CAS2105 Homework 6: Mini AI Pipeline Project 🤗
MBTI Post Classification: Feeling (F) vs Thinking (T)

JunyoungLee (2020146059)

1 Introduction

This project builds a small AI pipeline to classify MBTI forum posts into **Feeling (F)** or **Thinking (T)**. The goal is not to build a perfect model, but to practice a typical workflow: data preprocessing, designing a simple baseline, building an improved AI method, and comparing them with quantitative metrics.

A Kaggle MBTI dataset is used, where each row contains an MBTI type and many posts in a single text field [1]. A keyword-based baseline is implemented first. Then a transformer-based model is trained using `microsoft/deberta-v3-base` as an encoder and an MLP classifier head [2].

2 Task Definition

- **Task description:** Given a forum post snippet, predict whether it is **F** or **T**.
- **Motivation:** This task is simple but realistic because the text is noisy and the label may not be perfect.
- **Input / Output:** Input is a short English text snippet. Output is one label in {F, T}.
- **Success criteria:** Higher Accuracy and Macro-F1 on a test split, and reasonable predictions on examples.

3 Methods

3.1 Naïve Baseline

The baseline is a keyword-count rule. Two word sets are defined:

- **FEEL_WORDS:** feel, feeling, emotion, love, like, heart, emotional, care, compassion, empathy, relationship
- **THINK_WORDS:** logic, reason, analyze, analysis, objective, data, rational, facts, structure, system, argument

For each input text, the text is lowercased and the number of occurrences from each set is counted (substring match). If the Feeling count is larger, the prediction is **F**. Otherwise (including ties), the prediction is **T**.

Why it is naïve / failure cases

- The method does not capture semantics; it only checks a small list of words.
- If a post uses different words or expresses emotion/logic indirectly, the method fails.

- Many posts do not contain these keywords, so predictions often default to T.

3.2 AI Pipeline

Model

`microsoft/deberta-v3-base` is used as the encoder, with a small MLP classifier:

```python
from __future__ import annotations

import torch
from torch import nn

class MLPClassifier(nn.Module):
    def __init__(self, input_size: int, hidden_size: int) -> None:
        super().__init__()
        self.net = nn.Sequential(
            nn.Linear(input_size, hidden_size),
            nn.ReLU(),
            nn.Dropout(0.1),
            nn.Linear(hidden_size, 2),
        )

    def forward(self, features: torch.Tensor) -> torch.Tensor:
        return self.net(features)
```

The encoder and classifier are fine-tuned together (full fine-tuning).

Pipeline steps

1. Preprocess raw posts (remove URLs, normalize spaces).
2. Split the `posts` field by `|||` into snippets.
3. Tokenize text with a Hugging Face tokenizer.
4. Obtain representations from the DeBERTa encoder.
5. Predict F/T with the MLP head.

4 Experiments

4.1 Datasets

Source. Kaggle MBTI dataset [1].

Label rule. The dataset contains MBTI types such as `INFJ`. The **3rd letter** is used as the label:

$$\mathrm{MBTI}[2] = F \Rightarrow F, \quad \mathrm{MBTI}[2] = T \Rightarrow T$$

Each snippet from the same user is assigned the same F/T label. This can be noisy because a single snippet may not clearly represent the F/T trait.

Preprocessing.

- Split by `|||`
- Remove URLs
- Normalize whitespace
- Remove very short snippets

After splitting, the number of rows becomes much larger. In the baseline run, the following split size is observed:

$$n = 383{,}347 \quad (\text{train } 306{,}677, \ \text{test } 76{,}670)$$

4.2 Metrics

The following metrics are reported:

- **Accuracy**
- **Macro-F1** (average of F1 for each class)

4.3 Results

Main dataset results

Table 1: Baseline vs AI pipeline results.

