AWS-EP: A Multi-Task Prediction Approach for MBTI/Big5 Personality Tests

Anonymous ACL submission

Abstract

Personality and preferences are essential variables in computational sociology and social science. They describe differences between people at both individual and group levels. In recent years, automated approaches to detect personality traits have received much attention due to the massive availability of individuals' digital footprints. Furthermore, researchers have demonstrated a strong link between personality traits and various downstream tasks such as personalized filtering, profile categorization, and profile embedding. Therefore, the detection of individuals' personality traits has become a critical process for improving the performance of different tasks. In this paper, we build on the importance of the individual personality and propose a novel multitask modeling approach that understands and models the user personality based on its textual posts and comments within a multimedia framework. Experiments and results demonstrate that our model outperforms state-of-the-art performances across multiple famous personality datasets.

1 Introduction

Personality traits highlight the difference among the various individuals' characteristic patterns such as feeling, thinking, and behaving. Understanding people's core personality traits and knowing what people are good at can be very important in a wide variety of situations. It could ameliorate its social relationships, personal development, thinking patterns, and daily interaction capabilities. People are now very familiar with personality test systems such as the MBTI (Myers Briggs Type Indicator), 16 personalities, Big5 (Big five-factor model), and other tests. MBTI Isabel Briggs Myers and Hammer (1987) and Big5 Goldberg (1993) are the most well-known personality test systems. Both are used within a large scale of companies and therapy intuitions.

The MBTI system categorizes a person into 16 different categories using four main factors (Introverted, Intuitive, Thinking, and Perceiving). In this system, a user can only belong to one category. The Five-Factor (Big5) model measures five key dimensions of people's personalities. It measures its openness' OPN,' its Conscientiousness' CON,' its extraversion' EXT,' its agreeableness' AGR,' and its neuroticism' NEU.' In this personality system, a person belongs to all five categories to a certain degree, unlike the MBTI test, where a person can only be one of 16 categories. Recent research demonstrates that people prefer expressing their emotions, thoughts, and complaints on social media platforms such as Twitter, Instagram, Facebook, among other ones Yosephine Susanto and Cambria. (2020). Therefore, in modern times, there is a massive interest in designing automatic learning models that benefit from human digital footprints for different end-goals (example: online posts personality detection).

Recent works demonstrate that social media individual digital footprints are very effective for measuring personality traits Wu Youyou and Stillwell (2015). Despite the serious privacy concerns for individuals Sandra C Matz and Kosinski (2020), this challenging task has gained significant interest from psycholinguistics and natural language processing researchers due to its extensive downstream applications such as profile categorization and psychological treatment. Significant strides in machine learning and deep learning-based personality detection research have taken place in the past few years Yash Mehta and Eetemadi (2020), Wu Youyou and Stillwell (2015), Li et al. (2021), Tao Yang (2021). Moreover, other psychological research highlights the correlation and the dependency between pair personality test systems Furnham (1996). However, all existing automated approaches focus heavily on personality test systems independently, whether modeling the MBTI or the Big5 system.

Motivated by the above discussions, we propose the first automated multi-personality test systems modeling approach. We propose a novel multi-task personality prediction model named AWS-EP (All Weight Shared Electra for Personality prediction) 3.1. Our proposed model consists of an MLP (Multi-Layer Perceptron) architecture with two prediction heads (classification and regression), built on top of a fine-tuned Electra transformer model (see section A.1), to model both MBTI and Big5 personality test systems at the same time. We choose to use the Electra model because most recent published papers use Bert as their primary model to predict personality traits. No one has investigated the use of the Electra model to predict individuals' personality traits. Therefore, this paper aims to explore the benefits of using Electra instead of Bert for the personality trait prediction task by comparing its performance with existing state-of-the-art baselines on different datasets.

Moreover, we propose three other baselines, named **OC-EP** (Only Classification Electra for Personality prediction) 3.1, **OR-EP** (Only Regression Electra for Personality prediction)3.1, and **EWS-EP** (Electra Weights Shared for Personality prediction) 3.1, to locally evaluate the AWS-EP model and measure its performance compared to local baselines. Our proposed solution outperforms existing state-of-the-art models in different metrics. To the limit of our knowledge, this is the first automated personality detection approach that models individual personalities while considering more than one personality test system. Moreover, this is the first work that uses shared weights to predict both the categorical values for the MBTI system and the numerical values for the Big5 system at the same time. Also, it is the first work that tackles the Big5 personality trait prediction as a multi-label regression task. Experiments conducted on different benchmark datasets show that our AWS-EP model outperforms state-of-the-art models on different metrics.

It is important to highlight that our contribution in this work is not creating a novel model architecture for the NLP (Natural Language Processing) field in general. Our contribution is the implementation of different existing NLP mechanisms (pre-trained models, multi-task learning, and weight sharing) to create a novel architecture for the personality trait prediction problem. Moreover, we aim to explore the Electra model performance on the personality trait detection task compared to the existing state-of-the-art models.

