Multi-Document Transformer for Personality Detection

Feifan Yang, Xiaojun Quan*, Yunyi Yang, Jianxing Yu
School of Data and Computer Science, Sun Yat-sen University, China
{yangff6, yangyy37}@mail2.sysu.edu.cn, {quanxj3, yujx26}@mail.sysu.edu.cn

Abstract

Personality detection aims to identify the personality traits implied in social media posts. The core of this task is to put together information in multiple scattered posts to depict an overall personality profile for each user. Existing approaches either encode each post individually or assemble posts arbitrarily into a new document that can be encoded sequentially or hierarchically. While the first approach ignores the connection between posts, the second tends to introduce unnecessary post-order bias into posts. In this paper, we propose a multi-document Transformer, namely Transformer-MD, to tackle the above issues. When encoding each post, Transformer-MD allows access to information in the other posts of the user through Transformer-XL's memory tokens which share the same position embedding. Besides, personality is usually defined along different traits and each trait may need to attend to different post information, which has rarely been touched by existing research. To address this concern, we propose a dimension attention mechanism on top of Transformer-MD to obtain trait-specific representations for multi-trait personality detection. We evaluate the proposed model on the Kaggle and Pandora MBTI datasets and the experimental results show that it compares favorably with baseline methods.

Introduction

Personality detection is a fundamental task in psychology with wide applications in areas such as public health (Friedman and Kern 2014), personalized medicine (Matz et al. 2017), and mental health (Bagby et al. 1995). Besides, it can provide insightful information for many natural language processing (NLP) tasks. For example, Fung et al. (2016) developed a virtual interactive dialogue system by incorporating the user's emotion, sentiment, and personality. Lynn et al. (2017) proposed a general NLP task by adapting user factors such as age, gender, and personality traits.

Traditional questionnaire-based approaches to personality detection are time-consuming and laborious. Whereas in the era of social media, users generate a tremendous number of posts containing their behavior trace every day, providing new possibilities for personality detection research (Xue et al. 2018; Keh, Cheng et al. 2019; Jiang, Zhang, and Choi 2020). The typical setting of this task is: for each user, multiple scattered posts are provided, and the objective is to piece together information in the posts into a comprehensive user personality profile.

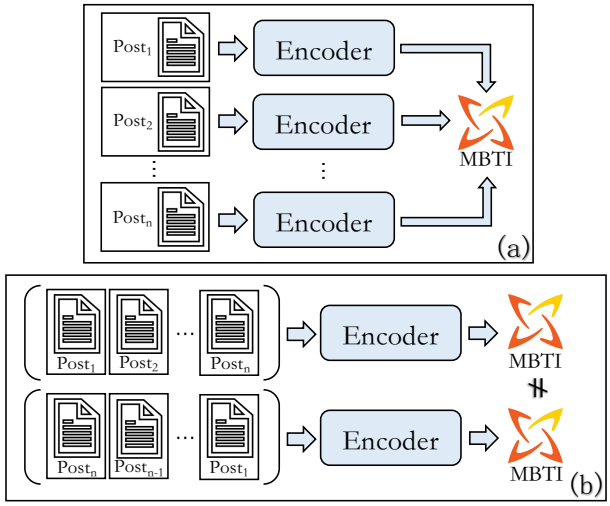

Figure 1: Existing approaches to combine multiple posts for personality detection. (a) encodes each post individually. (b) encodes posts by integrating them into a new document with arbitrary order, where "[ ]" means the concatenation of posts and "≠" means an inconsistent result for the posts containing the same personality information.

Existing approaches to automatically combining multiple posts for personality detection can be broadly divided into two categories. Firstly, as shown in Figure 1(a), each post is first encoded independently and then averaged into the user representation (Hernandez and Knight 2017; Keh, Cheng et al. 2019). This approach simply ignores the fact that the posts of a user may not be of equal importance and they may also need to interact to complement the personality profile. Secondly, as shown in Figure 1(b), the posts are integrated into a long flat sequence with an arbitrary post order and then encoded hierarchically (Lynn, Balasubramanian, and Schwartz 2020; Xue et al. 2018) or sequentially (Zhou et al. 2019; Jiang, Zhang, and Choi 2020; Yang et al. 2019b). However, models trained in this way may learn the extra post-order bias, resulting in inconsistent results of each

* Xiaojun Quan is the corresponding author.

user[1] and affecting the generalization ability of the models. Besides, personality is defined in terms of different dimensions (traits) and different post information is likely to contribute to different dimensions, which has rarely been mentioned by existing research.

In this paper, we propose a **M**ulti-**D**ocument **Transformer** model, namely Transformer-MD, to overcome the above limitations. Transformer-MD is a post-order-agnostic encoder based on Transformer-XL (Dai et al. 2019) and initialized by the pre-trained XLNet (Yang et al. 2019b). When encoding each post, it allows access to other post information of the same user through memory tokens that share the same position embedding. Moreover, we define a personality trait-specific attention, dimension attention (DA), on top of Transformer-MD, which allows each personality dimension to focus on specific post information they care most about.

The contributions of this work can be summarized as:

- We propose a novel post-order-agnostic encoder to put together the posts of a user to depict an overall personality profile without introducing the post-order bias.
- We propose a dimension attention mechanism to focus each personality dimension on relevant post information.
- We provide thorough analyses and discussions to demonstrate the effectiveness of the proposed model.

