



A General Personality Analysis Model Based on Social Posts and Links

Xingkong Ma¹(✉), Houjie Qiu¹, Shujia Yao², Xinyi Chen¹, Jingsong Zhang¹,
Zhaoyun Ding¹, Shaoyong Li¹, and Bo Liu¹

¹ College of Computer, National University of Defense Technology, Changsha, China

{maxingkong, qiuhoujie, xinyi_chen,
zhangjingsong17, zyding, shaoyongli}@nudt.edu.cn

² Shanxi University, Taiyuan, China

Abstract. Personality plays a vital role in psychological feature analysis, product recommendation, and mental health assessment. Analyzing personality based on social networks is becoming mainstream since it allows collecting user behaviors and continuously output personality prediction results in a non-intrusive manner. However, existing methods face either over-fitting problems due to the small-sized training datasets or inaccurate feature representation due to the limited information of the testee. This paper proposes a general personality analysis model based on posts and links in social networks, called GPAM. To solve the problem of insufficient training data, we use a user linkage technique to collect large-scale and high-quality labeled personality data in a short time. By introducing posts from high-influence friends, we propose a unified personality feature extraction model to represent the users without enough information. Under various parameter settings, the experimental results demonstrate that importing moderate posts from high-influence friends benefits state-of-the-art models. The average f1-scores of predicting both MBTI and Big Five in GPAM are higher than the latest model Trignet. Compared to without introducing extra posts, the average f1-scores of in GPAM improve at least 4% for wordless users and 51% for silent users.

Keywords: Personality analysis · Big five · MBTI · User linkage · Personality feature extraction

1 Introduction

Personality is the characteristic sets of behaviors, cognitions, and emotional patterns that evolve from biological and environmental factors [1]. Since personality is relatively stable, it plays a vital role in diverse fields, such as recruitment, counseling, personalized advertising, recommendation, mental health assessment, etc. For instance, Personality tests have become a recruitment trend in recent years. Data source from the Society for Industrial and Organizational Psychology [2] displays that 29% of employers use one or more forms of psychological measurement or assessment, and 13% of employers use personality tests. According

to Psychology Today [3], around 80% of Fortune 500 companies use personality tests to assess their employees. Another example is the recommendation. Compared with content filtering or collaborative filtering, personality-aware recommendation systems solve the problems of the cold start and data sparsity [4] and have been applied to the recommendation of musics [5], books [6], etc.

Psychologists propose various models to describe the individual personality. Currently, two personality measurement models are considered to be reliable and operable. One is the Big Five model. It describes the personality trait using five dimensions: **O**penness, **C**onscientiousness, **E**xtraversion, **A**greeableness, and **N**euroticism. The adjective definers of these dimensions can be found in [7]. The other is Myers Briggs Type Indicator (MBTI) [8]. It describes the personality from four dimensions of how a person interacts with the world (**E**xtraversion versus **I**ntroversion), gathers information (**S**ensing versus **i**Ntuition), processes information (**T**hinking versus **F**eeling), and makes decisions (**J**udging versus **P**erception).

To evaluate individual personality, psychologists provide well-designed questionnaires to testees. This method has two disadvantages. Firstly, the answers to questionnaires are probably untruthful since testees tend to conceal their personality defects because of privacy protection. Secondly, it is difficult to expand to a large scale since the costs of time, human resources and money significantly increase with the growing number of testees.

Analyzing personality based on social networks has become a prevailing trend in recent years. However, most existing methods face the following two challenges. The first is the lack of labeled training data. Although several datasets [9–12] have been published on Internet, their sizes are small and the labels are doubtful, which leads to inadequate training and over-fitting problem. The second is that many users neither fill out their profiles nor frequently express themselves on social networks. It is hard to extract features from these users, which leads to inaccurate personality prediction.

To this end, we propose a general personality analysis model based on posts and links in social networks called GPAM. Generally speaking, we provide the following contributions: (1) We adopt a user linkage method to correlate the same person on different websites to collect labeled data. It allows to collect large-scale and high-quality training data quickly. (2) We propose a unified personality extraction model to extract features from users without enough posts. (3) We implement extensive experiments to verify the performance of GPAM under various parameter settings.

2 Related Works

Social networks encompass a large number of user information, such as age, gender, emotional state, address, education, posts, comments, friends, etc. Many researchers try to build a connection between social networks and personalities.

