Fake News Detection Project Report

By Keerthana Saravanan (CS22B1019)

Introduction

This report is about fake news using various deep learning approaches. In the era of widespread digital media consumption, fake news detection has become a critical area of research to combat misinformation. This project implements and compares four different deep learning architectures to classify news articles as either genuine or fake.

Dataset Overview

The dataset consists of labeled news articles with the following characteristics:

- Balance of true and fake news samples
- Features include text content, publication date (separated into day, month, year), and category
- Data preprocessing involved text cleaning, tokenization, and normalization

Model Architectures

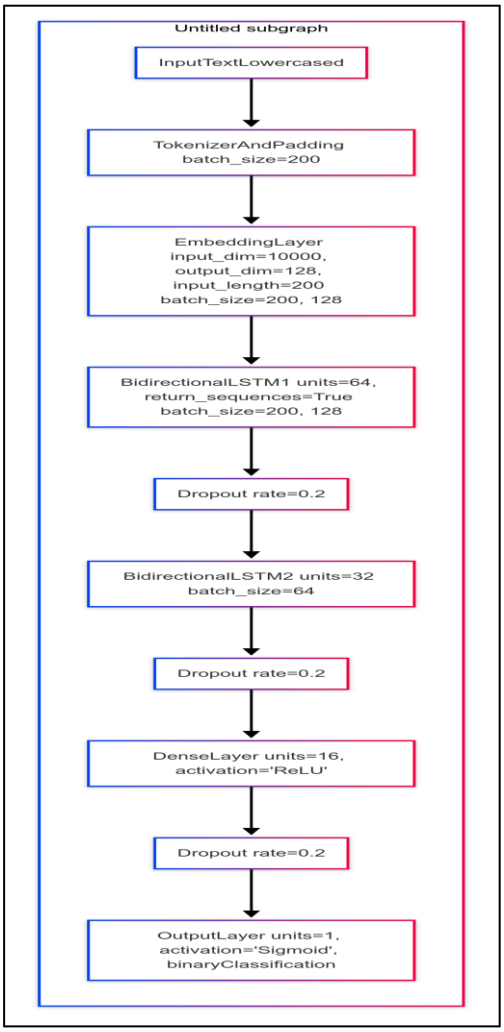
1. Bidirectional LSTM (Bi-LSTM)

Workflow:

1. Raw text is cleaned and tokenized into integer sequences
2. Sequences are padded to 200 tokens
3. Words are embedded and processed through Bidirectional LSTM layers
4. Dropout (regularization) is applied to prevent overfitting
5. A Dense layer followed by sigmoid activation provides the final classification

Advantages:

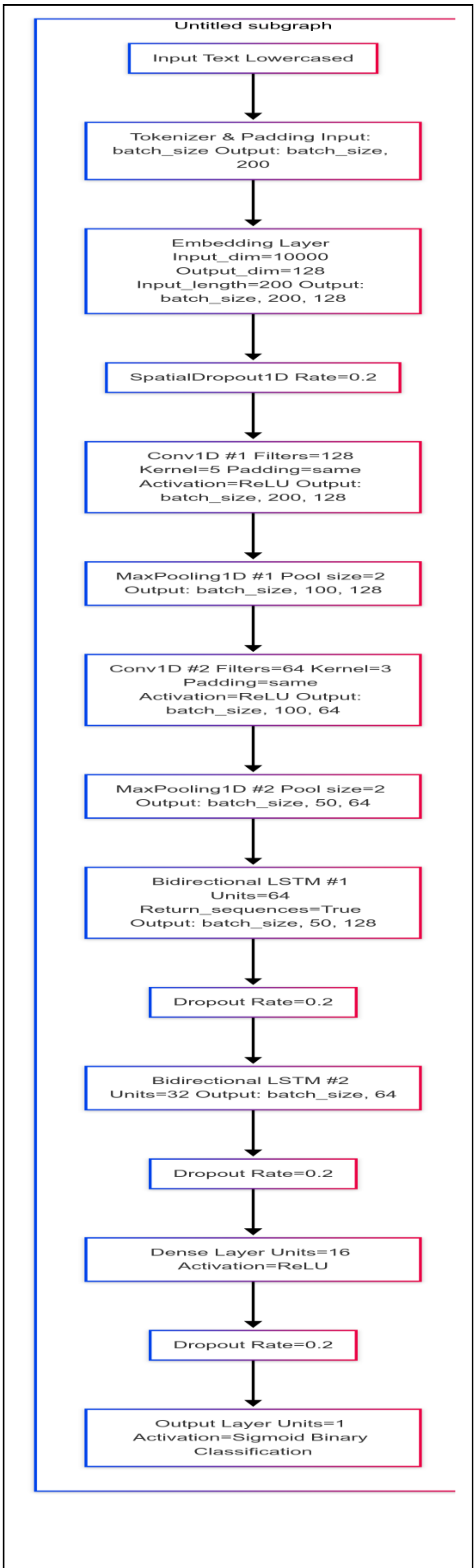
- Processes text in both directions to capture full context
- Better captures long-term dependencies than standard LSTM