Related Work

Most previous works on personality detection rely on handcrafted feature engineering (Yarkoni 2010; Schwartz et al. 2013; Amirhosseini and Kazemian 2020). They may use various psycholinguistic attributes extracted by Linguistic Inquiry and Word Count (Pennebaker, Francis, and Booth 2001) or statistical features extracted by bag-of-words models. Obviously, these methods are limited by their capability in extracting many useful features (Lynn, Balasubramanian, and Schwartz 2020). Recently, deep learning methods have been widely applied and become mainstream for personality detection. Essentially, personality detection can be regarded as a multi-document multi-label classification task, which is related to the following domains.

Multi-Document Encoding

Multi-document encoding has been extensively studied, yet we will only review those related to personality detection from four perspectives: single encoder, graph encoder, hierarchical network, and concatenation encoder.

The single-encoder approach is predominately adopted in personality detection, which encodes each post independently with a single model. Hernandez and Knight (2017) and Tandera et al. (2017) used LSTM to encode each post with the GloVe embeddings. Keh, Cheng et al. (2019) used BERT to encode each post. However, these methods generally ignore the dependency between posts, which is detrimental to integrate the scattered personality information across different posts. An alternative approach is to use a graph encoder, modeling the interaction between posts by graph neural networks. This approach has been widely investigated in multi-document evidence reasoning tasks (Zhao et al. 2019; Zhou et al. 2019; Ding et al. 2019; Liu et al. 2020). Unlike Wikipedia evidence documents, there are no hyperlinks or priori personality-links between user posts, making it non-trivial to directly apply such models for personality detection.

The hierarchical-network approach tries to encode the posts hierarchically. Amirhosseini and Kazemian (2020) proposed to encode each post first via a gated recurrent unit (GRU), and then passed the encodings to a second GRU to realize the interaction between posts. Xue et al. (2018) first designed an AttRCNN module to encode each post and then used a convolutional neural network for the interaction between posts. The final strategy is the concatenation encoder which concatenates the user's posts into a new document in a certain order and then encodes it with a sequence model such as BERT (Zhou et al. 2019; Jiang, Zhang, and Choi 2020) and XLNet (Yang et al. 2019b). Intuitively, both of the hierarchical-network and concatenation-encoder approaches introduce extra post-order bias into the posts, impairing the generalization ability of models.

Multi-Dimension Classification

One of the challenges for multi-dimension classification is to allow different dimensions to focus on different information. Lynn, Balasubramanian, and Schwartz (2020) attempted to address this by training an appropriative model for each dimension, which ignores the interaction between personality dimensions. In other multi-label classification tasks, Yang et al. (2018, 2019a) adopted an attention mechanism to automatically extract informative features for each category. Vu, Nguyen, and Nguyen (2020) also proposed an attention-based method to obtain label-specific vectors that represent useful clinical text fragments relating to certain international classification of disease diagnosis codes. In this paper, we explored a dimension attention for personality detection.

Approach

In this section, we first formulate the problem of personality detection and then provide the details of our Transformer-MD model and the dimension attention module.

Problem Definition

Given a set $X = \{x_1, x_2, \cdots, x_n\}$ of n posts from a user, where $x_i = [t_{i1}, t_{i2}, \cdots, t_{ik}]$ is i-th post with k tokens, personality detection can be formulated as a user-level classification problem. The model takes X as input and produces a user representation $U \in \mathbb{R}^d$, or $U \in \mathbb{R}^{t \times d}$ as in our model, where d is the hidden size and t is the number of personality traits. Based on U, t personality traits $Y = \{y_1, y_2, \cdots, y_t\}$ are predicted by t classifiers individually, where y_i is chosen from a trait-specific label set $\{y_{i1}, y_{i2}, \cdots, y_{im}\}$, e.g., $t = 4$ and $m = 2$ in the MBTI taxonomy. We also refer to the traits technically as personality dimensions in this paper.

Multi-Document Transformer

The proposed Transformer-MD is inspired by Transformer-XL, attempting to encode and integrate multiple posts of

[1]Given the same content of posts, the personality type they represent should be consistent whatever order they are integrated in.

