# Large Language Model as Personality Classifier

Jakub Halmeš

May 2024

## 1 Introduction

In recent years, large language models (LLMs) have become increasingly proficient at a wide range of natural language understanding tasks, such as for example text sumarization, generation, or code completion. In this work, I try to use them for personality classification, where the model has to predict a personality class of a person based on the person's text.

Using LLM for classification has been done before – even the original GPT paper [1] included fine-tuning for classification. Another study [2] investigates the performance of LLMs in semantic classification, noting that although large models outperform smaller ones, they still fall short in tasks requiring deeper understanding. Similar tasks are explored in several online notebooks, such as in [3].

The motivation for this study is two-fold: firstly, to apply the model to a classification task with more than three classes—here, 16 personality types. Secondly, to gain hands-on experience in fine-tuning a large language model.

## 2 Personality classification

Personality classification involves predicting the personality traits of individuals based on their written text. One popular framework for personality classification is the Myers-Briggs Type Indicator (MBTI), which categorizes individuals into one of 16 personality types based on four binary categories: Introversion (I) vs. Extraversion (E), Sensing (S) vs. Intuition (N), Thinking (T) vs. Feeling (F), and Judging (J) vs. Perceiving (P). For example, an individual might be classified as an "INTJ" if they exhibit Introversion, Intuition, Thinking, and Judging preferences.

## 3 Data

I used a dataset compiled from PersonalityCafe forum, which is available on Kaggle [4]. The dataset contains person's last 50 messages on the forum together with their self-eported MBTI types. In total, the dataset contains 8675 entries, which I split to train, validation, and test split with proportions 64%, 16%, and 20%, respectively. The data was of good quality and I did only minimal pre-processing.

### 3.1 Pre-processing

In all experiments I modified the links in the messages, and in some of the latter experiments I replaced the personality classes present in the messages with a placeholder.

#### 3.1.1 Removing links

The forum messages frequently included links to external websites; however, these links generally had little relevance to the classification task. Additionally, messages containing these links tended to be excessively lengthy. Nonetheless, understanding the types and frequency of links a person uses could be relevant for classification purposes.

To shorten the links while still keeping some relevant information, I replaced them with a placeholder containing the website name, but stripped of other details. So, for example, *http://wallpaperpassion.com/upload/23700/friendship-boy-and-girl-wallpaper.jpg* was replaced with *[WALLPAPERPASSION_COM_LINK]*. This was done in all experiments.

#### 3.1.2 Removing personality classes

Apart from links, the messages often contained some mention of personality types themselves. This comes from the nature of the forum, which is centered around personality types. Here's an example of such message:

> *"Is there one? For example, I am an INFP, and I grew up in a houshold where my mom was an INFP and my dad was an ENFP. My brother turned out an ESFP. Any correlations with you guys?"*

This is problematic, as the dataset may contain the correct answers. I fix this issue by replacing these occurences with a *[redacted]* string. Unfortunately, I only realized this after training the first model, so unless stated otherwise, the experiments were done on data without this modification.

## 4 Fine-tuning

I fine-tuned the 4-bit quantized Gemma7B model for one epoch using data that included several messages combined with a task-specific prompt and the correct personality label. The training was performed on a P100 GPU available on Kaggle, taking approximately 8-10 hours for the epoch.

Due to memory constraints, I divided each original data point, which contained 50 messages, into four new data points. These consisted of 12, 12, 12, and 14 messages each.

## 5 Evaluation

The evaluation consists of several parts: Traditional approaches, base models and fine-tuned models.

### 5.1 Traditional approaches

Traditional approaches like Support Vector Machines (SVM), Naive Bayes, and Random Forests with TfIDF vectorization serve as baseline models in our evaluation. These methods are widely used in text classification tasks due to their simplicity, speed, and often surprisingly good performance.

| **Model** | **Accuracy (%)** |
|---|---|
| SVM | 63.52 |
| Random Forest | 48.53 |
| Naive Bayes | 21.90 |

Table 1: Accuracy of Traditional Approaches

I also evaluated these approaches on data with replaced mentions of personality classes, as described in 3.1.2.

| **Model** | **Accuracy (%)** |
|---|---|
| SVM | 40.98 |
| Random Forest | 27.72 |
| Naive Bayes | 21.50 |

Table 2: Accuracy of Traditional Approaches with *[redacted]* personality classes.

### 5.2 Prompt

One of the most important part of this project was to come up with a good prompt to use for the LM, which would help elicit the desired behavior. To find the best prompt, I experimented with Gemma before any training, using both manual inspection of the answers and calculating the accuracy on the evaluation set. The final prompts used are shown in Appendix A.

### 5.3 Base models

First, the language models were evaluated without any additional fine-tuning.