The first category is based on user expression. Pennebaker et al. [13] develop LIWC, a computerized text analysis program that outputs the percentage of words

in a given text that falls into different psychological categories [14]. LIWC enlightens researchers to establish a linkage between linguistic patterns and personality or psychological state. Yang et al. [15] propose a recommending algorithm to players according to their identified personality traits. They compute the Pearson’s Correlation Coefficients between the OCEAN personality traits and LIWC. Thus, the algorithm recommends games based on both user-user personality similarity and game-user personality similarity. SIMPA [16] detects self-referencing descriptions of personality in a target’s text and utilizes these descriptions for personality assessment. Because of the ability to automatically extract features from texts, many researchers adopt deep learning methods to predict personality traits. HIE [17] first integrates heterogeneous information, including self-language usage, avatar, emoticon, and responsive patterns, then extracts semantic features through LIWC and Text-CNN. 2CLSTM [18] extracts user personality features by using LSTM concatenating with CNN. To avoid the post-order bias, TransformerMD [19] proposes a post-order-agnostic encoder to put together the posts of a user to depict an overall personality profile. To exploit psycholinguistic knowledge, Trignet [20] constructs a heterogeneous tripartite graph by injecting structural psycholinguistic knowledge from LIWC, and proposes a flow graph attention network to obtain the embedding of posts. To alleviate the impact of polysemy in the personality detection tasks, SEPRNN [21] combines word embedding with contextual information to obtain precise semantics for words.

The second category is based on user profiles. Golbeck et al. [22] collect personal profiles of 279 Facebook users. The authors build a correlation between user attributes and the Big Five personality. Gu et al. [23] collect over six thousand profiles on Weibo in China. The results show that with the growth of age, the scores of conscientiousness and agreeableness increased, and openness and extroversion decreased. Besides, Wald et al. [24] analyzed the Big Five personality traits of Facebook users by using 31 profile attributes and 80 post attributes.

The third category is based on user behavior. Chittaranjan et al. [25] collects the usage data of 117 Nokia N95 smartphone users for 17 months. By extracting features from the logs of calls, short messages, Apps, Bluetooth, and profiles, they adopt multiple regression analysis techniques to analyze the correlation between the terminal data and personality. TECLA et al. [26] predicts temperaments and psychological types based on linguistic and behavioral analysis of Twitter data.

In conclusion, most existing methods do not consider two important issues that impact the performance of personality models. One is the small-sized training datasets. The other is the limited posts of testees. In GPAM, we propose a user linkage method and a unified personality extraction model to solve these issues.

3 Data Collection

The quantity and quality of labeled training data significantly affect the training and prediction of the personality model. As far as we know, there are mainly three data collection methods in existing works.

- The first is inviting social network users to answer questionnaires online and then crawling the social data of these users. Similar to the offline questionnaires, it is hard to extend to a large scale because of the privacy protection.
- The second is crawling the social data of the user who provides his/her Big Five score or MBTI type in the profile or posts [11]. Since the crawler has to search for users from the whole social network, the searching process leads to much time and resource costs.
- The third is crawling the user comments from personality forums like PersonalityCafe [27]. Users in these forums mainly talk about the behaviors or feelings of their personalities. Even if well-behaved personality prediction models are trained based on these discussions, they are not applicable to daily talking, including topics of economy, politics, society, living, etc.

To obtain a large-scale and high-quality labeled personality dataset, our basic idea is to link the same person from both personality websites and social networks. To increase the accuracy of user linkage, we choose famous persons as our targets. There are two reasons. The first is the personality types of famous persons easy to be collected from their fans or personality websites. The second is most famous persons ensure the authenticity of their social accounts through the real-name authentication system.

We firstly crawl the personality types of the Big Five and MBTI of famous persons from Personality-Database [28]. Note that ordinary people vote for these personality types. To avoid the wrong labeled personality type, we check the vote count over the threshold value. Secondly, since the famous person’s nickname is the same as the real name, we can search for the real name and get the corresponding social network account from Facebook or Twitter with a high probability. Following the policy of Twitter API or Facebook API, it is easy to obtain each famous person’s profile, posts, and links. Thus, we can collect both personality labels and social data from famous persons within a short time.

4 Personality Representation

As mentioned in Sect. 2, existing works extract personality features from user profile, expression, behavior, etc. However, a report from Twopcharts [29] shows that 44% of Twitter accounts have never sent a tweet, 30% of the accounts have sent 1–10 tweets, and only 13% of the accounts have written at least 100 tweets. Therefore, it is hard to collect enough data from most users, which leads to inaccurate feature representation.