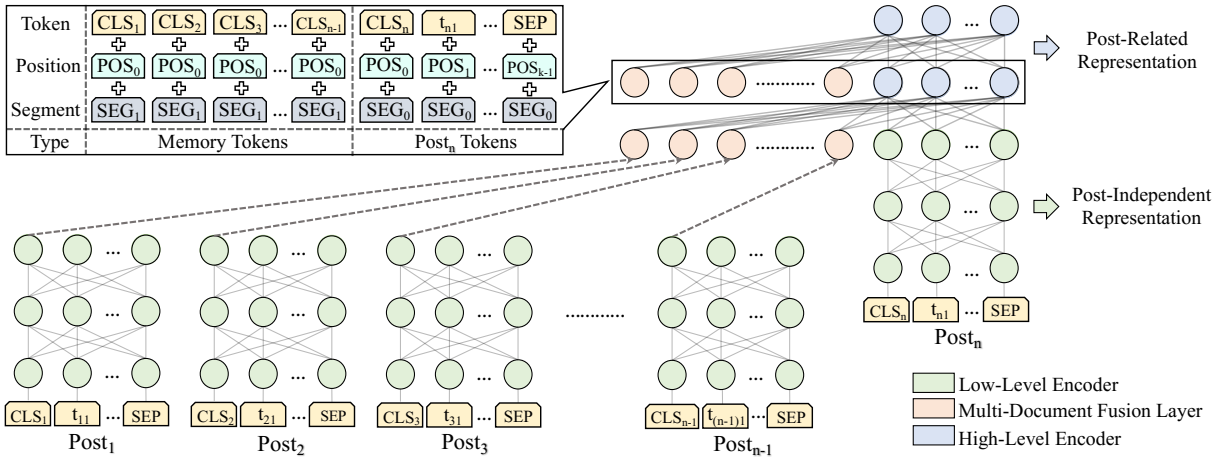

Figure 2: Overview of our Transformer-MD, which consists of a low-level encoder, a multi-document fusion layer and a high-level encoder, distinguished by different colors. All the posts of a user are encoded in parallel but we only show the whole process of Post$_n$. Three types of embedding used in the model are listed in the upper left corner.

a user in a post-order-agnostic manner. The purpose of Transformer-XL is to relieve the limit of Transformer on input size to enable learning of long-distance dependency beyond a fixed length without disrupting temporal coherence (Dai et al. 2019). Specifically, Transformer-XL divides a long document into multiple fixed-length segments and encodes them with a segment-level recurrence mechanism. After encoding a segment, Transformer-XL stores its hidden states in a memory bank and reuses them for future segments. Inspired by this mechanism, we treat each post as a segment. While processing each post, we store the rest posts from the same user in the memory, so that dependencies between posts can be modeled as in Transformer-XL.

Nevertheless, a user may have dozens to hundreds of posts, and each post may contain dozens of tokens, so it is unrealistic to put all of them into memory. Inspired by the two-stage encoding scheme of Transformer-XH (Zhao et al. 2019), we pre-encode each post separately for several initial layers of Transformer-XL, in which post information will be aggregated into their respective CLS tokens. Then, we put only the CLS tokens instead of all the post tokens into memory and model the dependency between posts in the remaining layers of Transformer-XL. Besides, Transformer-XL encodes memory tokens sequentially for semantic coherence between segments, implemented by position embeddings. However, posts are scattered and submitted by users randomly, and there is no natural order between them for personality detection. Therefore, we make the memory tokens of all posts to share the same position embedding so that posts can interact without introducing post-order bias. As shown in Figure 2, Transformer-MD can be decomposed into a low-level encoder, a multi-document fusion layer, and a high-level encoder.

Low-level encoder is composed of several bottom layers (9 in this paper) of Transformer-XL, aiming to learn post-independent representation for each post by encoding them individually. In this way, the information of a post will be aggregated into its CLS token. This encoder is similar to Transformer-XL while encoding the first segment without memory. Formally, for the i-th post, this encoder updates the representations H_i layer by layer as follows. At layer l, it computes the representation of each token by gathering information from the other tokens in the post:

$$Q_i^l, K_i^l, V_i^l = H_i^{l-1}W_q^T, H_i^{l-1}W_k^T, H_i^{l-1}W_v^T \quad (1)$$

$$H_i^l = \mathrm{TransformerLayer}\left(Q_i^l, K_i^l, V_i^l\right) \quad (2)$$

In Equation (1) and (4), W_q^T, W_k^T and W_v^T represent learnable parameters for the query, key and value in Transformer (Vaswani et al. 2017), respectively. In Equation (2) and (5), $\mathrm{TransformerLayer}$ contains a multi-head self-attention and a feed-forward network (Vaswani et al. 2017). Particularly, H_i^0 in the first layer is set to the embeddings of x_i.

Multi-document fusion layer is used to construct memory tokens for the high-level encoder. Formally, for the i-th post to be encoded in layer l, we concatenate the CLS representations $M_i^{l-1} = \{H_{1,cls}^{l-1}, \cdots, H_{i-1,cls}^{l-1}, H_{i+1,cls}^{l-1}, \cdots, H_{n,cls}^{l-1}\} \in \mathbb{R}^{(n-1)\times d}$ of the other posts from the preceding layer to construct the memory. In doing this, the information of the other posts from the same user is stored in the memory. Besides, we allow the position embedding of each CLS to be shared in the memory so that the post order is ignored. We also add segment embeddings to distinguish the post from the other posts, as shown in Figure 2.