2 Related work

Detecting personality traits can be based on various types of features, such as demographical data (gender, age, followers, etc.), text data (social media content, self-description, etc.) Different research studies have demonstrated that users' online behavior is significantly related to their personality Samuel D Gosling and Gaddis (2011), David John Hughes et al. (2012). Many have successfully applied different learning approaches for a social media-generated content personality trait detection Fabio Celli and Pianesi (2014). Wu Youyou and Stillwell (2015), demonstrate that the digital footprint-based analysis was better at measuring personality traits than close relatives or acquaintances (friends, family, colleagues, etc.) Mayuri Pundlik Kalghatgi and Sidnal (2015) detected the personality trait using an MLP network employing statistical and manual-crafted features. Despite the effectiveness of the manual-crafted features, these types of features are very time-consuming and computationally expensive. That is why researchers have been exploring new data types for personality trait detection.

Carducci et al. (2018) were the first to apply textual data for personality detection. They used an SVM (Support Vector Machine) model to do the personality detection on top of textual features instead of the statistical manual-crafted features. Following this work and with the advancement of deep learning approaches, Tommy Tandera (2017) applied personality detection over the text data using LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) approaches.

Gjurković et al. (2021)used BERT (Bidirectional Encoder Representations from Transformers) Devlin et al. (2019) to set a benchmark for their huge Pandora dataset Gjurković et al. (2021), which include three different personality tests', OCEAN which refers to the Big-Five model categories, MBTI, and Enneagram tests. The authors of this paper developed six regression models to predict age and Big5 traits and eight classification models (The four MBTI features, gender, region, and Enneagram features).

Experiments were done using traditional machine learning approaches such as linear/logistic regression and deep learning approaches such as MLP. In each model, the comments were encoded

using 1024-dimensional vectors derived using BERT, which produced a new benchmark for both regression and classification tasks for this dataset using macro F1- score and P-r-C (Pearson Correlation Coefficient) metrics.
Following this work Yang et al. (2021), used both textual and questionnaire answer information to enhance the contextual representation to benefit the personality prediction task.
Tao Yang (2021) combined graphical neural networks with a BERT transformer embedding model to detect personality traits. Their experiments show that their model outperforms the existing state-of-art model by 3.47 and 2.10 points on the average F1-score. To further enhance the effectiveness of personality traits, prediction models Yang Li et al. proposed a new 'Multitask Learning for Emotion and Personality Detection' Li et al. (2021) model. They combined the Bert transformer model and a 3 CNN layers model, allowing information sharing between the different layers to predict user personality and emotion using two different datasets. They also demonstrated that their work surpasses different state-of-the-art models on different metrics such as accuracy, macro-precision, macro-recall, and the macro-F1 metric. The contribution of their work consists of the use of a classification multitask neural network to classify two different tasks (personality and emotion).

Inspired by all the previous work and the significant performance improvements that the multi-task learning approach provides, we investigated the effect of using a multi-task MLP approach on top of a fine-tuned Electra transformer model Kevin Clark and Manning (2020), and compared its performance to the already exiting state-of-the-art baselines. We also investigate sharing weights between the MBTI and the Big5 personality tests. Furthermore, we looked at the similarity between these two personality tests.

3 Description of Models

Throughout this section, we define the different OC-EP (Only Classification Electra for Personality prediction), OR-EP (Only Regression Electra for Personality prediction), EWS-EP (Electra Weights Shared for Personality prediction), and AWS-EP (All Weights Shared Electra for Personality prediction) models architecture.

The four architectures are built on top of the Electra transformer model. Therefore to understand the proposed architectures, we need first to explain the working mechanism of this model (see Appendix section A.1). Using the pre-trained Electra masked language modeling head, we aim to produce a more contextual representation for each user textual sentence to achieve a better text classification performance.

3.1 Models architecture

We created four different baseline models to investigate the weight sharing performance for classification and regression personality prediction tasks: the OC-EP, OR-EP, EWS-EP, and AWS-EP models. We were curious if the independent prediction models would perform better than the weight-shared multi-task models. Figures 1, 2, 3, and 4 describe the main architecture for each baseline.

- **OC-EP:**

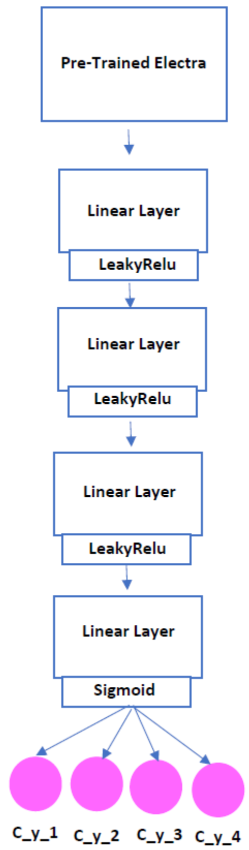

Figure 1: Only Classification Electra for personality prediction Architecture

This baseline is designed only for the classification task (predict the MBTI categories), and it is independent of the regression task.