Based on existing researches [30], personality type compatibility exists among individuals. Thus, we believe that introducing extra posts from high-influence friends to the users without enough posts is reasonable. There are two problems we need to solve. The first is how to measure the influence of each friend in the view of personalities. To this end, we propose an interaction-based influence sorting algorithm in Sect. 4.1. The second is how to fuse the personalities of high-influence friends into the testee’s personality. To this end, we propose a unified feature extraction model in Sect. 4.2. Table 1 shows key notations used in this section.

Table 1. Key notations in GPAM

Parameters	Meanings
N_v	The number of feature vectors sampled from a user
N_{sp}	The number of sampled posts at a time
N_{xh}	The maximum number of high-influence friends
N_{nh}	The minimum post number of a high-influence friend publishes
N_{np}	The maximum post number a silent or wordless user publishes
N_{nr}	The maximum post number a silent or wordless user reserves
N_p	The post number of a user published

4.1 Interaction-based Influence Sorting

To select high-influence friends, we propose an interaction-based influence sorting algorithm (IISA) in this section. In specific, this algorithm follows three rules:

- Rule 1: Selecting following but not followers. For a testee, his/her following have much more influence than followers.
- Rule 2: Selecting the following who is mentioned in the posts of the testee. One may argue that why do not select the following whose post is given a like or commented by the testee. Theoretically, we are able to collect all posts from the following of the testee. However, it costs much time and resources in practice.
- Rule 3: Selecting the following with a large number of posts. Since the testee receives posts from the following, we suppose the influence of the following is in direct proportion to the number of posts.

Algorithm 1: *Interaction-based Influence Sorting*

```

Input : FOWL: the following list.
Output: HIFL: the high-influence friend list.
; /* $N_m$ : the mentioned times by the testee. */
1 for each  $i$  in  $[0, FOWL.size()-1]$  do
2   if mentioned(testee, FOWL[i]) and  $(FOWL[i].N_p \geq N_{min})$  then
3     HIFL.append(FOWL[i])
4 if HIFL.size()  $\geq N_{xh}$  then
5   sort HIFL by  $N_m$  of each item;
6   HIFL.remove( $N_{xh}$ , HIFL.size()-1);
7 else
8   sort FOWL by  $N_m$  of each item;
9   for each  $i$  in  $[0, FOWL.size()-1]$  do
10     if  $(HIFL.size() \leq N_{xh}$  and  $FOWL[i].N_p \geq N_{min}$  and
11       FOWL[i]  $\notin$  HIFL) then
12       HIFL.append(FOWL[i])

```

The detailed process of IISA is shown in Algorithm 1. Firstly, if one following is mentioned by the testee, and his/her post number is bigger than the threshold

N_{min} , the following is appended to the high-influence friend list (Line 1–3). Note that we filter out the following with low activity, whose features are hard to be extracted, as mentioned at the beginning of this section. Secondly, if the size of the high-influence list is bigger than the threshold N_{xh} , we sort the list by the mentioned times and only keep the top N_{xh} items (Line 4–6). Thirdly, if the size of the high-influence list is less than N_{xh} , we sort the following by the number of posts and append the following with a bigger number of posts into the high-influence list (Line 7–12).

Take Fig. 1 as an example. Alice follows four friends, publishes three posts, and mentions Bob two times and Denise one time. Suppose $N_{min} = 100$ and $N_{xh} = 2$ in Algorithm 1, Bob and Denise are picked based on Rule 2. Suppose $N_{xh} = 3$, Bob, Denise and Eva are picked based on Rule 2 and 3.

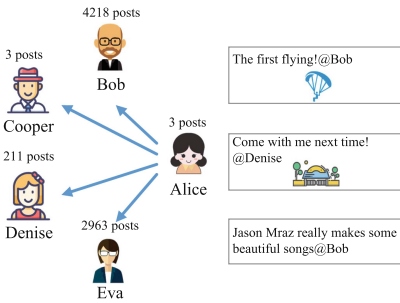


Fig. 1. Case of sorting friend influence

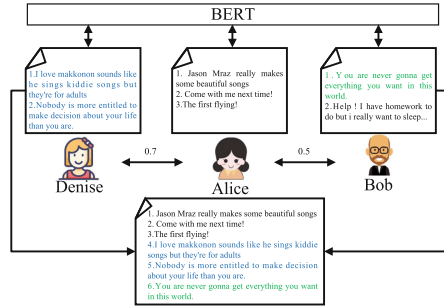


Fig. 2. Case of unified feature extraction

4.2 Unified Feature Extraction

In this section, we propose a unified feature extraction model. The basic idea is to fuse the personalities of high-influence friends into users without enough posts. We classify all users into three types. The first is the *silent user*, who does not publish any posts. The second is the *wordless user*, whose post number is between 1 to N_{np} , where N_{np} is a fixed threshold value. The third is the *active user*, whose post number is bigger than N_{np} .