High-level encoder is composed of the last few layers (3 in this paper) of Transformer-XL. It aims to learn a post-related representation for each post by selectively collecting information from the other posts in the memory with the multi-head self-attention of Transformer. This process is similar to Transformer-XL while encoding the second segment with memory storing the first segment. Formally, for the i-th post, this encoder updates the representation H_i layer by layer as follows. At layer l, it computes the representation of each token by gathering information from the other tokens in the

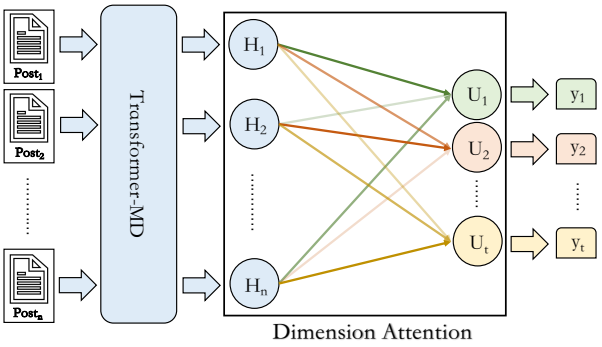

Figure 3: Overview of our dimension attention module.

post and the memory:

$$H_{i,mem}^{l-1} = [M_i^{l-1} \circ H_i^{l-1}] \tag{3}$$

$$Q_i^l, K_i^l, V_i^l = H_i^{l-1} W_q^T, H_{i,mem}^{l-1} W_k^T, H_{i,mem}^{l-1} W_v^T \tag{4}$$

$$H_i^l = \text{TransformerLayer}\left(Q_i^l, K_i^l, V_i^l\right) \tag{5}$$

In Equation (3), $\circ$ denotes the concatenation operation of two hidden states.

Dimension Attention

As described above, Transformer-MD put multiple posts together to obtain the post-related representations $H \in \mathbb{R}^{n \times d \times k}$ for a user. It is also mentioned that different parts of a post may contribute to different personality dimensions. Therefore, we develop a dimension attention module to further refine the representation H and obtain t dimension-specific vectors $U = [U_1, U_2, \cdots, U_t]$. The architecture of this module is shown in Figure 3. Following Lin et al. (2017) and Vu, Nguyen, and Nguyen (2020), we employ a multi-layer perceptron to compute the weights as:

$$H_r = \text{reshape}\left(H\right) \in \mathbb{R}^{d \times (n \times k)} \tag{6}$$

$$Z_r = \tanh\left(W_t H_r\right) \in \mathbb{R}^{d_t \times (n \times k)} \tag{7}$$

$$A_r = \text{softmax}\left(W_a Z_r\right) \in \mathbb{R}^{t \times (n \times k)} \tag{8}$$

Here, $reshape$ in Equation (6) is a matrix deformation function, $W_t \in \mathbb{R}^{d_t \times d}$ in Equation (7) and $W_a \in \mathbb{R}^{t \times d_t}$ in Equation (8) are learnable parameters, where d_t is a hyper-parameter to be tuned. The i-th row of the attention matrix A_r^i corresponds to the weights of the i-th dimension, which is then multiplied by the hidden states H_r to produce the dimension-specific representation U_i as:

$$[U_1, U_2, \cdots, U_t] = [A_r^1 H_r^T, A_r^2 H_r^T, \cdots, A_r^t H_r^T]$$

Objective Function

For each personality dimension i, we pass U_i to a single-layer feed-forward network followed by a softmax function to generate the probabilities $\bar{y}_i$ for this dimension. Then, the cross-entropy loss is used to measure the loss and the t dimensions are jointly optimized in a general approach:

$$Loss(X, Y, \theta) = \frac{1}{t} \sum_{i=1}^{t} -y_i \log \bar{y}_i$$

Dataset	Types	Train	Validation	Test
Kaggle	*I* vs. *E*	4011 vs. 1194	1326 vs. 409	1339 vs. 396
	S vs. *N*	727 vs. 4478	222 vs. 1513	248 vs. 1487
	T vs. *F*	2410 vs. 2795	791 vs. 944	780 vs. 955
	P vs. *J*	3096 vs. 2109	1063 vs. 672	1082 vs. 653
Pandora	*I* vs. *E*	4278 vs. 1162	1427 vs. 386	1437 vs. 377
	S vs. *N*	610 vs. 4830	208 vs. 1605	210 vs. 1604
	T vs. *F*	3549vs. 1891	1120 vs. 693	1182 vs. 632
	P vs. *J*	3211 vs. 2229	1043 vs. 770	1056 vs. 758

Table 1: Statistics of the Kaggle and Pandora datasets.

Experiments

Datasets

Following previous studies (Hernandez and Knight 2017; Keh, Cheng et al. 2019; Gjurković et al. 2020), we conduct experiments on the Kaggle[2] and Pandora[3] MBTI personality datasets. While the former has 40-50 posts for each user, the later has dozens to hundreds and is annotated based on self-diagnoses of user (Gjurković et al. 2020). MBTI personality type divides people's personality into four dimensions, each containing two aspects: **I**ntroversion vs. **E**xtroversion (I vs. E), **S**ensing vs. i**N**tuition (S vs. N), **T**hink vs. **F**eeling (T vs. F), and **P**erception vs. **J**udging (P vs. J). Statistics of the datasets are presented in Table 1. As Hernandez and Knight (2017), we remove the words (e.g., "INTP") that match personality labels from all posts to avoid information leaks. Then, we randomly split them into a 60-20-20 proportion for training, validation, and testing, respectively. Due to the imbalanced distribution of labels, we use the macro-F1 metric for evaluation.