Disadvantages:

- Requires custom setup to use metadata features like date/category
- Deep BiLSTMs may face vanishing gradients or slower training compared to Transformers

2. CNN-BiLSTM



Workflow:

1. Input text is cleaned, tokenized, and padded to 200 tokens
2. Tokens are passed through an embedding layer
3. A 1D Convolutional layer (CNN) extracts local n-gram features
4. BiLSTM processes the CNN output to capture sequential context
5. Dropout is applied for regularization
6. A Dense layer with sigmoid activation provides classification

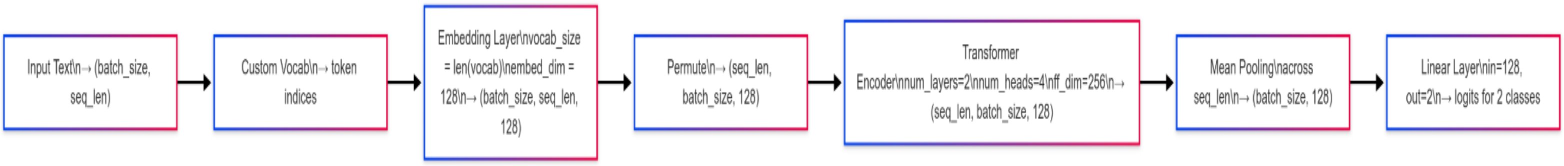
Advantages:

- Captures both local (CNN) and sequential (BiLSTM) patterns
- More efficient than pure BiLSTM as CNN reduces input size

Disadvantages:

- Slightly slower training due to stacked CNN + BiLSTM architecture
- More hyperparameters make tuning complex
- Potential gradient flow issues which can slow or destabilize training

3. Transformer-Based Classifier



Workflow:

1. Title and text are combined into a single input
2. Custom vocabulary with tokenization is created
3. Input sequences are padded to 128 tokens
4. Tokens are embedded into 128-dimensional vectors
5. A 2-layer Transformer Encoder with self-attention captures contextual relationships
6. Mean pooling followed by a fully connected layer produces the classification
7. Training uses CrossEntropyLoss and Adam optimization