The detailed process is shown in Algorithm 2. For each silent user, we pick posts from his/her high-influence friends based on their number of posts (Line 1–6). For a wordless user, we use the Bert model [31] to extract features (Line 8–10). Note that posts of each user are sampled into multiple groups, and a fixed-length vector represents each group. It brings two advantages. One is to avoid the vanishing gradient problem of long text, and the other is to increase the training samples. Next, the similarity weight between the wordless user and each high-influence friend is computed based on the maximum cosine distance among their feature vectors (Line 11–13). Each high-influence friend contributes a part of the posts to the testee based on the similarity until the total number

of posts reaches the threshold N_{nr} (Line 14–17). Finally, the feature vectors of the testee are updated based on the new post list (Line 18). Note that active users do not append extra posts from friends in IISA.

Algorithm 2: *Unified Feature Extraction*

```

Input : HIFL: the high-influence friend list of the testee.
Output: testee.vec: the list of feature vectors.
1 if (testee.Np == 0) then
2   for each i in [0, HIFL.size()-1] do
3     TotalPosts += HIFL[i].Np;
4   for each i in [0, HIFL.size()-1] do
5     posts = pickPosts(HIFL[i].posts, HIFL[i].Np / TotalPosts * Nnr);
6     testee.posts = testee.posts ∪ posts;
7 if (testee.Np > 0 and testee.Np < Nnr) then
8   testee.vec = sample(testee.posts, Nsp);
9   for each i in [0, HIFL.size()-1] do
10    HIFL[i].vec = sample(HIFL[i].posts, Nsp);
11   for each i in [0, HIFL.size()-1] do
12     Sim[i] = cos(HIFL[i].vec, testee.vec);
13     TotalSim += Sim[i];
14   NeedPostNum = Nnr - testee.Np;
15   for each i in [0, HIFL.size()-1] do
16     posts = pickPosts(HIFL[i].posts, Sim[i] / TotalSim * NeedPostNum);
17     testee.posts = testee.posts ∪ posts;
18 testee.vec = sample(testee.posts, Nsp);

```

Take Fig. 2 as an example. Suppose Alice is a wordless user, and her high-influence friends are Denise and Bob. Firstly, each user’s posts are transformed into a group of feature vectors through the Bert model. Secondly, we compute the similarities between Alice and her friends based on their feature vectors. Thirdly, based on their similarities, two posts from Denise and one post from Bob are appended to the post list of Alice. Finally, updated posts of Alice are transformed into new feature vectors through the Bert model.

5 Personality Model Training and Testing

The quantity of the labeled data greatly affects the training accuracy. According to the Algorithm 2, posts of each user are sampled and extracted as a group of fixed-length vectors, each of which is treated as a training or testing sample. By default, the sampling frequency of each user is in direct proportion to the number of posts.

We use multiple classifiers like SVM, XGBoost and Random Forest to train Big Five and MBTI models. Since the prediction results of different testing items may represent the same user, we use these prediction results to vote for the final label. As shown in Fig. 3, Denise has three testing vectors, each of which is classified into a personality type. Take MBTI for instance. These three vectors are classified into INFP, INTP, and INTJ types. After voting on each dimension, INTP is treated as the MBTI type of Denise.

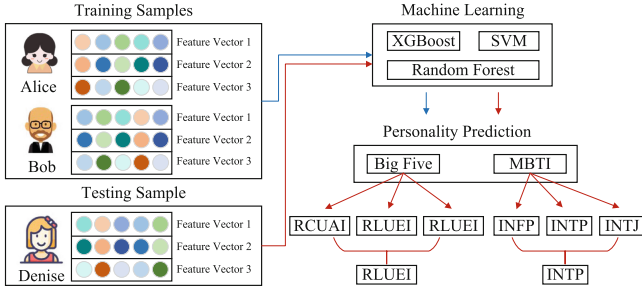


Fig. 3. Personality model training and testing

6 Experiment

6.1 Datasets

According to the user linkage method in Sect. 3, we collect 2007 users from Personality Database and Twitter. Although there are 16 personality labels in MBTI, we can build four binary classifiers rather than a multiple classifier with 16 labels, which brings higher classifying accuracy. Similarly, we build five binary classifiers for Big Five predication. Besides, the maximum gap of the label counts under the same dimension in MBTI and Big Five is not significant, which is propitious to build the classifiers.