Baselines

To evaluate the proposed model intensively, we employ the following mainstream models as baselines for comparison:

- **SVM** and **XGBoost** (Amirhosseini and Kazemian 2020): Posts are firstly concatenated into a new document, and then SVM or XGBoost is applied based on TF-IDF features extracted from the document.
- **GloVe-LSTM$_{\text{mean}}$** (Tandera et al. 2017; Hernandez and Knight 2017): LSTM is adopted to independently encode each post with GloVe embeddings and then the mean of the post embeddings is taken to represent the user.
- **GRU-MLA$_{\text{BERT}}$** (Lynn, Balasubramanian, and Schwartz 2020): Based on BERT embeddings, this model uses two-level GRUs to generate the post encodings and the user representation, respectively.
- **BERT$_{\text{concat}}$** (Zhou et al. 2019; Jiang, Zhang, and Choi 2020): This method simply concatenates the posts of a user into a long document and encodes it by BERT.

[2]https://www.kaggle.com/datasnaek/mbti-type
[3]https://psy.takelab.fer.hr/datasets/all/

Methods	Kaggle					Pandora				
	I* vs. *E	***S* vs. *N***	***T* vs. *F***	***P* vs. *J***	**Average**	***I* vs. *E***	***S* vs. *N***	***T* vs. *F***	***P* vs. *J***	**Average**
SVM	53.34	47.75	76.72	63.03	60.21	44.74	46.92	64.62	56.32	53.15
XGBoost	56.67	52.85	75.42	65.94	62.72	45.99	48.93	63.51	55.55	53.50
GloVe-LSTM$_{\text{mean}}$	57.82	57.87	69.97	57.01	60.67	48.01	52.01	63.48	56.21	54.93
GRU-MLA$_{\text{BERT}-4}$	64.75	60.24	75.17	62.89	65.76	54.60	49.19	61.82	53.64	54.81
BERT$_{\text{concat}}$	58.33	53.88	69.36	60.88	60.61	54.22	49.15	58.31	53.14	53.71
BERT$_{\text{concat}-5}$	61.72	58.74	71.72	59.83	63.00	53.32	49.94	60.46	55.31	54.76
XLNet$_{\text{concat}}$	60.65	54.50	71.98	56.00	60.78	50.49	49.59	58.10	54.09	53.07
BERT$_{\text{cls}-\text{mean}}$	63.50	55.34	78.55	66.06	65.86	53.35	50.56	64.06	56.83	56.20
XLNet$_{\text{cls}-\text{mean}}$	62.53	61.93	77.19	64.84	66.62	52.66	48.75	68.66	57.14	56.80
BERT$_{\text{DA}}$	65.67	61.28	**79.19**	66.52	68.17	54.88	55.49	66.71	59.01	59.02
XLNet$_{\text{DA}}$	65.99	63.80	78.53	66.44	68.69	**55.49**	57.50	65.04	60.80	59.71
Transformer-MD$_{\text{cls}-\text{mean}}$	**67.80**	63.67	78.83	64.62	68.73	54.78	55.51	67.28	59.89	59.37
Transformer-MD$_{\text{DA}}$	66.08	**69.10**	**79.19**	**67.50**	**70.47**	55.26	**58.83**	**69.03**	**60.57**	**60.92**

Table 2: Overall results of different models in macro-F1(%), where the best results are shown in bold.

- $\mathbf{BERT_{cls-mean}}$ (Keh, Cheng et al. 2019): BERT is used to encode each post individually, and the average of the CLS representations is used for the user representation.
- $\mathbf{XLNet_{concat}}$: This model is similar to $\mathbf{BERT_{concat}}$ but uses XLNet to encode the concatenated document.
- $\mathbf{XLNet_{cls-mean}}$: This model is similar to $\mathbf{BERT_{cls-mean}}$ but uses XLNet as the post encoder.

Implementation Details

We use Pytorch (Paszke et al. 2019) to implement all the deep learning models on four 2080Ti GPU cards. For training, we use the Adam (Kingma and Ba 2014) optimizer with an initial learning rate α = 2e-5 and a mini-batch size of 24. Following previous work, we set the max number of posts to 50 for each user and the max length to 70 for each post. For GloVe-LSTM, we use the 300-dimensional GloVe word embeddings (Pennington, Socher, and Manning 2014) and set the hidden size to 300. For BERT and XLNet, their parameters are initialized by the bert-base-cased (Devlin et al. 2018) and xlnet-base-cased (Yang et al. 2019b) models, respectively. The hidden size d_t of our dimension attention module is set to 768. After training a fixed number of epochs, we select the model with the best macro-F1 on the validation set and evaluate its performance on the test set.