Method	**Split size**	**Accuracy**	**Macro-F1**
Baseline (keyword rule)	Test $n = 76{,}670$	0.5181	0.5079
AI Pipeline (DeBERTa + MLP, fine-tuned)	Test $n = 38{,}335$	0.6500	0.6445

The baseline performs close to random, while the AI pipeline performs substantially better. One important note is that the test sizes are different across logs. For a strictly fair comparison, both methods should be evaluated on the exact same split. This report presents the currently logged results.

Training log (AI pipeline)

Validation results during training:

- Epoch 1: val_acc=0.6513, val_macro_f1=0.6509
- Epoch 2: val_acc=0.6695, val_macro_f1=0.6679
- Epoch 3: val_acc=0.6715, val_macro_f1=0.6643
- Epoch 4: val_acc=0.6610, val_macro_f1=0.6560
- Epoch 5: val_acc=0.6549, val_macro_f1=0.6493

Performance peaks around epoch 2–3 and decreases afterwards, which may indicate mild overfitting or label noise.

Synthetic inference set results

On a small synthetic set ($n = 100$):

- Baseline inference: acc=0.7200, macro_f1=0.7190
- AI inference: acc=0.6600, macro_f1=0.6511

This result is likely due to limitations of the synthetic evaluation set. First, the number of synthetic samples is very small, which makes the measured performance highly sensitive to data bias. Second, many synthetic sentences contain overly explicit cue words that are directly used by the baseline, which can unfairly favor the keyword-based method. Therefore, the higher baseline score on the synthetic set does not necessarily indicate that it generalizes better than the AI pipeline.

Example cases

Example predictions (AI model):

```
{"text":"I like clear rules when dealing with learning new skills.","pred":"F","gold":"T"}
{"text":"Objective evidence about health goals matters most to me.","pred":"F","gold":"T"}
{"text":"Reasoning through career choices is more important than feelings.","pred":"F","gold":"T"}
```

These cases suggest that the model can still make incorrect predictions even when a sentence appears to exhibit a clearly Thinking (T)-like style. Possible reasons include *label noise* (the gold label is derived from the author's overall MBTI type rather than the individual sentence) and the possibility that the model has learned topic- or writing-style patterns instead of the underlying F/T trait.

5 Reflection and Limitations

The keyword-based baseline was easy to implement; however, its performance was close to random guessing approximately 0.5 accuracy, which is essentially comparable to a coin flip. This indicates that a small, fixed keyword list is insufficient for solving the target task. In contrast, the DeBERTa-based model improved performance to approximately 0.65 accuracy and 0.64 macro-F1, suggesting that learning contextual representations is beneficial for this classification problem.

Nevertheless, the labeling strategy introduces noise. Even when an individual snippet does not clearly reflect Feeling (F) or Thinking (T) characteristics, the label is assigned uniformly based on the author's overall MBTI type. In addition, splitting long posts into multiple snippets produces many samples from the same user. With a snippet-level split, highly similar writing styles may appear in both the training and test sets, potentially inflating the evaluation results. With more time, a user-level split would be applied so that the same user never appears in both training and test sets. For further improvement, early stopping around 2–3 epochs and stronger regularization could be considered. Finally, for a fair comparison, the baseline and the AI pipeline should be evaluated on the exact same test split.

The dataset is collected from online MBTI forums, and therefore the distribution of MBTI types is not balanced. For example, Introversion (I) and Intuition (N) appear more frequently than other traits. This skewed distribution could correlate with writing style and content, but this relationship was not examined in the current project.

In addition, transformer-based models have an input length limit (typically 512 tokens). As a result, truncation is required for long posts, which may lead to information loss and potentially degrade performance. Due to these dataset-specific characteristics and distributional biases, the generalizability of the current results remains uncertain and requires further validation.

Finally, the classifier head used in this project is a shallow MLP with a single hidden layer. This limited capacity may restrict the model's ability to infer more complex decision boundaries from the encoder representations. If the classifier component is further enhanced, additional performance gains may be achievable.

References

[1] Mbti (myers-briggs) personality type dataset. Kaggle. URL https://www.kaggle.com/datasets/datasnaek/mbti-type. Accessed: 2025-12-14.

[2] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2006.03654.