

---

# CAS2105 Homework 6: AG News Headline Classification 🧐

Jein Seo (2022148090)

---

## 1 Introduction

This project focuses on designing a small yet complete **AI pipeline** for classifying news headlines into four categories: World, Sports, Business, and Sci/Tech. Instead of relying on large-scale training or model fine-tuning, the assignment encourages building a compact but meaningful pipeline that compares a naïve baseline against a modern pre-trained model.

This task is interesting because news headlines are short and often ambiguous; relying solely on keyword cues can lead to incorrect predictions. By comparing a simple heuristic system with a semantic embedding model, we can clearly observe how representation learning improves classification accuracy and robustness. The work follows the assignment guidelines provided in the course material.

## 2 Task Definition

The goal of this project is to classify a single English news headline into one of four AG News categories. The task is formally defined as:

- **Input:** A headline string  $x$
- **Output:** A predicted label  $y \in \{0, 1, 2, 3\}$
- **Objective:** Learn a mapping  $f : x \rightarrow y$  that generalizes well to unseen headlines

The system is considered successful if the improved pipeline substantially outperforms the naïve baseline in accuracy and macro F1 score.

Motivationally, automated news classification is used in content curation, topic filtering, and personalized recommendation systems. Headlines are short but semantically dense, making them an ideal test case for comparing simple heuristics with embedding-based models.

## 3 Methods

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

### 3.1 Naïve Baseline

The baseline is a simple keyword-matching classifier. A small dictionary of keywords is manually assigned to each class. The classifier scans the headline and assigns the label of the first matching keyword class.

**Why this baseline?** This method represents the most intuitive and lightweight strategy for headline classification. Although simple, it captures the common intuition that certain words strongly

indicate topical categories (e.g., “team” for Sports, “technology” for Sci/Tech). It also sets a low bar for evaluating the improved pipeline.

**Expected Failure Modes** This method cannot capture meaning beyond explicit words. For example, a headline like “Tech company reports quarterly losses” is about Business, but the keyword “Tech” causes the baseline to misclassify it as Sci/Tech. Similarly, political or economic headlines lacking explicit business words confuse the baseline, revealing its inability to generalize.

### Baseline Implementation

```
def baseline_predict(text):
    text = text.lower()
    for label, keywords in keyword_map.items():
        if any(k in text for k in keywords):
            return label
    return 0 # default fallback
```

## 3.2 AI Pipeline

The improved system uses the **all-MiniLM-L6-v2** model from SentenceTransformers to generate 384-dimensional semantic embeddings. These embeddings capture contextual meaning rather than relying on specific keywords.

A logistic regression classifier is trained on these embeddings.

**Why MiniLM?** MiniLM provides a powerful balance of performance and efficiency. It generates high-quality sentence representations without requiring large computational resources, making it suitable for this assignment.

### Pipeline Stages

1. **Preprocessing:** Lowercasing and whitespace normalization.
2. **Embedding:** Convert headline into a dense vector via MiniLM.
3. **Classification:** Multiclass logistic regression predicts the final label.

### Pipeline Code Snippet

```
model = SentenceTransformer("all-MiniLM-L6-v2")
X_train_embed = model.encode(X_train)
clf = LogisticRegression(max_iter=3000)
clf.fit(X_train_embed, y_train)
```

## 4 Experiments

### 4.1 Datasets

Datasets from AG News dataset from Hugging Face [1, 2] were used. For computational manageability, downsampling was done as follows:

- 3,000 training headlines (750 per class)
- 1,000 test headlines (250 per class)

Preprocessing is done with lowercase, whitespace normalization. It is kept minimal to evaluate the robustness of the embedding model.

**Dataset Choice Justification** AG News is balanced, diverse, and widely used in text classification research. It provides enough complexity to challenge a keyword baseline but is lightweight enough to run efficiently on a CPU.

## 4.2 Metrics

Accuracy and macro F1 score are used. Accuracy reflects overall correctness, while macro F1 penalizes poor performance in any class and is appropriate given the equal class sizes.

## 4.3 Results

Method	Accuracy	Macro F1
Naïve Baseline	0.537	0.524
Embedding Pipeline	<b>0.873</b>	<b>0.874</b>

**Detailed Interpretation** The embedding pipeline achieves a substantial improvement over the baseline. The baseline struggles to distinguish ambiguous cases, particularly between Business and Sci/Tech. In contrast, MiniLM embeddings effectively encode thematic meaning, leading to high accuracy across all four classes. The improvement of +33.6 percentage points in accuracy and +35 points in macro F1 demonstrates the limitations of keyword heuristics and the strengths of semantic representation learning.

### 4.3.1 Qualitative Examples

**Example 1** **Text:** “General Mills goes whole grains...”, **True label:** Business, **Baseline:** World, **Embedding:** Business (correct)

*Analysis:* Keywords like “announced” mislead the baseline; the embedding model understands that this is a corporate product announcement.

**Example 2** **Text:** “Profit Plunges at International Game Tech... profit fell 50 percent.”, **True label:** Business, **Baseline:** Sci/Tech, **Embedding:** Business (correct)

*Analysis:* Baseline is confused by “Tech”; embedding recognizes the financial context.

## 5 Reflection and Limitations

This project reveals how dramatically model performance improves when moving from keyword rules to semantic representations. The naïve baseline performed slightly better than random chance but completely failed on headlines with ambiguous or cross-domain vocabulary. The embedding classifier, however, handled these cases gracefully and delivered robust results with simple logistic regression.

If given more time, I would explore fine-tuning a transformer model such as DistilBERT, experimenting with a confusion-matrix-driven error analysis, and increasing the size of the training set. Additionally, including uncertainty estimates or class-specific thresholds could further refine predictions.

## References

- [1] Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level Convolutional Networks for Text Classification*. NeurIPS.
- [2] Lhoest, Q. et al. (2021). *Datasets: A Community Library for Natural Language Processing*. Hugging Face.

- [3] Reimers, N. & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. EMNLP.
- [4] Pedregosa, F. et al. (2011). *Scikit-learn: Machine Learning in Python*. JMLR.
- [5] Wang, W., et al. (2020). *MiniLM: Deep Self-Attention Distillation for Task-Agnostic NLP*. ACL.