Overall Performance

The overall results are shown in Table 2, in which the models are organized into two groups. The first group are the existing models (from SVM to BERT$_{\text{cls}-\text{mean}}$) and variants (XLNet$_{\text{cls}-\text{mean}}$ and XLNet$_{\text{concat}}$), while the second group are models based on our Transformer-MD or DA module or both. We can observe that our final model (Transformer-MD$_{\text{DA}}$) achieves the highest average macro-F1 score, outperforming the existing state-of-the-art (SOTA) baseline (BERT$_{\text{cls}-\text{mean}}$) by 4.61 and 4.72 on Kaggle and Pandora, respectively. What's more, in the same pre-trained setting, our model also outperforms the variant of the SOTA baseline (XLNet$_{\text{cls}-\text{mean}}$) by 3.85 and 4.12 on Kaggle and Pandora, respectively. The results verify the effectiveness of our model in personality detection. Besides, we also present a variant of our model by removing the DA module, and the results (Transformer-MD$_{\text{cls}-\text{mean}}$) show that this novel multi-document encoder alone still outperforms the SOTA baseline (BERT$_{\text{cls}-\text{mean}}$) by 2.87 on Kaggle and 3.17 on Pandora, and outperform the variant of SOTA baseline (XLNet$_{\text{cls}-\text{mean}}$) by 2.11 on Kaggle and 2.57 on Pandora, respectively, demonstrating that the post-related representations (encoded by Transformer-MD) are better than post-independent representations (encoded by BERT or XLNet) for personality detection. A notable phenomenon of Transformer-MD$_{\text{cls}-\text{mean}}$ is that the results for different personality dimensions vary more obviously than our final model. We speculate the reason is that the former represents the whole personality dimensions with a shared representation which contains more information for the simple dimensions ("I/E" and "T/F") than the difficult dimensions ("S/N" and "P/J"). By contrast, our DA module can generate a specific representation for each dimension to alleviate this problem. To further verify this, we implement the DA module with XLNet and BERT as the post encoder, respectively, and the results (BERT$_{\text{DA}}$ and XLNet$_{\text{DA}}$) in Table 2 confirm that the performances are also improved.

Effect of Post Order

From Table 2 we can note that the models without post order (subscripted by "cls-mean") are superior to those with post order (subscripted by "concat"). To further examine the negative effect of post order, we take XLNet and BERT as an example and train XLNet$_{\text{concat}}$ and BERT$_{\text{concat}}$ in the original post order until they converge. We then run them on the training set again with a new random post order 5 times and record the average scores. Ideally, given the same content of posts, the personality type they represent should be consistent no matter what order they are integrated in. However, the results shown in Figure 4 suggest that it is not the

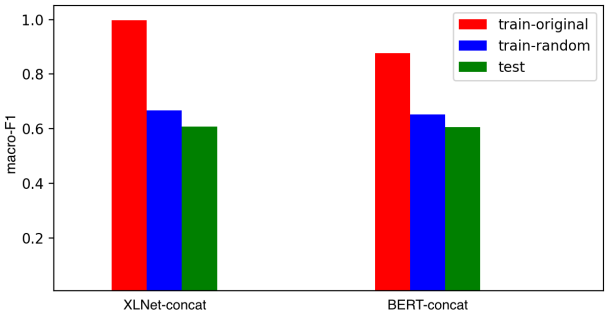

Figure 4: Results of study on post order on Kaggle. Red bars denote the performance of $\text{XLNet}_{\text{concat}}$/$\text{BERT}_{\text{concat}}$ trained until converge on the training set and green bars denote the test set. Blue bars are the average performance of five random post orders on the training set.

case, as the performance of $\text{XLNet}_{\text{concat}}$ and $\text{BERT}_{\text{concat}}$ on the training set are greatly reduced after altering the initial post order. This implies that the initial models tend to overfit the extra post-order bias for each user, affecting their generalization ability. A feasible remedy for this problem is to concatenate posts in different orders to generate different new documents and to reduce the post-order bias. As the results ($\text{BERT}_{\text{concat-5}}$) shown in Table 2, there are indeed certain amounts of improvement observed when each set of posts from the same users are organized into different documents in random orders. However, with each user containing dozens to hundreds of posts, the number of combinations is exponentially increasing. Thus, an order-agnostic approach is more desirable for personality detection.

Analysis of Transformer-MD

Ablation Study We conduct an ablation study to investigate the effects of the multi-document fusion layer (MFL), the shared position embedding, and the segment embedding in Transformer-MD. As shown in Table 3, the performance of the model drops visibly after removing any of them. Specifically, the performance drops by 1.78 when the MFL is removed, demonstrating that this module contributes considerably to personality detection. Moreover, when the shared position embedding or the segment embedding is removed, the performance drops by 0.99 and 1.00 respectively, which shows that it is crucial for the MFL module to ignore the post order bias and distinguish between different posts when putting them together for personality detection.

Effect of Post Connection One of the roles of our Transformer-MD is to model the connection between posts. To verify the effectiveness, we investigate the performance of our model while keeping only the top λ percent of connections between posts learned by our model (in terms of attention weights on memory tokens). As the results show in Figure 5, the improvements of Transformer-MD over the baselines mainly come from the 60% links, while the remaining 40% add little impact to the performance or even cause a slight degradation. This demonstrates that multi-head self-attention can effectively establish positive post connections and ignores useless or noisy ones by assigning different

Methods	macro-F1 (%)				
	I vs.E	*S vs.N*	*T vs.F*	*P vs.J*	Ave.
$\text{Transformer-MD}_{\text{DA}}$	66.08	69.10	79.19	67.50	**70.47**
–MFL	65.99	63.80	78.53	66.44	68.69
–share-position	66.13	66.70	79.03	66.05	69.48
–segment	65.12	66.99	79.70	66.07	69.47

Table 3: Results of ablation study of Transformer-MD on Kaggle, where "–" denotes the removal of a component.