The white boxes represent the different layers in the OC-EP architecture. The output of the sigmoid layer defines the probabilities of each category in the MBTI system, where C_{y1}, C_{y2}, C_{y3}, and C_{y4}, define the introverted, intuitive, thinking, and perceiving MBTI axis. The reason behind using the sigmoid function instead of the softmax function is that the softmax function is generally used when we have a multi-classification task (for example, from the five classes, we need to choose only 1 class). However, in our work, we have a multi-label task (from 5 classes, we can choose 1, 2,3, or even all five classes). This baseline is trained using the BCE (Binary Cross Entropy) loss function applied for each class (equation 1).

$$
\begin{aligned}
LOSS_{class} = -\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} Cy_{ij}\cdot\log(\hat{Cy_{ij}}) \\
+(1-Cy_{ij})\cdot(1-\log(\hat{Cy_{ij}}))
\end{aligned} \quad (1)
$$

where N $\{1..n\}$ defines the data size, M defines the different classes $\{1..4\}$, Cy_{ij} defines the i^{eth} row and j^{eth} class original value $\{0,1\}$, and $\hat{Cy_{ij}}$ defines the i^{eth} row and j^{eth} class predicted value $\{0,1\}$

This model is trained under the objective of minimizing the $LOSS_{class}$ (equation 2) where X defines the training data and θ_{class} defines the OC-EP model learning parameters.

$$
min_{\theta_{class}} \sum_{x\in X} LOSS_{class}(x,\theta_{class}) \quad (2)
$$

- **OR-EP:**

  This baseline is designed only for the regression task (predict the Big5 categories). It is independent of the classification task. The output of the last linear layer defines the numerical values (from 0 to 100) of each factor in the Big5 system, where R_{y1}, R_{y2}, R_{y3}, R_{y4}, and R_{y5} define the agreeableness, openness, conscientiousness, extraversion, neuroticism Big5 factors. This baseline is trained using the MSE (Mean Squared Error) loss function for each category (equation 3).

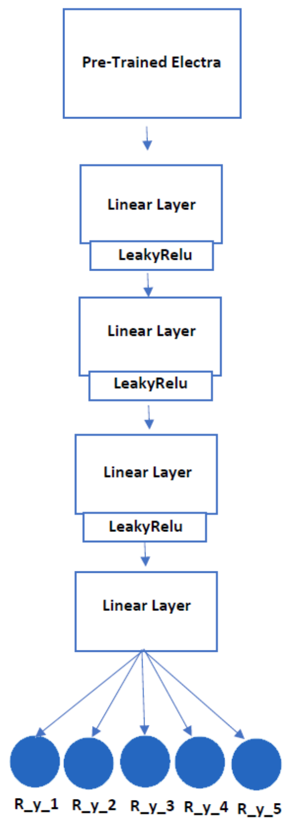

Figure 2: Only Regression Electra for Personality prediction Architecture

$$
LOSS_{reg} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(Ry_{ij}-\hat{Ry_{ij}})^2 \quad (3)
$$

where N $\{1..n\}$ defines the data size, M defines the different labels $\{1..5\}$, Ry_{ij} defines the i^{eth} row and j^{eth} label original value $\{0..100\}$, and $\hat{Ry_{ij}}$ defines the i^{eth} row and j^{eth} predicted value $\{0..100\}$
This model is trained under the objective of minimizing the $LOSS_{reg}$ (equation 4) where X defines the training data and θ_{reg} defines the OR-EP model learning parameters.

$$
min_{\theta_{reg}} \sum_{x\in X} LOSS_{reg}(x,\theta_{reg}) \quad (4)
$$

- **EWS-EP:**

  This model is designed to predict both classification (MBTI) and regression (Big5) tasks by sharing only the pre-trained Electra weights $h1$. Regression and classification heads are partially dependent as they share only the $h1$

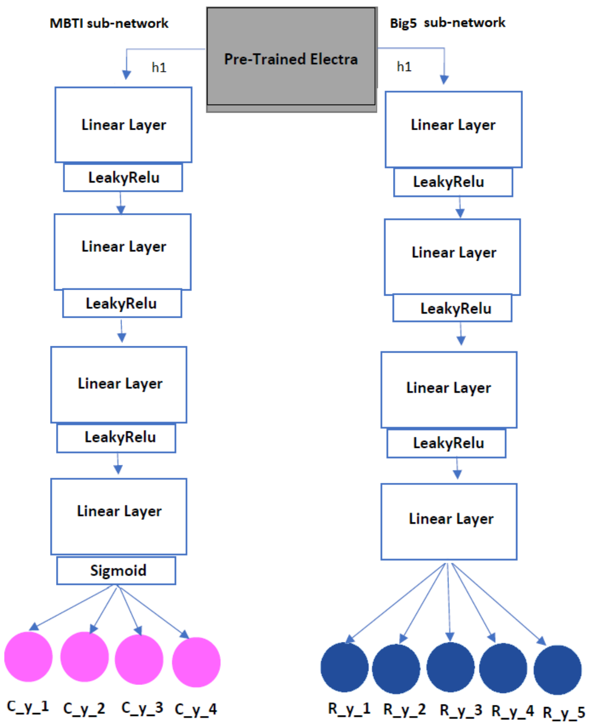

Figure 3: Electra Weights Shared for Personality prediction Architecture

Electra pre-trained model weights. The white boxes represent the independent layers for each sub-architecture. The gray box represents the shared layer 'weights' between the classification MBTI sub-network and the regression Big5 sub-network. The output of the last linear layer defines the numerical values (from 0 to 100) of each Big5 system personality factor (R_{y1}, R_{y2}, R_{y3}, R_{y4}). The output of the last sigmoid layer defines the probabilities of each category in the MBTI system, where C_{y1}, C_{y2}, C_{y3}, and C_{y4}, define the four MBTI personality factors. This model is trained using a combination of the MSE and BCE loss functions.

