographs*, *47*(1), i.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Cattell, R. B. (1943). The description of personality. I. Foundations of trait measurement. *Psychological Review*, *50*(6), 559.

Chen, Y., Liu, Y., Dong, L., Wang, S., Zhu, C., Zeng, M., et al. (2022). AdaPrompt: Adaptive model training for prompt-based NLP. In *Findings of the association for computational linguistics: EMNLP 2022* (pp. 6057–6068). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, URL https://aclanthology.org/2022.findings-emnlp.448.

Chen, H., Yin, H., Li, X., Wang, M., Chen, W., & Chen, T. (2017). People opinion topic model: opinion based user clustering in social networks. In *Proceedings of the 26th international conference on world wide web companion* (pp. 1353–1359).

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1179, URL https://aclanthology.org/D14-1179.

Costa, P. T., & McCrae, R. R. (1992). Normal personality assessment in clinical practice: The NEO personality inventory. *Psychological Assessment*, *4*(1), 5.

Donnellan, M. B., Conger, R. D., & Bryant, C. M. (2004). The big five and enduring marriages. *Journal of Research in Personality*, *38*(5), 481–504.

Fang, H., Chen, C., Long, Y., Xu, G., & Xiao, Y. (2022). DTCRSKG: A deep travel conversational recommender system incorporating knowledge graph. *Mathematics*, *10*(9), 1402.

Galton, F. (1884). Measurement of character. *Fortnightly*, *36*(212), 179–185.

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 3816–3830). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.295, URL https://aclanthology.org/2021.acl-long.295.

Golbeck, J., Robles, C., & Turner, K. (2011). Predicting personality with social media. In *CHI'11 extended abstracts on human factors in computing systems* (pp. 253–262).

Goldberg, L. R. (1995). So what do you propose we use instead? A reply to block. *Psychological Bulletin*, *117*(2,221-225).

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., et al. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8342–8360). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.740, URL https://aclanthology.org/2020.acl-main.740.

Hu, S., Ding, N., Wang, H., Liu, Z., Wang, J., Li, J., et al. (2022). Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 2225–2240). Dublin, Ireland: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.acl-long.158, URL https://aclanthology.org/2022.acl-long.158.

Iacobelli, F., Gill, A. J., Nowson, S., & Oberlander, J. (2011). Large scale personality classification of bloggers. In *International conference on affective computing and intelligent interaction* (pp. 568–577). Springer.

Jain, D., Kumar, A., & Beniwal, R. (2022). Personality BERT: A transformer-based model for personality detection from textual data. In *Proceedings of international conference on computing and communication networks* (pp. 515–522). Springer.

Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, *8*, 423–438.

Jiang, H., Zhang, X., & Choi, J. D. (2020). Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (student abstract). *Vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (10), (pp. 13821–13822).

Jun, H., Peng, L., Changhui, J., Pengzheng, L., Shenke, W., & Kejia, Z. (2021). Personality classification based on bert model. In *2021 IEEE international conference on emergency science and information technology* (pp. 150–152). IEEE.

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.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*(9), 1–35.

Lynn, V., Balasubramanian, N., & 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* (pp. 5306–5316).

Mairesse, F., & Walker, M. (2006). Automatic recognition of personality in conversation. In *Proceedings of the human language technology conference of the NAACL, companion volume: short papers* (pp. 85–88).

Mairesse, F., Walker, M. A., Mehl, M. R., & Moore, R. K. (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, *30*, 457–500.

Mehl, M. R., Gosling, S. D., & Pennebaker, J. W. (2006). Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life. *Journal of Personality and Social Psychology*, *90*(5), 862.

Mischel, W., Shoda, Y., & Ayduk, O. (2007). *Introduction to personality: Toward an integrative science of the person*. John Wiley & Sons.

Moreno, D. R. J., Gomez, J. C., Almanza-Ojeda, D.-L., & Ibarra-Manzano, M.-A. (2019). Prediction of personality traits in twitter users with latent features. In *2019 international conference on electronics, communications and computers* (pp. 176–181). IEEE.

Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, *71*(2001), 2001.