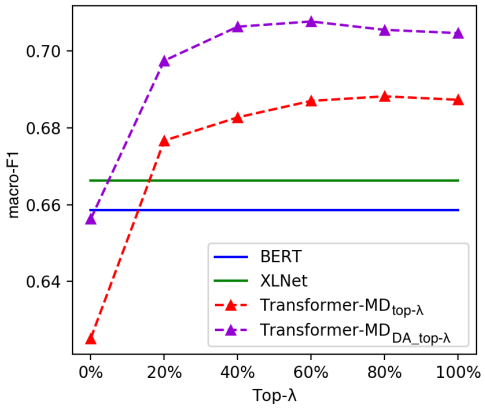

Figure 5: Performance of Transformer-MD on Kaggle when keeping only the top-λ percent links of posts.

weights. Another interesting observation is that the top 20% links can improve the model performance quickly, implying that high-quality connections are extremely predictive.

Case Study To further study what kind of connections are positive, we examine the memory attention weights generated by Equation (3)-(5) for a user with INFP personality types in the Kaggle dataset. We then pick out the posts with high-weight links and visualize them. As shown in Figure 6, positive connections are those having similar emotions, themes, or viewpoints which ensure the information in multiple posts can be put together to depict a profile.

Effect of High-Level Encoder Layers We also investigate how the number of layers for the low-level and high-level encoders affects the performance. To this end, we change the number of layers of the high-level encoder from 0 to 6 and plot the results in Figure 7. We can observe that $\text{Transformer-MD}_{\text{DA}}$ achieves the best performance when this number is set to 3. This implies that Transformer-MD needs about 9 layers to aggregate post information into CLS and another 3 layers to model the connection between posts.

Granularity of Dimension Attention

Our DA module gathers information at word-level rather than post-level, for word-level contains more information for multi-dimension personality detection. To verify this, we compare the DA module with a post-level attention method (CLS-Attn) that gathers information from only the CLS tokens of all posts. As shown in Table 4, our DA module outperforms the other approaches across all the post encoders.

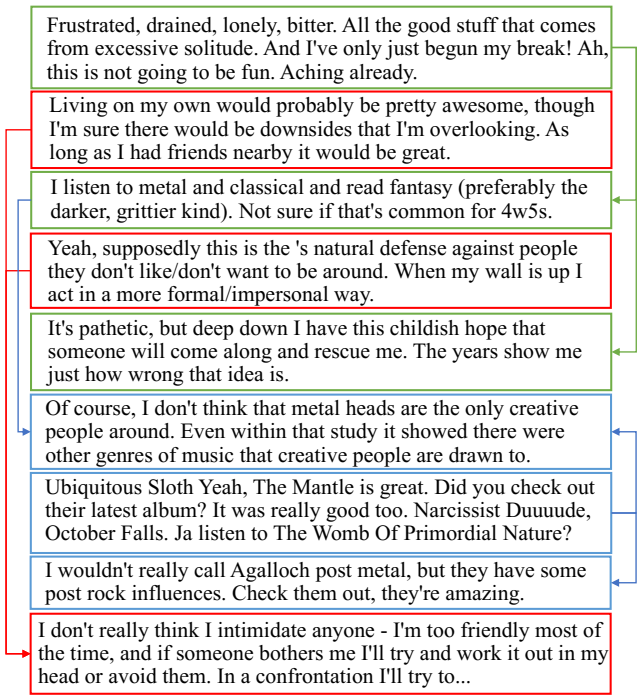

Figure 6: Links of posts established by Transformer-MD.

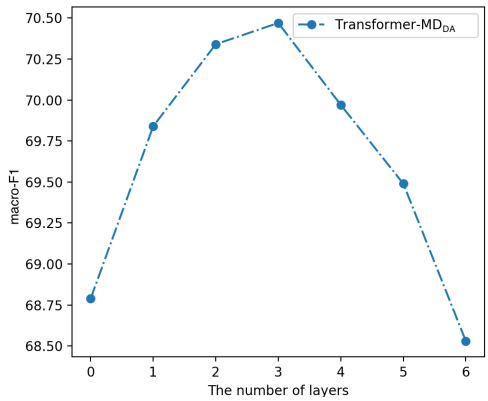

Figure 7: Performance of Transformer-MD$_{\mathrm{DA}}$ with different numbers of layers for the high-level encoder on Kaggle.

Error Analysis

The results in Table 2 show that our model achieves the best performance in all personality dimensions except "I/E". We finally conduct an error analysis using Transformer-MD$_{\mathrm{DA}}$ on the Kaggle dataset. Specifically, we record the macro-F1 scores of this model on the validation set during training. As shown in Figure 8, the model converges faster in the "I/E" and "T/F" personality dimensions than in "S/N" and "P/J". The best epoch range for "I/E" and "T/F" is from 4 to 7 but 7 to 10 for "S/N" and "P/J". Besides, the "T/F" dimension not only converges fast but also keeps good performance in the later epochs. The "S/N", "T/F" and "P/J" personality dimensions all achieve the best performance at the 9-th epoch, but "I/E" is slightly worse in this setting than its best epoch

Methods	Macro-F1 (%)		
	BERT	XLNet	Transformer-MD
CLS-Mean	65.86	66.62	68.73
CLS-Attn	66.37	67.19	69.56
DA	**68.17**	**68.69**	**70.47**

Table 4: Results of mean (CLS-Mean), word-level attention (DA) and post-level attention (CLS-Attn) on Kaggle.