This model is trained to minimize both the $LOSS_{class}$ and the $LOSS_{reg}$ (equation 5) where X defines the training data, θ_{class} defines the classification sub-model learning parameters, and θ_{reg} defines the regression sub-model learning parameters.

$$min_{\theta_{class},\theta_{reg}} \sum_{x \in X} LOSS_{class}(x, \theta_{class}) + LOSS_{reg}(x, \theta_{reg}) \tag{5}$$

- **AWS-EP:**

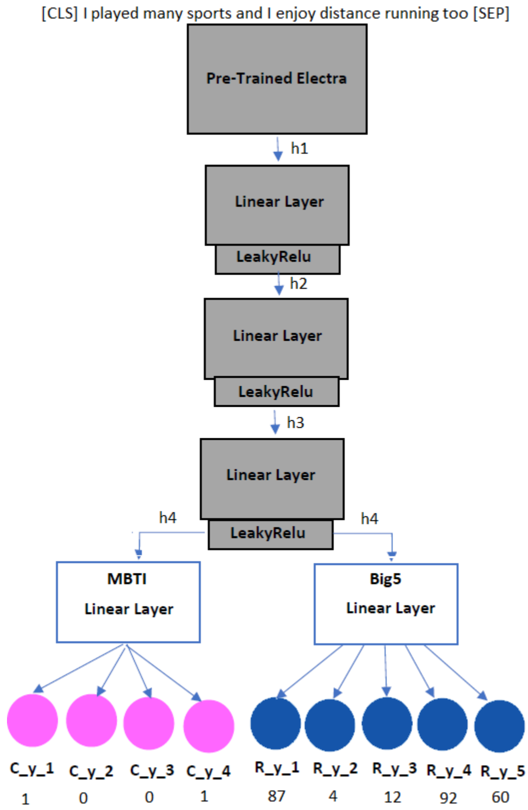

Figure 4: ALL Weights Shared Electra for Personality prediction Architecture

Similar to the previous approach, this model is designed to predict both classification (MBTI) and regression (Big5) tasks. However, instead of only sharing the Electra weights $h1$, this approach shares all the network weights (the pre-trained Electra weights $h1$ and the MLP network weights $h2$, $h3$, and $h4$) between the regression and classification heads. The two prediction heads are strongly dependent on each other as they both share the same weights except for the last layer weights. The white boxes represent the independent layers for each sub-AWS-EP architecture. The gray boxes represent the shared layers 'weights' between the classification sub-network and the regression sub-network. Similar to the previous EWS-EP, this model is trained using the same loss function and configuration. The only difference is that all the layers are shared except for the last two heads.

This model is trained under the objective of minimizing both the $LOSS_{class}$ and the $LOSS_{reg}$ (equation 6) where X defines the training data and $\theta_{class,reg}$ defines the model learning parameters. Unlike the previous model, which had two different model parameters, θ_{reg} and θ_{class} in AWS-EP we have only one model's parameters that combine both regression and classification weights.in AWS-EP, we have only one model's parameters that combine regression and classification weights. Combining the two losses into one loss helps the model focus on both tasks during the training phase.

$$min_{\theta_{class,reg}} \sum_{x \in X} LOSS_{class}(x, \theta_{class,reg}) + LOSS_{class}(x, \theta_{class,reg}) \tag{6}$$

All proposed baselines in section 3.1 are trained using the same hyper-parameters and share the same work pipeline. First, a sequence of words defined by a sentence start [CLS] and a sentence end [SEP] tokens is given as an input to the Electra base encoder (see the AWS-EP figure 3.1). The encoder will create a contextual vector representation for the sequence of words h1. Then the contextual embedding is passed to the MLP network, where we have different linear layers and normalization approaches. The shared MLP linear layers weights (h2, h3, and h4) are used to learn the optimal weights that effectively predict both personality factor tests. The last layers (Big5 and MBTI layers) are used as prediction heads. Given the h4 vector representation, both MBTI and Big5 linear layers try to predict the convenient values for each personality trait factor (example: [1,0,0,1] for the MBTI personality test and [87,4,12,92,60] for the Big5 personality test).

4 Experiments and results

Experiments and results are done using three different datasets. To investigate the performance of our proposed models, we used the Pandora dataset. This dataset combines both Big5 and MBTI features. To evaluate the different model's generalization performances, the MyPersonality Celli et al. (2013) and the MBTI datasets are used. The MyPersonality dataset is used for the Big5 features validation, and the Myers-Briggs Personality Type dataset is used to validate the MBTI features.

4.1 Datasets

- **Pandora dataset Gjurković et al. (2021) :** Pandora is the largest and the first dataset in the research field that contains more than 17 million Reddit comments written by more than 10k users annotated with both MBTI and Big5 factors with users' demographical features (age, gender, and location). 1.6k users are labeled with the Big5 personality model with more than 3M comments. It also comprises 9k users' annotations with the MBTI personality traits. Due to its massive amount of textual data, throughout this work, Pandora is used as the main dataset to train our baseline models.
  It is important to highlight that Pandora is a private dataset and the authors employ different terms of use to protect the users within this dataset Irina Masnikosa and Bakić (2020). Some of the terms consist of not transferring or reproducing any part of the dataset, attempting to identify any user in the dataset, contacting any user in the dataset, displaying users' names and sensitive messages publicly, reporting findings publicly unless it is at an aggregate level. The following two datasets are used as unseen data to evaluate the generalization of our proposed solutions.

- **MyPersonality dataset Celli et al. (2013) :** This dataset was collected in 2013 by Celli et al. It contains more than 250 different users with 10000 labeled Facebook statuses in total with the Big5 personality traits. It also combines network properties such as network size, density, transitivity, etc.

