CAS2105 Homework 6: Mini AI Pipeline Project 🤗

Hyewon Nam (2024149002)

1 Introduction

This project provides a gentle introduction to designing simple **AI pipelines**. Rather than training large models or reading extensive research literature, you will:

- Choose a small, concrete problem solvable with an AI pipeline (e.g., text classification, retrieval, simple QA, image classification).
- Choose or collect a small dataset (e.g., from `datasets`).
- Implement a simple *naïve baseline* (e.g., rule-based or heuristic).
- Build an improved pipeline using existing pre-trained models.
- Evaluate both approaches using appropriate metrics.
- Reflect on what worked, what failed, and what you learned.

The emphasis is on the *process* of AI work: problem definition, pipeline design, evaluation, iteration, and writing. Your problem should be small enough to run comfortably on a single GPU (e.g., RTX 3090) or CPU.

2 Task Definition

- **Task description:** The goal of this project is to classify news headlines into one of four categories: World, Sports, Business, and Sci/Tech. The system receives a short headline as input and predicts the most relevant category.[1]
- **Motivation:** News classification plays a significant role in real-world applications such as personalized news recommendations, online content moderation, and information retrieval. Automating this task allows users to efficiently navigate large volumes of news articles and find information relevant to their interests.
- **Input / Output:** The input to the system is a textual news headline. The output is a single predicted category label among World, Sports, Business, Sci/Tech.
- **Success criteria:** The system is considered successful if the AI pipeline achieves higher classification performance than the naive baseline when evaluated on the same test dataset. In particular, improvements in Accuracy and F1-score indicate that the pipeline is "good."

3 Methods

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

3.1 Naïve Baseline

- **Method description:** The naive baseline is a rule-based keyword classifier. It predicts the category of a headline by checking whether the text contains specific keywords associated with each class. For example, headlines containing "game", "team", or "match" are classified as Sports; headlines containing "market", "company", or "stock" are classified as Business; and headlines containing "technology", "research", or "software" are classified as Sci/Tech. Headlines that do not match any keyword pattern are assigned to the World category.
- **Why naïve:** The baseline relies solely on keyword presence and does not understand sentence structure, context, or semantic relationships between words. It assumes that a small set of explicit keywords fully represents each category, which oversimplifies natural language.
- **Likely failure modes:** The baseline is expected to fail when headlines imply topics indirectly, use figurative or ambiguous language, or contain domain-overlapping terms. For example, headlines referring to technology companies might be incorrectly classified as Business rather than Sci/Tech, and headlines about sports business deals may be misclassified due to keyword ambiguity.

3.2 AI Pipeline

- **Models used:** I used the DistilBERT-base-uncased transformer model from Hugging Face. It is a lightweight variant of BERT that preserves most of the performance while significantly reducing model size and computational cost. A pre-trained version of DistilBERT was fine-tuned on the AG News dataset for 4-class text classification (World, Sports, Business, Sci/Tech).[2]
- **Pipeline stages:**
- **Pipeline stages (mapped to the generic AI pipeline specification):**
  - **(1) Preprocessing**
    The AG News dataset is loaded using `load_dataset("ag_news")`, containing four news categories (World, Sports, Business, Sci/Tech). The raw news text is then converted into token IDs using the DistilBERT tokenizer with max–length padding and truncation (128 tokens), producing `input_ids`, `attention_mask`, and `labels` tensors for model input.
  - **(2) Embedding / Representation**
    A pretrained `distilbert-base-uncased` transformer is used to produce contextualized token embeddings from the tokenized input. Through stacked transformer layers, the `[CLS]` token representation captures the semantic content of the full news article and serves as the document-level embedding for downstream classification.
  - **(3) Decision**
    A classification head (feed-forward layer + softmax) on top of the `[CLS]` embedding predicts the probability distribution over the four news categories. The model is fine-tuned on the AG News dataset using the Hugging Face `Trainer` API with AdamW optimization, weight decay regularization, and mini-batch updates. I started from the pretrained `distilbert-base-uncased` model and fine-tuned it for 4-way classification using the Trainer API.
  - **(4) Optional post-processing (not used in this project)**
    No additional heuristic rules or calibration methods were applied after model prediction; the highest-probability category (argmax of the softmax output) was taken as the final label.
- **Design choices and justification:**

- (1) Use DistilBERT rather than full BERT: DistilBERT was selected because it preserves most of BERT's accuracy while significantly reducing GPU memory footprint and computational cost. This enables faster experimentation and training without sacrificing much performance.
- (2) Max-length padding and truncation (128 tokens): AG News articles are generally short-to-medium length. A sequence length of 128 balances information preservation and training speed. Longer sequences bring minimal benefit but increase computation cost.
- (3) Learning rate of 2e-5 and weight decay 0.01: These are standard, empirically validated hyperparameters for fine-tuning BERT-like models. They promote stable training and help avoid overfitting on relatively small datasets.
- (4) Batch size = 16 and single-epoch training: A batch size of 16 fits well into limited GPU memory. Training for one epoch is sufficient to demonstrate full pipeline functionality and baseline transformer performance without excessive runtime.