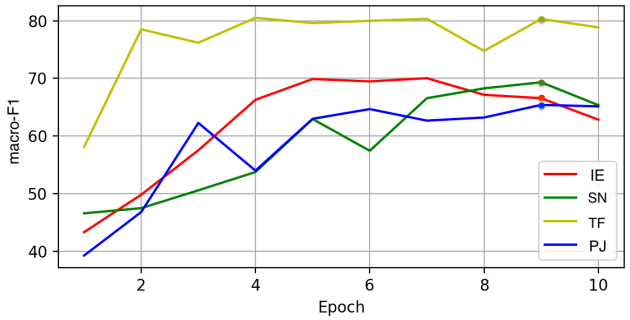

Figure 8: Performance of Transformer-MD$_{\mathrm{DA}}$ in different personality dimensions as the epoch increases.

(7-th). This phenomenon may indicate that personality dimensions have different training difficulties. The reason why Transformer-MD$_{\mathrm{DA}}$ performs less satisfied in the "I/E" dimension is probably that the model overfits this dimension quickly compared with the other dimensions.

Conclusion

In this paper, we proposed a multi-document Transformer, Transformer-MD, for personality detection. Transformer-MD intends to put together information in different posts to depict a personality profile for each user without introducing post orders. To this aim, it first encodes each post independently to obtain post-independent representations. Then, it generates post-related representations by disorderly fusing information from other posts. On top of Transformer-MD, we further proposed a dimension attention mechanism to generate a trait-specific representation for each personality dimension. Experimental results on two datasets show that combining Transformer-MD and dimension attention leads to a model that outperforms the baselines significantly.

Acknowledgments

We thank the anonymous reviewers for their constructive reviews. This work was supported by the Fundamental Research Funds for the Central Universities (No.19lgpy220), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355), and the National Natural Science Foundation of China (No. 61906217).

References

Amirhosseini, M. H.; and Kazemian, H. 2020. Machine Learning Approach to Personality Type Prediction Based on

the Myers–Briggs Type Indicator®. *Multimodal Technologies and Interaction* 4(1): 9.

Bagby, R. M.; Joffe, R. T.; Parker, J. D. A.; Kalemba, V.; and Harkness, K. L. 1995. Major Depression and the Five-Factor Model of Personality. *Journal of Affective Disorders* 38(2-3): 89–95.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1285.

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* .

Ding, M.; Zhou, C.; Chen, Q.; Yang, H.; and Tang, J. 2019. Cognitive graph for multi-hop reading comprehension at scale. *arXiv preprint arXiv:1905.05460* .

Friedman, H. S.; and Kern, M. L. 2014. Personality, Well-Being, and Health*. *Annual Review of Psychology* 65(1): 719.

Fung, P.; Dey, A.; Siddique, F. B.; Lin, R.; Yang, Y.; Bertero, D.; Wan, Y.; Chan, R. H. Y.; and Wu, C.-S. 2016. Zara: A virtual interactive dialogue system incorporating emotion, sentiment and personality recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, 278–281.

Gjurković, M.; Karan, M.; Vukojević, I.; Bošnjak, M.; and Šnajder, J. 2020. PANDORA Talks: Personality and Demographics on Reddit. *arXiv preprint arXiv:2004.04460* .

Hernandez, R.; and Knight, I. 2017. Predicting Myers-Bridge Type Indicator with text classification. In *Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA*, 4–9.

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* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* .

Liu, Z.; Xiong, C.; Sun, M.; and Liu, Z. 2020. Fine-grained Fact Verification with Kernel Graph Attention Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7342–7351. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.655.

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.

Lynn, V.; Son, Y.; Kulkarni, V.; Balasubramanian, N.; and Schwartz, H. A. 2017. Human Centered NLP with User-Factor Adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1146–1155. Copenhagen, Denmark: Association for Computational Linguistics. doi:10.18653/v1/D17-1119.

Matz, S. C.; Kosinski, M.; Nave, G.; and Stillwell, D. J. 2017. Psychological targeting as an effective approach to digital mass persuasion. *Proceedings of the National Academy of ences of the United States of America* 12714–12719.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* .

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

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Schwartz, H. A.; Eichstaedt, J. C.; Kern, M. L.; Dziurzynski, L.; Ramones, S. M.; Agrawal, M.; Shah, A.; Kosinski, M.; Stillwell, D.; Seligman, M. E.; et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one* 8(9): e73791.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems* 30: 5998–6008.

Vu, T.; Nguyen, D. Q.; and Nguyen, A. 2020. A Label Attention Model for ICD Coding from Clinical Text. *arXiv preprint arXiv:2007.06351* .

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, P.; Luo, F.; Ma, S.; Lin, J.; and Sun, X. 2019a. A Deep Reinforced Sequence-to-Set Model for Multi-Label Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5252–5258. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1518.

Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; and Wang, H. 2018. SGM: sequence generation model for multi-label classification. *arXiv preprint arXiv:1806.04822* .

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.

Yarkoni, T. 2010. Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality* 44(3): 363–373.

Zhao, C.; Xiong, C.; Rosset, C.; Song, X.; Bennett, P.; and Tiwary, S. 2019. Transformer-XH: Multi-Evidence Reasoning with eXtra Hop Attention. In *International Conference on Learning Representations*.

Zhou, J.; Han, X.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. *arXiv preprint arXiv:1908.01843* .