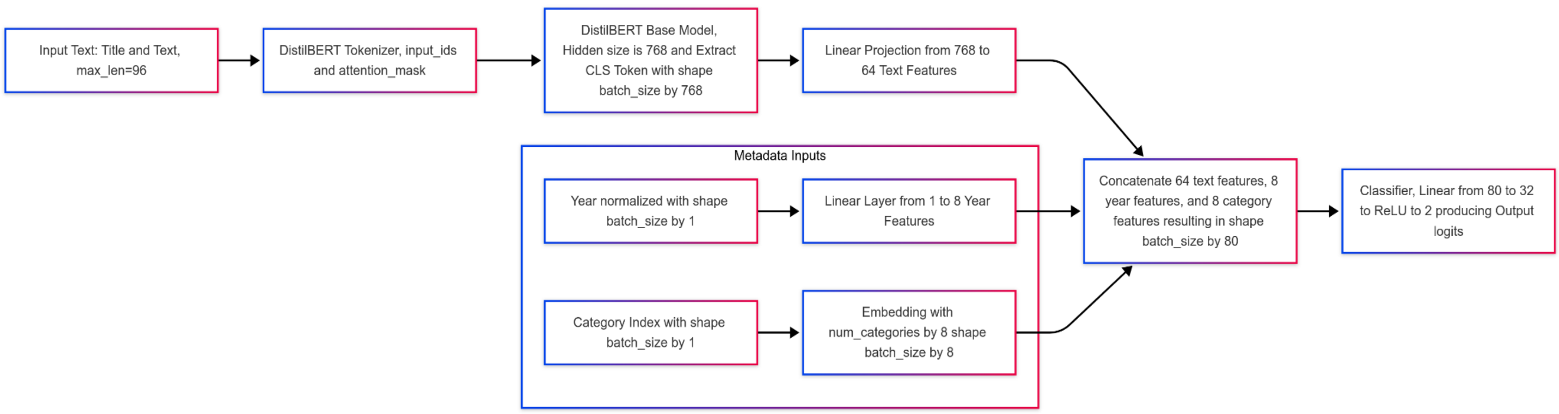
Advantages:

- Better parallelism: Processes all tokens simultaneously unlike sequential BiLSTMs
- Effectively handles long-range dependencies through direct attention mechanisms

Disadvantages:

- Requires more training data to generalize effectively
- Less effective at capturing local features without specific tuning

4. Multimodal DistilBERT



Workflow:

1. Text, year, and category are preprocessed (year normalized, category converted to numerical)
2. DistilBERT tokenizer creates token IDs and attention masks
3. A PyTorch Dataset provides batches of tokenized text, structured data, and labels
4. DistilBERT's transformer layers produce contextual embeddings
5. The [CLS] token embedding is combined with year/category vectors
6. A ReLU-activated dense layer with dropout produces the classification
7. End-to-end training uses binary cross-entropy loss and AdamW optimizer

Advantages:

- Multi-modal feature fusion combines semantic power with structured metadata
- Comprehensive evaluation with multiple metrics beyond accuracy

Disadvantages:

- Higher training complexity due to integration of structured and unstructured data
- More computationally intensive than simpler models

Comparative Analysis

Model	Architecture	Strengths	Weaknesses	Best For
Bi-LSTM	Sequential	Good for context, simpler	Slower for long texts	Medium-length articles
CNN-BiLSTM	Hybrid	Captures local patterns well	Complex tuning, training stability	Articles with distinct phrase patterns
Transformer	Parallel	Excellent for long dependencies	Data hungry	Longer articles, large datasets
DistilBERT	Transfer learning + multimodal	Uses metadata, pre-trained knowledge	Resource intensive	Complex cases requiring metadata

Conclusion

The comparison of these four deep learning architectures for fake news detection reveals important trade-offs between model complexity, computational requirements, and detection capabilities:

1. **BiLSTM** offers a good balance of performance and simplicity, making it suitable for scenarios with limited computational resources.

2. **CNN-BiLSTM** improves on BiLSTM by capturing local textual patterns, which is particularly useful for detecting stylistic elements common in fake news.

3. **Transformer-based** models excel at capturing long-range dependencies in text, which helps identify inconsistencies across an entire article.

4. **Multimodal DistilBERT** represents the most sophisticated approach, leveraging both textual content and metadata (publication date, category) to make more informed classifications.

For practical applications, the choice of model should consider the available computational resources, the size and quality of the training dataset, and the specific characteristics of the fake news being targeted. In environments with sufficient resources, the multimodal approach utilizing DistilBERT offers the most comprehensive analysis by incorporating both semantic understanding and contextual metadata.