

# CAS2105 Homework 6: Mini AI Pipeline Project

## News Article Classification with a Naïve Baseline and BiLSTM Pipeline

ZHAOBOMING  
2021147547

### 1 Introduction

This project provides a gentle introduction to designing simple AI pipelines through a concrete text classification task. Rather than training very large models or conducting an extensive literature review, the project focuses on the core workflow of applied AI: problem definition, baseline design, pipeline construction, evaluation, and reflection.

In this project, I address a news article classification task using a public dataset. I first implement a simple naïve keyword-based baseline, and then compare it with an improved neural network-based AI pipeline built using a Bidirectional LSTM (BiLSTM) model. The two approaches are evaluated using standard classification metrics and qualitative error analysis.

The emphasis of this project is not on achieving state-of-the-art performance, but on understanding how and why an AI pipeline improves upon a simple heuristic method. All experiments are designed to run efficiently on a single GPU or CPU within a short time.

### 2 Task Definition

**Task description.** The task is to classify a news article into one of four categories: *World*, *Sports*, *Business*, or *Science/Technology*.

**Motivation.** News classification is a practical and widely used task in real-world systems such as news aggregation, content recommendation, and information filtering. It also serves as a clear example of how different modeling approaches handle natural language understanding.

#### Input / Output.

- **Input:** A news article represented by its title and description.
- **Output:** A single class label from {World, Sports, Business, Sci/Tech}.

**Success criteria.** The system is considered “good” if it significantly outperforms a naïve rule-based baseline on the test set, measured using classification accuracy and related metrics.

### 3 Methods

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

### 3.1 Naïve Baseline

**Method description.** The naïve baseline is a keyword-based rule system. Each input text is converted to lowercase, and predefined keywords are matched against the text. If certain keywords appear (e.g., “stock”, “market” for Business or “game”, “team” for Sports), the corresponding category is returned. If no keyword matches, the article is classified as *World* by default.

**Why naïve.**

- Does not learn from data.
- Ignores word order, context, and semantics.
- Relies entirely on manually selected keywords.

**Likely failure modes.**

- Articles without explicit keywords are often misclassified.
- Ambiguous words (e.g., “company” or “team”) can lead to incorrect predictions.
- The baseline cannot generalize to new vocabulary or paraphrases.

### 3.2 AI Pipeline

**Models used.**

- Tokenizer: Keras Tokenizer
- Model: Bidirectional LSTM (BiLSTM) neural network with an embedding layer

**Pipeline stages.**

*Preprocessing*

- Concatenate title and description
- Tokenization with a fixed vocabulary size
- Sequence padding and truncation to a maximum length of 200 tokens

*Representation*

- Trainable word embeddings
- Two stacked Bidirectional LSTM layers

*Decision*

- Global max pooling
- Fully connected layers with dropout
- Softmax output for 4-class classification

**Design choices and justification.** The BiLSTM model is chosen because it can capture sequential and contextual information in text, which the naïve baseline lacks. The architecture remains relatively lightweight to ensure short training time while still demonstrating a clear improvement over heuristic methods.

## 4 Experiments

### 4.1 Datasets

**Source.** The dataset is derived from the AG News classification dataset.

**Size.**

- Total training examples: 120,000
- Test examples: 7,600

**Train/Test split.**

- Training set: 120,000 samples
- Test set: 7,600 samples

A subset of 20,000 training samples is used for faster experimentation.

**Preprocessing steps.**

- Lowercasing
- Tokenization
- Padding and truncation to 200 tokens

### 4.2 Metrics

The following standard classification metrics are used:

- Accuracy
- Precision
- Recall
- F1-score

Accuracy is used as the primary metric for comparison, while precision, recall, and F1 provide additional insight into overall performance.

### 4.3 Results

**Quantitative results.**

Method	Accuracy
Keyword Baseline	0.4025
BiLSTM Pipeline	0.6455

The AI pipeline improves accuracy by more than 24 percentage points compared to the naïve baseline.

**Qualitative examples.**

*Example 1*

- Text: “Fears for T N pension after talks...”
- Ground Truth: Business
- Baseline: World
- AI Pipeline: Sports

Observation: The baseline fails due to missing keywords, while the AI model is misled by contextual cues.

#### *Example 2*

- Text: “The Race is On: Second Private Team Sets Launch Date for Human Spaceflight...”
- Ground Truth: Sci/Tech
- Baseline: Sports
- AI Pipeline: Sci/Tech

Observation: The AI pipeline correctly captures the technological context beyond the word “Race”.

#### *Example 3*

- Text: “Ky. Company Wins Grant to Study Peptides...”
- Ground Truth: Sci/Tech
- Baseline: Business
- AI Pipeline: World

Observation: Both methods struggle, highlighting the difficulty of borderline cases.

## 5 Reflection and Limitations

The most successful aspect of this project was the clear performance improvement achieved by the BiLSTM model over the keyword-based baseline. Even with only one training epoch, the AI pipeline significantly outperformed the heuristic approach. The baseline performed better than expected on very obvious cases but failed frequently on ambiguous or keyword-sparse articles.

One challenge was balancing model complexity with training time, especially given the size of the dataset. Accuracy proved to be a reasonable metric for overall comparison, but it does not fully capture category-specific errors. With more time or computational resources, I would explore using pre-trained transformer models such as DistilBERT and conduct a more detailed error analysis per class.

**Note:** The `train.csv` file is too large to be uploaded.