Since we select high-influence friends through IISA in Sect. 4.1, it is critical to know the distributions of posts, following, and mentioned following in our dataset. According to the statistics, the post counts of 1.3% of users are zero, 8.3% of users are less than 50, and over 74% users are larger than 1000. The following counts of 3.7% of users are zero, 12.7% of users are less than 25, and over 73% users are larger than 100. The mentioned user counts of 15.8% of users are zero, 22.1% of users are less than 25, and over 60% users are larger than 100. In general, the distributions of numbers of post, following, and mentioned users are wide enough to verify the effectiveness of the feature extraction in Sect. 4.

6.2 Implementation

We deploy GPAM in a private server equipped with 24 processor cores, 64 GB memory, and a NVIDIA V100 GPU to reduce the training latency. For feature representation, we implement Doc2Vec and Bert. Both of them can transform texts into fixed-length vectors. For classification, we implement SVM, Random-Forest(RF), and XGBoost, which have been widely applied in the research and industry fields.

For the users with a large number of posts, we sample and transform these posts into multiple feature vectors as mentioned in Sect. 4.2. In the implementation, the sampling frequency is proportional to the number of posts for each user, and the detailed parameters are described in Sect. 6.3.

Remember that we propose an interaction-based influence sorting algorithm (IISA) in Sect. 4.1. In comparison, we implement another two strategies. One is the following with most posts first (MPF), and the other is the following with most followers first (MFF).

6.3 Parameters and Metrics

We measure GPAM under various parameter settings. To extract features from users, the number of sampled posts from a user at a time N_{sp} is set to 10, and the number of feature vectors sampled from a user N_v ranges from 1 to 20. One may argue that why do not increase N_v linearly with the number of posts. This is because it probably leads to unbalanced labels during training. For silent and wordless users mentioned in Sect. 4.2, the maximum number of posts they published N_{np} is set to 50. To append posts, the maximum number of high-influence friends of each user N_{xh} ranges from 5 to 20, the minimum post number of a high-influence friend publishes N_{nh} is set to 100. The maximum post number a silent or wordless user reserves N_{nr} is set to 100.

To evaluate the performance of GPAM, we compute average accuracy (AvgAcc), average precision (AvgPre), average recall (AvgRecall), and average f1-score (AvgF1) of all dimensions of MBTI and Big Five in each experiment.

6.4 Baseline Performance

This section tests the baseline performance of both MBTI and Big Five under various feature representation models and classification models. Our testees are users whose vote counts are larger than five and post counts are larger than 50. We train their posts and evaluate the performance of GPAM as the baseline. Besides, the state-of-the-art method Trignet [20] is also evaluated as a comparison. Specifically, we sample posts of users based on the parameters of N_v and N_{sp} in Sect. 6.3.

Table 2. Baseline performance of MBTI

Model	Avg. Acc	Avg. Pre	Avg. Recall	Avg. F1
Doc2Vec-SVM	61.88%	64.57%	64.57%	63.46%
Doc2Vec-RF	59.64%	62.20%	59.34%	56.91%
Doc2Vec-XGBoost	60.48%	63.07%	63.10%	62.22%
Bert-SVM	63.31%	66.80%	62.88%	60.94%
Bert-RF	62.50%	65.85%	63.91%	63.49%
Bert-XGBoost	62.39%	65.32%	65.34%	64.53%
Trignet	61.82%	59.91%	60.82%	59.95%

In the experiment of MBTI models, there are 10427 items for training, and 2075 items for testing. As shown in Table 2, the Bert-SVM model has the best AvgAcc (63.31%) and AvgPre (66.80%), and the Bert-XGBoost model has the best AVGRecall (66.55%) and AvgF1 (64.53%). For Doc2Vec and Bert, their average values of AvgF1 are 60.92% and 62.99%, respectively. For SVM, RF, and XGBoost, their average values of AvgF1 are 62.20%, 60.20%, and 63.38%, respectively.

