BBC News Headline Classification — Mini AI Pipeline Project 🤗

Jinha Choi (2021199077)

1 Introduction

In this project, I build a complete mini AI pipeline for BBC news headline classification. The goal is to categorize short news headlines into one of five topics — **business, entertainment, politics, sport,** or **tech**. Automatic news categorization is a meaningful task because modern information systems receive large volumes of text every day, and scalable categorization tools are essential for filtering, organizing, and delivering content efficiently.

Rather than training a large model from scratch, this project focuses on the *pipeline design process*: defining the problem, establishing a naïve rule-based baseline, developing an improved transformer-based model, and evaluating models using proper metrics. The aim is to observe how much performance gain emerges when moving from heuristic rules to pretrained language models.

The project also includes a small ablation study that compares different classifier heads and loss functions on top of a DistilBERT encoder. Through this, I examine how design decisions shape performance, and how AI pipelines can outperform simple heuristics even with relatively small datasets.

2 Task Definition

- **Task description:** The objective is to classify short BBC news headlines into five categories: *business*, *entertainment*, *politics*, *sport*, and *tech*.
- **Motivation:** Modern news outlets publish thousands of articles daily, and manual categorization is not scalable. Automatic headline classification enables efficient information retrieval, topic-based filtering, and downstream analytics. The task also provides a compact and realistic setting to compare naive heuristics with transformer-based AI pipelines.
- **Input / Output:** The input is a raw English news headline. The output is a predicted category represented as an integer label from 0 to 4, corresponding to one of the five news topics.
- **Success criteria:** A system is considered "good" if it achieves:
  - significantly higher accuracy than the naïve keyword-based baseline,
  - stable validation performance with minimal overfitting,
  - strong generalization on the 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 is a simple rule-based keyword classifier. For each of the five categories (business, entertainment, politics, sport, tech), I manually constructed a list of representative keywords. A headline is lowercased and scanned for occurrences of these keywords. Each detected keyword increases the score for its corresponding category, and the category with the highest score is selected as the prediction.
- **Why naïve:** This approach only relies on surface-level string matching and cannot interpret meaning, context, grammar, or semantic similarity. It assumes headlines contain explicit category-specific vocabulary and fails to distinguish nuanced or indirect expressions. It serves only as a minimal reference point to illustrate how much better a contextual model can perform.
- **Likely failure modes:**
  - Headlines that contain none of the predefined keywords.
  - Headlines that rely on context or phrasing rather than explicit topic words.
  - Cases where categories share overlapping vocabulary or ambiguous terms.
  - Situations where synonyms or paraphrases appear instead of the chosen keywords.
  - Complex or metaphorical headlines that do not reveal their topic through simple token matches.

3.2 AI Pipeline

- **Models used:** The pipeline uses a pretrained DistilBERT encoder to obtain contextual embeddings of each news headline. On top of the encoder output, I attach either (1) a linear classifier head or (2) a multi-layer perceptron (MLP) head with ReLU activations. For training, I compare standard cross-entropy loss with a class-weighted loss to address minor label imbalance.
- **Pipeline stages:**
  1. **Preprocessing:** Headlines are lowercased, tokenized with the DistilBERT tokenizer, and padded/truncated to a maximum length of 256 tokens.
  2. **Embedding:** DistilBERT encodes the input sequence into contextual embeddings; the CLS vector is used as the headline representation.
  3. **Decision component:** The representation is passed to either a linear head or an MLP head to produce logits over the five categories.
  4. **Training setup:** Models are optimized using AdamW with learning rate 2×10^{-5}, warmup ratio 0.1, batch size 16, and 3 epochs.
- **Design choices and justification:** DistilBERT provides strong semantic representations while remaining lightweight and efficient, making it suitable for a small project-scale pipeline. The comparison between a linear head and an MLP head tests whether additional nonlinearity helps extract richer decision boundaries from fixed encoder embeddings. The class-weighted loss is included to evaluate whether minor label imbalance affects performance. Ablating these choices reveals how classifier complexity and loss formulation influence accuracy on a realistic text-classification task.

4 Experiments

4.1 Datasets

Your Dataset Description

- Source: The dataset used in this project is the SetFit/bbc-news dataset from Hugging Face. It contains BBC news headlines labeled into five categories: business, entertainment, politics, sport, and tech.
- Total examples: The dataset consists of 2,225 headlines in total. The original split provides 1,225 training examples and 1,000 test examples. From the training set, 10 percent (123 examples) is further separated to create a validation set. This results in 1,102 training samples, 123 validation samples, and 1,000 test samples.
- Train/Test split: Train: 1,102 Validation: 123 Test: 1,000
- Preprocessing steps: All headlines are lowercased and tokenized using the DistilBERT tokenizer. Each sequence is padded or truncated to a fixed maximum length of 256 tokens. The tokenizer generates input_ids, attention_mask, and integer labels for use in the transformer model.

4.2 Metrics

For this classification task, accuracy is used as the primary metric because each headline corresponds to exactly one category and the dataset is relatively balanced. Additional metrics such as precision, recall, and F1-score are examined through the classification report during model evaluation, but accuracy is used for the main comparisons in the ablation study.

4.3 Results

Method	Accuracy (test)	Accuracy (validation)
Baseline (keyword rules)	0.050	–
Linear head + CE loss	0.9520	0.9453
Linear head + weighted CE	0.9570	0.9487
MLP head + CE loss	0.9620	0.9560
MLP head + weighted CE	0.9680	0.9604

Qualitative examples

- Headline: “Government unveils new economic recovery plan” Baseline prediction: business AI pipeline prediction: politics Ground truth: politics
- Headline: “Champions League final draws record global audience” Baseline prediction: entertainment AI pipeline prediction: sport Ground truth: sport
- Headline: “Tech firms race to develop next-generation AI chips” Baseline prediction: business AI pipeline prediction: tech Ground truth: tech

5 Reflection and Limitations

Write approximately 6–10 sentences reflecting on:

- What worked better than expected,
- What failed or was difficult,
- How well your metric captured “quality”,

- What you would try next with more time or compute.

Your Reflection

The experiments showed that transformer-based models performed much better than expected, even with only a few epochs of training and a relatively small dataset. The MLP classifier head in particular provided a noticeable improvement over the linear head, suggesting that additional non-linearity helped capture more complex decision boundaries. The naïve baseline performed poorly and highlighted how ineffective simple keyword matching is for real news data, where context and phrasing play an important role. One challenge during the project was ensuring stable validation accuracy across different loss functions and preventing overfitting, especially because the dataset is not very large. Accuracy worked well as a general metric for this task, although per-class precision and recall would provide deeper insight into where the model struggles. If more time or computational resources were available, I would explore fine-tuning a larger encoder model, performing data augmentation, and adding regularization methods to further improve generalization.

References

References

[1] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[2] Thomas Wolf et al. Transformers: State-of-the-Art Natural Language Processing. *arXiv preprint arXiv:1910.03771*, 2020. URL: https://github.com/huggingface/transformers

[3] SetFit Team. BBC News Classification Dataset. Available at: https://huggingface.co/datasets/SetFit/bbc-news Accessed 2025.