CAS2105 Homework 6: Mini AI Pipeline Project 🤗

Giwan Eom (2024148058)

1 Introduction

This project explores a simple text classification task using a small AI pipeline. Rather than training a large model from scratch, the goal is to compare a naïve rule-based baseline with an improved pipeline built using a pre-trained language model. Sentiment classification provides a clear and interpretable setting to study the strengths and limitations of heuristic approaches versus modern transformer-based models.

The project emphasizes the overall AI workflow, including task definition, pipeline design, quantitative and qualitative evaluation, and reflection. All experiments are designed to be lightweight and runnable on a standard CPU environment.

2 Task Definition

- **Task description:** The task is binary sentiment classification: given an English sentence, classify it as either positive or negative.
- **Motivation:** Sentiment classification is a fundamental natural language processing task with applications in review analysis, opinion mining, and social media monitoring.
- **Input / Output:** The input is a single English sentence. The output is a binary sentiment label (`positive` or `negative`).
- **Success criteria:** A system is considered successful if it achieves higher accuracy and F1 score on a held-out test set.

3 Methods

This section includes both the naïve baseline and the improved AI pipeline.

3.1 Naïve Baseline

- **Method description:** The baseline uses a keyword-based heuristic. A small set of positive and negative sentiment words is manually defined. Each sentence is tokenized by whitespace, and a sentiment score is computed by adding +1 for each positive word and -1 for each negative word. If the final score is positive, the sentence is classified as positive; otherwise, it is classified as negative.
- **Why naïve:** This method relies on a fixed lexicon and does not use any learned representations. It ignores word order, context, and interactions between words.
- **Likely failure modes:** The baseline fails on negation (e.g., "not good"), sarcasm, sentences without sentiment keywords, and domain-specific sentiment expressions.

3.2 AI Pipeline

- **Models used:** The pipeline uses the pre-trained transformer model `distilbert-base-uncased-finetuned-sst-2-english`.
- **Pipeline stages:** (1) The input sentence is tokenized and truncated using the model tokenizer. (2) DistilBERT encodes the sentence into contextualized embeddings. (3) A classification head predicts a sentiment label. (4) The predicted label is mapped to a binary class.
- **Design choices and justification:** DistilBERT is a lightweight transformer that captures contextual meaning while remaining computationally efficient. Using an inference-only pretrained model satisfies the project requirements and enables a clear comparison with the naïve baseline without additional training complexity.

4 Experiments

4.1 Datasets

- **Source:** The SST-2 dataset from the GLUE benchmark.
- **Total examples:** 400 sentences were used in total.
- **Train/Test split:** 200 examples were used for training and 200 examples for testing (sampled from the validation split).
- **Preprocessing steps:** Tokenization and lowercasing are handled internally by the tokenizer. Sentences longer than the maximum length are truncated.

4.2 Metrics

Two standard classification metrics were used to evaluate performance.

- **Accuracy:** the proportion of correctly classified examples among all test examples. This metric provides an intuitive measure of overall correctness.
- **F1 score:** the harmonic mean of precision and recall. F1 is particularly useful for evaluating binary classification tasks where class balance and error types are important.

These metrics together capture both overall performance and the balance between false positives and false negatives.

4.3 Results

Method	Accuracy	F1 Score
Baseline (keyword rules)	0.51	0.22
AI Pipeline (DistilBERT)	0.915	0.923

Qualitative examples. The following examples illustrate cases where the naïve baseline and the AI pipeline produce different predictions.

- *"It gets onto the screen just about as much of the novella as one could reasonably expect, and is engrossing and moving in its own right."*
  The baseline incorrectly predicts a negative label due to the absence of explicit sentiment keywords. In contrast, the AI pipeline correctly identifies the overall positive sentiment by capturing the nuanced and descriptive language.
- *"My Big Fat Greek Wedding uses stereotypes in a delightful blend of sweet romance and lovingly dished out humor."*

The baseline fails because it relies on a limited keyword list, while the AI pipeline correctly recognizes positive sentiment expressed through phrases such as “delightful” and “sweet romance.”

- *“For the most part, director Anne-Sophie Birot’s first feature is a sensitive, extraordinarily well-acted drama.”*
  The baseline predicts a negative label due to missing sentiment cues, whereas the AI pipeline successfully captures the positive evaluative tone of the sentence.

5 Reflection and Limitations

The pre-trained model performed significantly better than expected given the small dataset size and the absence of fine-tuning. Designing a naïve baseline highlighted how limited rule-based approaches are for understanding natural language. The most challenging aspect was using github because I’m not familiar with github. Accuracy and F1 score effectively captured the large performance gap, although they do not fully reflect linguistic nuance. The baseline’s low F1 score demonstrates its inability to handle contextual meaning. With more time, fine-tuning the model or experimenting with different pre-trained architectures could further improve performance. Future work could also explore sentiment datasets with neutral or mixed labels.