In the experiment of the Big Five models, there are 6556 items for training and 992 items for testing. As shown in Table 3, the Bert-SVM model has the best AvgAcc (64.80%), AvgRecall (91.19%), and AvgF1 (75.23%), and the Bert-RF model has best AccPre (65.96%). For Doc2Vec and Bert, their average values of AvgF1 are 73.67% and 73.47%, which is 4% higher than Trignet on average. For SVM, RF, and XGBoost, their average values of AvgF1 are 74.39%, 73.51%, and 72.82%, respectively.

In general, GPAM show better performance than Trignet under different parameters. Bert shows slightly better performance than Doc2Vec on average since users the bidirectional transformer to solve the problem of polyseme. Besides, SVM and XGBoost offer marginally better performance than RF on average.

Table 3. Baseline performance of big five

Model	Avg. Acc	Avg. Pre	Avg. Recall	Avg. F1
Doc2Vec-SVM	64.26%	65.02%	85.47%	73.55%
Doc2Vec-RF	63.71%	64.04%	89.27%	74.25%
Doc2Vec-XGBoost	63.79%	64.91%	84.87%	73.22%
Bert-SVM	64.80%	65.06%	91.19%	75.23%
Bert-RF	64.06%	65.96%	81.95%	72.77%
Bert-XGBoost	63.68%	65.72%	81.30%	72.42%
Trignet	61.52%	65.60%	74.40%	69.52%

6.5 Impact of High-influence Friend Selection Strategies

In this section, we test the performance of both MBTI and Big Five models under three high-influence friend selection strategies, MFF, MPF, and IISA, mentioned in Sect. 6.2.

Table 4. Impact of high-influence friend selection strategies in MBTI

Strategy	Model	Avg. Acc	Avg. Pre	Avg. Recall	Avg. F1
MFF	Bert-SVM	63.89%	64.66%	58.66%	59.57%
	Bert-RF	62.70%	62.79%	60.17%	60.62%
	Bert-XGBoost	62.32%	61.96%	62.12%	61.73%
	Trignet	61.86%	60.02%	59.53%	59.50%
MPF	Bert-SVM	63.89%	64.42%	58.36%	59.48%
	Bert-RF	62.42%	62.75%	59.56%	60.34%
	Bert-XGBoost	62.13%	61.91%	61.90%	61.55%
	Trignet	60.43%	59.97%	58.04%	58.70%
IISA	Bert-SVM	63.93%	65.55%	60.11%	62.98%
	Bert-RF	63.20%	63.68%	60.50%	61.14%
	Bert-XGBoost	63.17%	63.22%	67.18%	65.29%
	Trignet	60.92%	59.34%	66.55%	62.42%

Table 5. Impact of high-influence friend selection strategies in big five

Strategy	Model	Avg. Acc	Avg. Pre	Avg. Recall	Avg. F1
MFF	Bert-SVM	64.93%	66.24%	90.81%	75.89%
	Bert-RF	64.45%	67.27%	82.18%	73.69%
	Bert-XGBoost	63.09%	67.03%	78.56%	72.11%
	Trignet	62.99%	66.95%	78.17%	72.02%
MPF	Bert-SVM	64.91%	66.02%	91.29%	76.01%
	Bert-RF	64.19%	67.18%	82.25%	73.62%
	Bert-XGBoost	62.93%	66.81%	78.83%	72.15%
	Trignet	62.00%	67.97%	74.83%	70.41%
IISA	Bert-SVM	65.02%	65.77%	91.78%	76.18%
	Bert-RF	65.05%	67.59%	83.08%	74.31%
	Bert-XGBoost	63.95%	67.23%	80.52%	73.15%
	Trignet	63.33%	68.68%	74.58%	71.40%

In the experiment of MBTI models, there are 1397 users in total, containing 396 wordless users. After introducing posts from high-influence friends, the training item sizes of MFF, MPF and IISA are 10090, 10105, and 8662, respectively, and the testing item sizes of MFF, MPF, and IISA are 3637, 3635, and 2325 respectively. As shown in Table 4, the Bert-SVM model using IISA has the best AvgAcc (63.93%) and AvgPre (65.55%), and the Bert-XGBoost using IISA has the best AvgRecall (62.18%) and AvgF1 (62.29%). Compared with the baseline in Table 2, the average AvgF1 of IISA in Table 4 increases slightly (62.98% vs. 63.13%, and 59.95% vs. 62.42%).