4 Experiments

All experiments are fully reproducible using the provided scripts, which fix random seed (42) across dataset loading, batching, and fine-tuning configuration.

4.1 Datasets

- **Source:** The dataset used in this project is the AG News corpus, a public news topic classification dataset available through the Hugging Face Datasets library.
- **Total examples:** The dataset contains 127,600 news articles in total, consisting of 120,000 training examples and 7,600 test examples.
- **Train/Test split:** I used the original predefined split provided by the AG News dataset without further re-partitioning. The training split was used for fine-tuning the model, and the test split was used exclusively for final evaluation. Both the baseline and the AI pipeline are evaluated on the same AG News test split for fair comparison.
- **Preprocessing steps:**
  - For the baseline classifier: text lowercasing for keyword-based matching.
  - For the DistilBERT pipeline: tokenizer-based preprocessing including max-length padding and truncation (128 tokens), conversion into token IDs and attention masks, and formatting into PyTorch tensors for model input.

4.2 Metrics

- **Accuracy:** Measures the overall proportion of correctly classified samples out of the entire test set. It reflects how often the model makes the right prediction in general, which is suitable for evaluating news classification systems where the goal is to assign each article to exactly one category.
- **F1 score:** Represents the harmonic mean of precision and recall, capturing how consistently the model performs across all classes. It is especially informative in multi-class settings because it penalizes models that perform well only on a subset of classes. This aligns well with this task, since the four AG News categories are balanced but semantically distinct, making class-wise consistency as important as overall correctness.

4.3 Results

Text (excerpt)	True Label	Baseline	AI Pipeline
Fears for T N pension after talks Unions representing workers at Turner Newall say they are disappointed after talks with parent firm Federal Mogul.	Business	Sci/Tech	Business
The Race is On: Second Private Team Sets Launch Date for Human Spaceflight (SPACE.com) ... competing for the $10 million Ansari X Prize for privately funded suborbital space flight.	Sci/Tech	Sports	Sci/Tech
Ky. Company Wins Grant to Study Peptides ... a chemistry researcher won a grant to develop a method of producing better peptides, the building blocks of proteins.	Sci/Tech	Sports	Sci/Tech

Table 1: Qualitative comparison of baseline vs. AI pipeline predictions on the AG News dataset.

Method	Accuracy	F1
Baseline	0.4691	0.4725
AI Pipeline	0.9422	0.9423

Table 2: Accuracy and F1 comparison of baseline and AI pipeline on the AG News dataset.

The AI pipeline improved accuracy from 0.4691 to 0.9422, representing a +100.9% relative performance increase over the naive baseline. The F1 score showed a similar trend, rising from 0.4725 to 0.9423 (+99.4%), indicating that the pipeline consistently improved performance across all four classes rather than excelling only in one category. This confirms that performance gains come from contextual understanding rather than keyword memorization or class-specific bias.

5 Reflection and Limitations

The baseline keyword classifier turned out to be much more limited than expected, making incorrect predictions when articles contained misleading surface-level keywords. Its sequential `if` structure caused it to decide on a class as soon as the first keyword match was found, meaning it did not compare multiple categories or consider which label was most strongly supported by the entire sentence. As a result, even when an article contained both business-related and science-related terminology, the baseline sometimes assigned the wrong label simply because one keyword appeared earlier in its priority order. The DistilBERT model avoided this failure mode by leveraging contextual information rather than surface-level word matching, and this contributed to its large performance improvement. The AI pipeline performed significantly better than expected, especially considering that only one epoch of fine-tuning was used. The evaluation metrics reflected this difference well: accuracy showed the overall performance gap, while F1 provided insight into how consistently each model handled all four categories. However, despite its strong performance on the AG News dataset, the current pipeline may not generalize well to out-of-distribution news sources or domains with different writing styles. Figurative or highly context-dependent language (e.g., irony or sarcasm) could also challenge the model, suggesting that further improvements may be required for real-world robustness. With more time or compute, I would explore training for additional epochs, experimenting with larger models such as RoBERTa or domain-adaptive pre-training, and incorporating validation-based early stopping to reduce overfitting and further improve robustness. I would also be interested in extending the task to include more than four news categories, as a larger label space could reveal additional strengths and limitations of the model.

References

[1] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 2015.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.