- **MBTI Personality Type dataset J (2017) :** This data was collected using the PersonalityCafe forum, as it displays a large selection of people and their MBTI personality type and what they have written. It contains more than 8k rows of data, where each row represents a different person. For each person, we have the last 50 texts they posted.

4.2 Training properties

The Pandora dataset is randomly partitioned into three parts during the training phase: training, validation, and test subsets. 20% of the data were considered a test set, and 80% were considered a training set. Then to create the validation data, we split the training set into two sub-parts. 20% were considered validation data, and the rest were kept to train the model, which is done using the scikit-learn library. The same data splitting process was done for all the different experiments using a seed value of zero. The sentence words were embedded into a 256-length token vector, using the pre-trained Electra-small model tokenizer from the pytorch hugging face framework. The pre-trained model was fine-tuned on the Pandora training sub-set, and all models (OC-EP, OR-EP, EWS-EP, AWS-EP) were trained for 10 epochs. We also compared the current validation results with the least validation loss for each epoch and stored the model that gave us the least generalization loss. In our experiments, we reported the performance of a single run (10 epochs) for each model. The hyper-parameters we used during our experiments are defined in table 4 in the appendix.

Different experiments were done to investigate the different generalization performance of the proposed baselines (OC-EP, OR-EP, EWS-EP, AWS-EP), and their performance was compared with state-of-the-art models on different datasets. We used the google collaboratory pro version as our computing infrastructure (166.83 Gb hard drive capacity, 25.46 GB memory capacity, and a 1 Tesla P100-PCIE GPU), which allows us to use a 20h window session of these computational resources.

4.3 Training Results

Training the four baselines using the same hyper-parameters led to different performance results on the training and the validation sets. Figure 6 in the appendix section A highlights the different training and validation performance for each baseline.
Training results show that EWS-EP and AWS-EP models (Multi-task models) have the highest trusted results in terms of generalization performance for the MBTI and Big5 traits. We can see that both are trying to reduce at the same time the training and validation loss during each epoch. This highlights the importance and the good performance of the multi-task learning approach compared to the single-task learning approach results. We can see that the validation error is almost constant along the training epochs for the OC-EP. So during the learning phase, this model is trying to decrease the training loss while keeping the validation loss almost the same. Therefore EWS-EP, and AWS-EP are better than the OC-EP on the classification generalization task.

4.4 Generalization Results

During the experiment phase, we were more curious about having effective results for predicting the Big5 and MBTI personality traits and investigating to which degree these two tests are similar. We also were curious to know the effect of weight sharing on the model predictions. Tables 1,2, and 3 highlight the generalization performance of the OC-EP, OR- EP, EWS-EP, and AWS-EP models. Table 1 highlights the performance of the proposed baselines (OC-EP, OR-EP, EWS-EP, AWS-EP) on the unseen Pandora test subset. The OC-EP model provides good performance for accuracy and F1-score metrics with 0.738 and 0.844, respectively. Results show that the OR-EP model provides an inferior performance in MSE, r2-score. By introducing a low level of weight sharing in the EWS-EP baseline, both classification and regression results improved. Moreover, allowing for more weight sharing between the MBTI and Big5 prediction tasks in the AWS-EP model significantly improved regression and classification results. It is also clear that the regression head is the one that benefits the most from the weight sharing with a more than 100% increase in terms of the Pearson r correlation metric compared to the OR-EP model. We also report a five-fold decrease in MSE compared to the OR-EP model.
The experiments demonstrate that the more we allow the Big5 prediction head to know and share weights with the MBTI model, the better results the head provides. This demonstrates the high correlation between the Big5 and MBTI personality test systems. The results provided in table 1 show that the most effective model from the 4 baselines is the AWS-EP model. For this reason, we aim to investigate the performance of this model further and evaluate its generalization performance. Tables 2 provide more details for the AWS-EP model performance for each trait factor compared to the Pandora baseline Gjurković et al. (2021) and PQ-Net Yang et al. (2021) baselines.

Our AWS-EP model outperformed the state-of-the-art benchmark of the Pandora paper for both MBTI and Big5 prediction tasks. For the MBTI classification task, we achieved a 0.1461 F1-score increase for the Introverted factor compared to the PQ-Net baseline. Also, we achieved a 0.278 increase for the Intuitive factor, a 0.0882 increase

	Classification				Regression		
Models	**Accuracy**	**Precision**	**Recall**	**F1-score**	**MSE**	**R2_score**	**P_r_C**
OC-EP	0.738	0.738	**1.0**	0.844	-	-	-
OR-EP	-	-	-	-	2910.39	-2.82	0.32
EWS-EP	0.739	0.739	**1.0**	0.845	839.03	0.05	0.47
AWS-EP	**0.788**	**0.792**	0.94	**0.860**	**564.12**	**0.35**	**0.66**

Table 1: The baselines performance on different metrics

Classification performance						
	Pandora	**PQ-Net**	**AWS-EP (Ours)**			
MBTI factors	**f1**	**f1**	**accuracy**	**precision**	**recall**	**f1**
Introverted	0.654	0.6894	0.7583	0.7629	0.9233	**0.8355**
Intuitive	0.606	0.6765	0.9131	0.9131	1.0	**0.9545**
Thinking	0.739	0.7912	0.7889	0.7939	0.9797	**0.8771**
Preceiving	0.642	0.6957	0.6916	0.7014	0.8625	**0.7736**
Average	0.6602	0.7132	0.7880	0.7928	0.9414	**0.8602**
Regression performance						
Metric	**Pandora**	**AWS-EP**				
P-r-C metric	0.2629	**0.66**				