In the Big Five models experiment, there are 829 users in total and 177 users whose post numbers are less than 50. After post-transfer, the training item sizes of MFF, MPF, and IISA are 5942, 5932, and 6035, respectively, and the testing item sizes of MFF, MPF, and IISA are 2279, 2283, and 2281, respectively. As shown in Table 5, the Bert-RF model using IISA has the best AvgAcc (65.05%), Trignet using IISA has the best AvgPre (68.68%), and the Bert-SVM using IISA has the best AvgRecall (91.78%) and AvgF1 (76.18%). Compared with Table 3, the average AvgF1 of IISA and Trignet in Table 5 increases slightly (73.47% vs. 74.55%, and 69.52% vs. 71.40%)

In general, our strategy IISA has the best performance than MPF and MFF under different metrics. Besides, importing moderate posts from high-influence friends does not hurt and even benefits the state-of-the-art models.

6.6 Importing Posts for Users with Limited Posts

To evaluate the influence of introduced posts, we design three scenarios. In scenario 1, silent users import posts. In scenario 2, wordless users import posts. In scenario 3, both silent and wordless users import posts.

Table 6 shows the AvgF1 of MBTI models in different scenarios. In scenario 1, there are 1202 users in total, containing 95 silent users. The best AvgF1 for testing silent users reaches 54.64%, which is lower than the best AvgF1 of testing active users (61.31%). Nevertheless, this result is still remarkable since the personality of silent users can not be predicted in existing works. In scenario 2, there are 1413 users in total, containing 358 wordless users. The best AvgF1 for testing wordless users is 66.65%, which is better than the value of testing active users (62.29%). Because of IISA, wordless users are able to replenish posts from their mentioned following. In scenario 3, there are 1202 users in total, containing 391 wordless users and 61 silent users. Compared to the result of best AvgF1 values, it shows similar conclusions to the first and second scenarios.

Table 6. AvgF1 of MBTI models in different scenarios

Model	Scenario 1		Scenario 2		Scenario 3		
	Active users	Silent users	Active users	Wordless users	Active users	Silent users	Wordless users
Bert-SVM	55.61%	51.76%	56.98%	61.88%	57.33%	53.73%	57.23%
Bert-RF	59.37%	53.00%	61.14%	65.72%	60.68%	55.61%	60.45%
Bert-XGBoost	61.31%	54.64%	62.29%	66.65%	61.86%	63.07%	65.91%
Trignet	59.94%	53.82%	59.40%	59.14%	61.30%	61.24%	54.36%

Table 7 shows the Average F1 of Big Five models in different scenarios. In scenario 1, there are 717 users in total, containing 49 silent users. The best AvgF1 for testing silent users reaches 53.89%. In scenario 2, there are 837 users in total, containing 205 wordless users. Note that the best AvgF1 for testing wordless users is 79.62%, which is better than the value of testing active users (75.18%). In scenario 3, there are 862 users in total, containing 182 wordless users and 35

Table 7. Avg F1 of big five models in different scenarios

Model	Scenario 1		Scenario 2		Scenario 3		
	Active users	Silent users	Active users	Wordless users	Active users	Silent users	Wordless users
Bert-SVM	75.73%	53.89%	75.18%	79.62%	74.52%	59.10%	78.77%
Bert-RF	73.48%	53.51%	74.31%	76.34%	74.35%	53.09%	67.48%
Bert-XGBoost	72.64%	51.53%	73.15%	75.18%	73.03%	54.14%	66.62%
Trignet	67.90%	49.36%	70.72%	62.87%	71.78%	67.20%	49.86%

silent users. The best AvgF1 for testing silent users reaches 59.10%. Besides, the best AvgF1 for testing wordless users is over 4% than active users.

In general, GPAM shows better performance than Trignet in different scenarios. The imported posts from high-influence friends bring great gains for silent users.

7 Conclusion

This paper proposes GPAM, a general personality analysis model based on posts and links in social networks. GPAM proposes a user linkage technique to collect large-scale and high-quality labeled personality data shortly, and a unified feature extraction model to tackle the problem of inaccurate representation of users without enough posts. The experimental results demonstrate that importing moderate posts from high-influence friends greatly benefits silent and wordless users, and brings better performance than state-of-the-art model Trignet.

In the future, we plan to design various strategies for selecting posts from high-influence friends and extract personality features based on both LIWC and pretrain models. Besides, we plan to further extend our approach to predicting other personality models like Enneagram, Temperaments, Socionics, etc.