The results on 2B models indicate that the quantization does not hurt performance too much, and it's reasonable to use the quantized model. Due to compute limitations, evaluating 7B version without quantization was not possible.

| **Model** | **Accuracy (%)** |
|---|---|
| Gemma-2B | 15.45 |
| Gemma-2B q. | 14.47 |
| DaVinci | 28.93 |
| Gemma-7B q. | 44.90 |

Table 3: Accuracy without fine-tuning

The DaVinci model was evaluated through the OpenAI API [1].

## 5.4 Fine-tuned models

In this section, I present the results of the fine-tuned models. This includes the initial fine-tuning and two additional training configurations.

The accuracy of the initially fine-tuned model was **82.07%**, roughly doubling the performance of the model without any fine-tuning. This impressive performance led me to uncover the issue with the training data described in Section 3.1.2. Evaluating the same model on the modified data resulted in accuracy **41.10%**, which was about the same as with SVM. After re-training the model on the modified data, the accuracy was increased to **50.95%**. This shows that the huge improvement of the first fine-tuned model came from focusing on occurences of the personality classes in the text.

Finally, I modified the label format to provide the model with more flexibility in predicting personality classes, rather than constraining it to predict the exact four-letter MBTI types. This change was motivated by the potential difficulty of exact letter prediction, especially considering tokenization issues. Specifically, I replaced the labels with comma-separated lists of the full descriptive words: for example, *ISTJ* was converted to *Introversion, Sensing, Thinking, Judging.* For each personality axis, the prediction is determined by which label stem appears first in the model's response (e.g., ”introv” or ”extrov” for the Introversion-Extroversion axis). The original personality classes in the output were replaced with a *[redacted]* label to maintain confidentiality. Details of the prompt used are illustrated in Figure 2.

The model achieved an accuracy of **48.93%**. This suggests that the expanded and less constrained option space did not hinder its performance.

[1] Which cost roughly \$6.50.

| **Model Configuration** | **Accuracy (%)** |
|---|---|
| Original fine-tuning | 82.07 |
| Evaluated with *[redacted]* | 41.10 |
| Retrained with *[redacted]* | 50.95 |
| Labels replaced with words + *[redacted]* | 48.93 |

Table 4: Accuracy of the fine-tuned model in different configurations.

## 6 Conclusion

In this work I explored the use of (LLMs) for personality classification based on the Myers-Briggs Type Indicator (MBTI). Using a dataset from the PersonalityCafe forum, I evaluated both traditional machine learning methods and fine-tuned LLMs. Initial high accuracy with trained LLMs was due to unintended leakage of personality class information in the training data. After addressing this issue, the fine-tuned models still outperformed traditional approaches, though with a reduced margin.

## A Prompts

I used two prompts: one for predicting the exact personality class label, shown in Figure 1, and the other one with more relaxed prediction of comma-separated words describing the classes, shown in Figure 2.

In both cases, the {*text*} placeholder is replaced with the actual forum messages. During evaluation, {*personality_class*} is always replaced with empty string, and the model has to continue in this prompt. During training, the input is the part up to *### Response*, and the actual personality class is used as the desired continuation of the model.

The templates differ in the text that replaces {*personality_class*} placeholder: in the first case, the exact label is used, such as ISTJ, while in the latter one it's replaced with Introversion, Sensing, Thinking, Judging.

```
### Input
Person's forum posts separated by |||:
{text}
### Personality classification
(options: ISTJ, ISFJ, INFJ, INTJ, ISTP, ISFP, INFP,
INTP, ESTP, ESFP, ENFP, ENTP, ESTJ, ESFJ, ENFJ, ENTJ).
First letter: Introversion (I) - Extroversion (E)
Second letter: Intuition (N) - Sensing (S)
Third letter: Thinking (T) - Feeling (F)
Fourth letter: Judging (J) - Perceiving (P)

From these options, the person can be best classified as
### Response
{personality_class}
```

Figure 1: Prompt template with exact classification.

```
### Input
Person's forum posts separated by |||: {text}
### Personality classification.
First axis: Introvert - Extrovert
Second axis: Intuition - Sensing
Third axis: Thinking - Feeling
Fourth axis: Judging - Perceiving

On each of these axis (separated by commas)
the person can be best classified as
### Response
{personality_class}
```

Figure 2: Prompt template with relaxed classification.

## B Installation instructions

To run the code, first install the required packages using the provided 'conda.yaml' file. You can do this by executing the following command in your terminal:

```
conda env create -f conda.yaml
```

## References

[1] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[2] W. Zhang, Y. Deng, B. Liu, S. J. Pan, and L. Bing, "Sentiment analysis in the era of large language models: A reality check. arxiv," *arXiv preprint arXiv:2305.15005*, 2023.

[3] Lucamassaron, "Fine-tune gemma 7b it for sentiment analysis (tpu)," Feb 2024. [Online]. Available: https://www.kaggle.com/code/lucamassaron/fine-tune-gemma-7b-it-for-sentiment-analysis-tpu

[4] M. J, "(mbti) myers-briggs personality type dataset," Sep 2017. [Online]. Available: https://www.kaggle.com/datasets/datasnaek/mbti-type