Table 2: The AWS-EP detailed performance compared to the Pandora paper and the PQ-Net state-of-the-art models on the Pandora benchmark dataset

for the Thinking factor, and a 0.0847 increase for the Perceiving factor. Overall, we achieved a 0.147 F1 score average increase for all the MBTI factors compared to the PQ-Net state-of-the-art model. For the Big5 classification task, we achieved a 0.3971 increase in the Pearson correlation metric. Table 2 show that our AWS-EP model outperforms the state-of-the-art models on the Pandora benchmark dataset.

Despite the promising performance of our AWS-EP model on the Pandora dataset, we were curious to measure its generalization performance using different unseen perosnality datasets. The datasets we use in this experiment are the MBTI personality dataset from Kaggle J (2017), and the MyPersonality dataset Celli et al. (2013). Table 3 demonstrates the generalization performance of the AWS-EP model on different datasets that it has not been trained on.

As shown in table 3, the AWS-EP model performs exceptionally well on different unseen data. Although it was only trained on the Pandora dataset, this model outperforms state-of-the-art MBTI Kaggle datasets baselines. Without any tuning, our model outperformed state-of-the-art models on different datasets.

Moreover, this model also provides good Pearson r correlation (P-r-c) results for predicting the regression Big5 trait values on the unseen MyPersonality dataset with a 0.66 correlation value. However, for the Big5 classification task, our model provides a very poor performance compared to the state-of-the-art baseline on the same dataset. While The MyPersonality dataset baseline is a multi-label classification model, and it is trained to classify the Big5 traits categories, our Bi5 sub-model is a regression model. Hence, we cannot compare both model results because they operate on two different tasks. However, despite the good results of our AWS-EP model on the regression task, we were curious to know its performance on the classification task as zero-shot learning. To evaluate its Big5 classification performance, we took the predicted regression values and transformed them

	MBTI Kaggle	**MyPersonality**	
Metric	**F1**	**accuracy**	**P_r_C**
TrigNet	0.7086	-	-
PQ-Net	0.7132	-	-
BERT	-	**0.7210**	-
AWS-EP (Ours)	**0.8276**	0.487	**0.6624**

Table 3: AWS-EP generalization performance on different unseen datasets

into classification values (0 or 1) by applying a 50% threshold. This transformation did not surpass the MyPersonality state-of-the-art baseline trained on a classification objective. However, as a zero-shot prediction, the AWS-EP results are promising. Also, this highlights the need to add a new Big5 classification head to the AWS-EP model.

5 Ethical impact of our work

Despite the vast benefits of knowing the user's personality on his/her daily life services, having the individual personality traits without his/her permission or explicitly indicating his/her personality to us can be unacceptable. We believe that attempting to detect the individual personality can be a personality intrusion. Knowing the individual's personality can help us know his/her preference, his/her behavior and his/her social relationship with others, etc. If the user did not consent to us knowing all stated information, then knowing them is simply a privacy intrusion. Moreover, acquiring such information about the users can lead to mental and physical harm. Knowing what the user likes or dislikes can easily affect him/her and can be detrimental either mentally or physically (for example, manipulating the user to do something dangerous). These are the main reasons why the Pandora dataset (Gjurković et al. (2021)) is not a public dataset, and to use it, you need to submit a request explaining why you are seeking the use of this dataset. Also, the authors of this dataset employ rigorous terms of use (Irina Masnikosa and Bakić (2020)) to protect the users within the dataset. For example, one cannot transfer or reproduce any part of the dataset and attempt to identify or contact any user in the dataset. One cannot publicly display users' names and sensitive information and messages. Also, one can report findings publicly only on an aggregate level. We believe that the user has the right to keep his/her personality private. Whether personality is consciously or unconsciously revealed in any way, it is the other person's responsibility to act diligently and protect the shared information to prevent from putting anybody in harm's way. Therefore, our work does not expose any users' private information, and we do not take users' unique identifiers or demographic information to predict their personalities. Our predictive model only focuses on the posted users' social media textual contents. In other words, we do not focus on " who" posted the content but rather on the content itself. Using only the textual content to predict the individual's personality helps us effectively reduce privacy intrusion risks. Our work is extremely valuable and can improve many service providers. Only using the content of the users' posted texts without employing specific users' information helped us reduce the privacy intrusion issues. However, we think that our model is limited in providing compelling encrypted personality predictions. For now, our model only predicts the personality traits in their original forms. However, it would be more secure in predicting them in an encrypted way. Therefore, we aim to enhance the capability of our model by introducing an encryption mechanism for the predicted results. We believe that it is essential for our personality predictive model to be used in the right, protected, and secured environment that does not harm the users or reveal their personalities in any way.

6 Conclusion

This work highlighted the effectiveness of using a multi-task learning approach on top of a pre-trained Electra model for the personality prediction task. We also highlighted the effect of sharing weights between the two popular personality trait tests. Empirical results demonstrate that using shared weights between MBTI and Big5 personality tests outperforms state-of-the-art results for both systems on different metrics. Our results show that both personality systems are correlated. Also, we found that despite the good Big5 regression results of our solution, it seems like our model is incapable of effectively classifying the Big5 traits from the regression values. More weight sharing, contextual information, and prediction heads will be considered in future work as we are curious to know the effect of demographical information such as age, gender, and country on personality detection.