References

1. Corr, P.J., Matthews, G.: The Cambridge Handbook of Personality Psychology. Cambridge University Press, Cambridge (2020)
2. Employment testing (2022). <https://www.siop.org/Business-Resources/Employment-Testing/Test-Use>
3. Personality testing (2022). <https://www.psychologytoday.com/intl/blog/credit-and-blame-work/200806/the-use-and-misuse-personality-tests-coaching-and-development>
4. Dhelim, S., Aung, N., Bouras, M.A., Ning, H., Cambria, E.: A survey on personality-aware recommendation systems. arXiv preprint [arXiv:2101.12153](https://arxiv.org/abs/2101.12153) (2021)
5. Schedl, M.: Deep learning in music recommendation systems. *Front. Appl. Math. Stat.* **5**, 44 (2019)
6. Annalyn, N., Bos, M.W., Sigal, L., Li, B.: Predicting personality from book preferences with user-generated content labels. *IEEE Trans. Affect. Comput.* **11**(3), 482–492 (2018)

7. McCrae, R.R., Costa, P.T.: Clinical assessment can benefit from recent advances in personality psychology. *Am. Psychol.* **41**(9), 1001 (1986)
8. Myers, I.B., McCaulley, M.H., Quenk, N.L., Hammer, A.L.: *MBTI Manual: a Guide to the Development and Use of the Myers-Briggs Type Indicator*. Consulting Psychologists Press, Mountain View (1998)
9. Kaggle mbti (2022). <https://www.kaggle.com/datasnaek/mbti-type>
10. Myperson dataset (2022). <https://github.com/jcl132/personality-prediction-from-text>
11. Plank, B., Hovy, D.: Personality traits on twitter-or-how to get 1,500 personality tests in a week. In: *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 92–98 (2015)
12. Twitter personality dataset (2022). <https://github.com/monicamanda/twitter-personality-classification>
13. Pennebaker, J.W., King, L.A.: Linguistic styles: language use as an individual difference. *J. Pers. Soc. Psychol.* **77**(6), 1296 (1999)
14. Tausczik, Y.R., Pennebaker, J.W.: The psychological meaning of words: LIWC and computerized text analysis methods. *J. Lang. Soc. Psychol.* **29**(1), 24–54 (2010)
15. Yang, H.C., Huang, Z.R.: Mining personality traits from social messages for game recommender systems. *Knowl.-Based Syst.* **165**, 157–168 (2019)
16. Gjurković, M., Vukojević, I., Šnajder, J.: SIMPA: statement-to-item matching personality assessment from text. *Future Gener. Comput. Syst.* **130**, 114–127 (2021)
17. Wei, H., et al.: Beyond the words: predicting user personality from heterogeneous information. In: *Proceedings of the tenth ACM International Conference on Web Search and Data Mining*, pp. 305–314 (2017)
18. Sun, X., Liu, B., Cao, J., Luo, J., Shen, X.: Who am i? personality detection based on deep learning for texts. In: *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE (2018)
19. Yang, F., Quan, X., Yang, Y., Yu, J.: Multi-document transformer for personality detection. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 14221–14229 (2021)
20. Yang, T., Yang, F., Ouyang, H., Quan, X.: Psycholinguistic tripartite graph network for personality detection. 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. 4229–4239 (2021)
21. Xue, X., Feng, J., Sun, X.: Semantic-enhanced sequential modeling for personality trait recognition from texts. *Appl. Intell.* **51**(11), 7705–7717 (2021). <https://doi.org/10.1007/s10489-021-02277-7>
22. Golbeck, J., Robles, C., Turner, K.: Predicting personality with social media. In: *CHI 2011 Extended Abstracts on Human Factors in Computing Systems*, pp. 253–262 (2011)
23. Gu, H., Wang, J., Wang, Z., Zhuang, B., Su, F.: Modeling of user portrait through social media. In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE (2018)
24. Wald, R., Khoshgoftaar, T., Sumner, C.: Machine prediction of personality from Facebook profiles. In: *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, pp. 109–115. IEEE (2012)
25. Chittaranjan, G., Blom, J., Gatica-Perez, D.: Mining large-scale smartphone data for personality studies. *Pers. Ubiquit. Comput.* **17**(3), 433–450 (2013)
26. Lima, A.C.E., de Castro, L.N.: Tecla: a temperament and psychological type prediction framework from twitter data. *PLoS ONE* **14**(3), e0212844 (2019)

27. Personalitycafe. <https://www.personalitycafe.com/>
28. Personality database (2022). <https://www.personality-database.com/>
29. 44% of twitter accounts have never sent a tweet (2022). <https://www.wsj.com/articles/BL-DGB-34255>
30. Socionics relations. <http://www.socionics.com/rel/rel.htm>
31. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)