Pennebaker, J. W., & King, L. A. (1999). Linguistic styles: language use as an individual difference. *Journal of Personality and Social Psychology*, *77*(6), 1296.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.. *Journal of Machine Learning Research*, *21*(140), 1–67.

Rammstedt, B., & John, O. P. (2007). Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German. *Journal of Research in Personality*, *41*(1), 203–212.

Rangel, F., Celli, F., Rosso, P., Potthast, M., Stein, B., & Daelemans, W. (2015). Overview of the 3rd Author Profiling Task at PAN 2015. In L. Cappellato, N. Ferro, G. Jones, & E. San Juan (Eds.), *Lecture notes in computer science*: *Vol. 1391*, *Working notes papers of the CLEF 2015 evaluation labs*. [ISSN: 1613-0073] URL http://ceur-ws.org/Vol-1391/.

Ren, Z., Shen, Q., Diao, X., & Xu, H. (2021). A sentiment-aware deep learning approach for personality detection from text. *Information Processing & Management*, *58*(3), Article 102532.

Rissola, E. A., Bahrainian, S. A., & Crestani, F. (2019). Personality recognition in conversations using capsule neural networks. In *IEEE/WIC/ACM international conference on web intelligence* (pp. 180–187).

Roccas, S., Sagiv, L., Schwartz, S. H., & Knafo, A. (2002). The big five personality factors and personal values. *Personality and Social Psychology Bulletin*, *28*(6), 789–801. http://dx.doi.org/10.1177/0146167202289008.

Roshchina, A., Cardiff, J., & Rosso, P. (2011). A comparative evaluation of personality estimation algorithms for the twin recommender system. In *Proceedings of the 3rd international workshop on search and mining user-generated contents* (pp. 11–18).

Saucier, G., & Goldberg, L. R. (1996). Evidence for the big five in analyses of familiar english personality adjectives. *European Journal of Personality*, *10*(1), 61–77.

Schick, T., Schmid, H., & Schütze, H. (2020). Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th international conference on computational linguistics* (pp. 5569–5578). Barcelona, Spain (Online): International Committee on Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.coling-main.488, URL https://aclanthology.org/2020.coling-main.488.

Schick, T., & Schütze, H. (2021a). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume* (pp. 255–269). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.eacl-main.20, URL https://aclanthology.org/2021.eacl-main.20.

Schick, T., & Schütze, H. (2021b). It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 2339–2352). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.naacl-main.185, URL https://aclanthology.org/2021.naacl-main.185.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., et al. (2013). Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLoS One*, *8*(9), Article e73791.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 4222–4235). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-main.346, URL https://aclanthology.org/2020.emnlp-main.346.

Singh, P., et al. (2002). The public acquisition of commonsense knowledge. In *Proceedings of AAAI spring symposium: acquiring (and using) linguistic (and world) knowledge for information access*.

Souri, A., Hosseinpour, S., & Rahmani, A. M. (2018). Personality classification based on profiles of social networks' users and the five-factor model of personality. *Human-Centric Computing and Information Sciences*, *8*(1), 1–15.

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54.

Tighe, E. P., Ureta, J. C., Pollo, B. A. L., Cheng, C. K., & de Dios Bulos, R. (2016). Personality trait classification of essays with the application of feature reduction. In *SAAIP@ IJCAI* (pp. 22–28).

Tkalcic, M., & Chen, L. (2015). Personality and recommender systems. In *Recommender systems handbook* (pp. 715–739). Springer.

Wang, Y., Xia, C., Wang, G., & Philip, S. Y. (2022). Continuous prompt tuning based textual entailment model for E-commerce entity typing. In *2022 IEEE international conference on big data (big data)* (pp. 1383–1388). IEEE.

Welch, B. L. (1947). The generalization of student's problem when several different population varlances are involved. *Biometrika*, *34*(1–2), 28–35.

Wen, Z., Cao, J., Yang, R., Liu, S., & Shen, J. (2021). Automatically select emotion for response via personality-affected emotion transition. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 5010–5020). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.findings-acl.444, https://aclanthology.org/2021.findings-acl.444.

Yin, C., Zhang, X., & Liu, L. (2020). Reposting negative information on microblogs: Do personality traits matter? *Information Processing & Management*, *57*(1), Article 102106.