References

Giulio Carducci, Giuseppe Rizzo, Diego Monti, Enrico Palumbo, and Maurizio Morisio. 2018. Twitpersonality: Computing personality traits from tweets using word embeddings and supervised learning. *Information (Switzerland)*, 9.

Fabio Celli, Fabio Pianesi, David Stillwell, and Michal Kosinski. 2013. Workshop on computational personality recognition: Shared task.

Mark Batey David John Hughes, Moss Rowe, Giuseppe Riccardi Andrew Lee, and Fabio Pianesi. 2012. A tale of two sites: Twitter vs. facebook and the personality predictors of social media usage. *Computers in Human Behavior*, pages 561–569.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Joan-Isaac Biel Daniel Gatica-Perez Giuseppe Riccardi Fabio Celli, Bruno Lepri and Fabio Pianesi. 2014. The workshop on computational personality recognition. *ACM*, pages 1245–1246.

Adrian Furnham. 1996. The big five versus the big four: the relationship between the myers-briggs type indicator (mbti) and neo-pi five factor model of personality. *Personality and Individual Differences*, pages 303–307.

Matej Gjurković, Mladen Karan, Iva Vukojević, Mihaela Bošnjak, and Jan Snajder. 2021. PANDORA talks: Personality and demographics on Reddit. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 138–152, Online. Association for Computational Linguistics.

Lewis R Goldberg. 1993. The structure of phenotypic personality traits. *American psychologist*, pages 26–34.

Ivan Crnomarković Jan Šnajder Josip Jukić Matej Gjurković Mihaela Bošnjak Mladen Karan Irina Masnikosa, Iva Vukojević and Sara Bakić. 2020. Pandora. Preprint at `https://psy.takelab.fer.hr/datasets/all/pandora/` Last visited 09-01-2022.

Mary H. Mc Caulley Isabel Briggs Myers and Allen L. Hammer, editors. 1987. *Introduction to Type: A description of the theory and applications of the Myers-Briggs type indicator*. Consulting Psychologists Press.

Mitchell J. 2017. (mbti) myers-briggs personality type dataset. Preprint at `https://www.kaggle.com/datasnaek/mbti-type` Last visited 1-06-2022.

Quoc V. Le Kevin Clark, Minh-Thang Luong and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ArXiv*, abs/2003.10555.

Yang Li, Amirmohammad Kazameini, Yash Mehta, and E. Cambria. 2021. Multitask learning for emotion and personality detection. *ArXiv*, abs/2101.02346.

Manjula Ramannavar Mayuri Pundlik Kalghatgi and Nandini S Sidnal. 2015. A neural network approach to personality prediction based on the big-five model. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, pages 56–63.

Simine Vazire Nicholas Holtzman Samuel D Gosling, Adam A Augustine and Sam Gaddis. 2011. Manifestations of personality in online social networks: Self-reported facebook-related behaviors and observable profile information. *Cyberpsychology, Behavior, and Social Networking*, pages 483–488.

Ruth E Appel Sandra C Matz and Michal Kosinski. 2020. Privacy in the age of psychological targeting. *Current opinion in psychology*.

Haolan Ouyang Xiaojun Quan Tao Yang, Feifan Yang. 2021. Psycholinguistic tripartite graph network for personality detection. *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

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

Michal Kosinski Wu Youyou and David Stillwell. 2015. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the national academy of sciences*, pages 1036–1040.

Feifan Yang, Tao Yang, Xiaojun Quan, and Qinliang Su. 2021. Learning to answer psychological questionnaire for personality detection. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1131–1142.

Amir mohammad Kazameini Clemens Stachl Erik Cambria Yash Mehta, Samin Fatehi and Sauleh Eetemadi. 2020. bottom-up and top-down: Predicting personality with psycholinguistic and language model features. *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1184–1189.

Bee Chin Ng Yosephine Susanto, Andrew Livingstone and Erik Cambria. 2020. The hourglass model revisited. *in IEEE Intelligent Systems*, pages 96–102.

A Appendix

A.1 Electra model

Electra is a new pre-training approach that aims to match or exceed the MLM (Masked Language Modeling) pre-trained model downstream performance while using significantly less compute resources for the pre-training process.

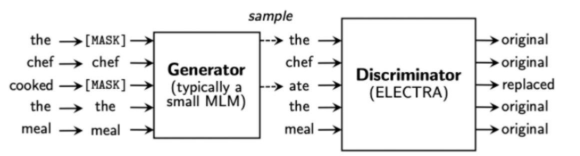

Figure 5: Electra Architecture Kevin Clark and Manning (2020)

Unlike BERT which heavily relies on the MLM approach during the pre-training phase, this model uses a new training approach called the replaced token detection approach. Figure 5 highlights the Electra model architecture. The Electra architecture combines both a generator and discriminator components. The generator will be trained using the MLM goal and the discriminator will try to predict for each word provided by the generator whether it has been replaced or not. Therefore instead of only knowing the 15% masked words in the sentence as BERT does, this model will have knowledge of all the tokens within the sentence and predict whether it is the original token or the replaced one. Having knowledge about all the words instead of only 15% of them gives the Electra model much more insights about the context within a group of words. Moreover, using the discriminator as a binary classifier to predict whether the word has been replaced or not will help the model gain time during the training phase. As binary classification is less computationally expensive compared to the word generation task. To effectively train this model the authors propose two losses, one for the generator L_{MLM} (equation 7), and one for the Discriminator L_{Disc} (equation 8).

$$L_{MLM}(x, \theta_G) = E(\sum_{i \in m} - \log p_G(x_i / x^{masked})) \tag{7}$$

θ_G is the generator learning parameters, x_i is the current token input and x^{masked} is the replacement tokens vector.

$$L_{Disc}(x, \theta_D) = E(\sum_{t=1}^{n} -1(x_t^{corrupt} = x_t) \log D(x_t^{corrupt}) - 1(x_t^{corrupt} \neq x_t) \log(1 - D(x_t^{corrupt}))) \tag{8}$$

θ_D defines the discriminator learning parameters, 1 defines the indicator function and $x_t^{corrupt}$ defines the replaced token. To train both the generator and the discriminator in an END-2-END process the authors combined both losses into a single loss function (equation 9) with the addition of a new penalty term λ for the discriminator loss.

$$min_{\theta_G, \theta_D} \sum_{x \in X} L_{MLM}(x, \theta_G) + \lambda L_{Disc}(x, \theta_D) \tag{9}$$

Table 4: The models hyperparameters

Hyper-Parameter	**Value**
Epochs number	10
Optimizer	Adam optimizer
Learning rate	2e-5
Weight decay	0.01
Activation function	LeakyRelu
Dropout degree	0.4
Classification loss (CL)	BCE with logits loss
Regression loss (RL)	MSE loss
Global loss	(CL+RL)/2
Batch size	15
Trainable parameters	13542167

A.2 The training hyperparameters

Supplementary Material

Datasets supplementary material:

MyPersonality dataset
MBTI Personality Type dataset
Big Five personality traits explanation
MBTI personality traits explanation
Pendora dataset request platform
Models card supplementary material:

OC-EP model card
OR-EP model card
EWS-EP model card
AWS-EP model card
AWS-EP model code

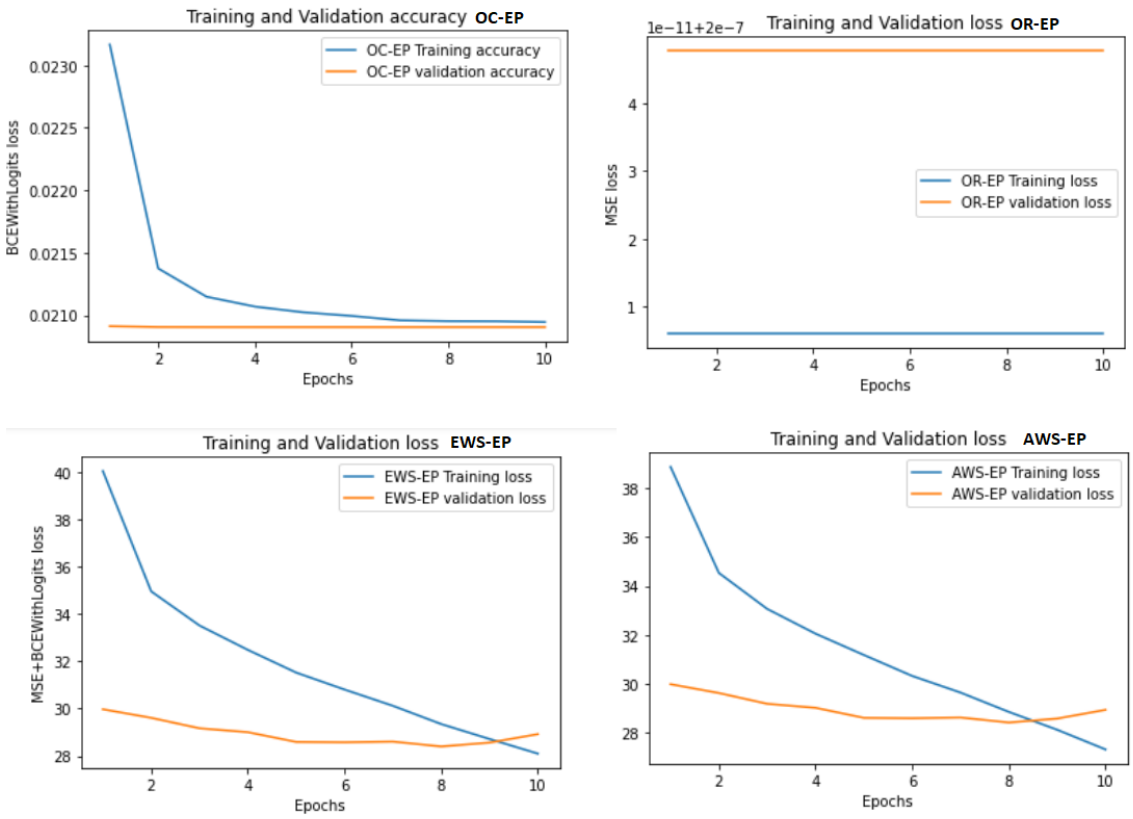

Figure 6: OC-EP, OR-EP, EWS-EP, and AWS-EP training performances