Fake News Classification

Shasank Reddy Pinnu	23b1015
Sree Vamshi Madhav Nenavath	23b1039

May 3, 2025

Introduction & Problem Statement

This Project of Fake news detection involves classifying articles as **Real** or **Fake** based on their content.

Input Format: Textual content including the **Title** and **Body** of the article.

Output:

- **Real:** Verified true news.
- **Fake:** False or misleading news.
- **Fake News Subclassification:** Further classification of fake news into categories.

Example

Input:

Title: `The 9/11 Commission Didn't Believe the Government... So Why Should We?`

Body: `9/11 Commissioners Admit They Never Got the Full Story The 9/11 Commissioners publicly expressed anger at cover ups and obstructions of justice by the government into a real 9/11 investigation: ...`

Output:

- **Classification:** `Fake`
- **Fake News Subclassification:** `Conspiracy Theory`

Motivation

Relevance:

- The widespread growth of online misinformation and fake news undermines public trust and informed decision-making.

Why This Problem is Interesting:

- Involves real-world impact combined with natural language processing.
- Addresses societal challenges using machine learning techniques.

Applications:

- Fake news detection tools to assist media platforms and users.
- Automated fact-checking systems.
- Educational platforms for promoting media literacy.

Prior Work and Inspiration

Paper: *A Fake News Detection System based on Combination of Word Embedded Techniques and Hybrid Deep Learning Model*
Authors: M. A. Ouassil et al. (IJACSA, 2022)

Summary:

- Used a CNN-BiLSTM hybrid model for fake news detection.
- Combined CBOW and Skip-Gram Word2Vec embeddings.
- Evaluated on WELFake dataset with 97.74% accuracy.

Inspiration for Our Approach:

- Hybrid CNN-BiLSTM for learning local + sequential features.
- Framework adaptable for subclassifying fake news (e.g., conspiracy, satire).

Dataset Overview

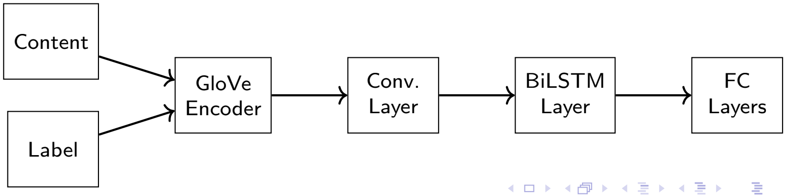
- **Data Format:**
  - Each instance includes: `type`, `title`, `content`, and `label`

Type	Title (truncated)	Content (truncated)	Label
conspiracy	The 9/11 Commission Didn't Believe...	9/11 Commissioners Admit They Never...	1
clickbait	Former Watergate Prosecutors:...	5545 SHARES Facebook Twitter...	1
hate	Hindu Group Criticizes Toronto...	A Hindu group that regularly...	1

- **Dataset Statistics:** Due to the large size of the dataset, we used a subset of the data for our experiments.
  - Total Samples: **110,000**
  - Total classes: **11** (10 types of *fake*, 1 *real*) 10,000 samples per class.
  - Training samples: **99,000**
  - Test samples: **11,000**
- **Source:** `https://github.com/several27/FakeNewsCorpus/`

Model Architecture

- For the two tasks, we trained two separate models.
  - **Primary Classifier:** Classifies articles as **Real** or **Fake**.
  - **Secondary Classifier:** Classifies **Fake** articles into 10 categories.
- Both models share the same architecture with the following components:
  - GloVe Encoder: Converts text into 100-dimensional GloVe embeddings.
  - CNN Layers: Extract local n-gram features from input embeddings.
  - BiLSTM: Captures sequential dependencies in both directions (forward and backward).
- They only differ at the final Fully Connected (FC) layers, where the number of output classes varies.
- The overview of the architecture is:



Model Architecture

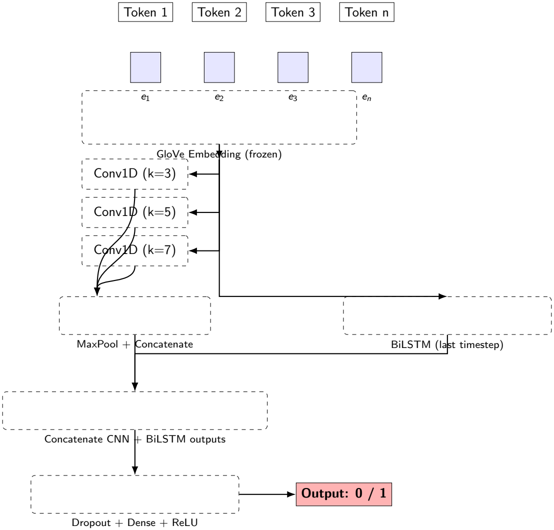


Figure: Architecture of CNN + BiLSTM model for Primary classification

Model Architecture Intuition

- Two separate models are used to allow the LSTM layers to specialize in learning different feature patterns.
- The primary model focuses on separating real and fake samples, capturing generic semantic cues.
- The secondary model is trained only on fake samples, allowing it to specialize in distinguishing fine-grained patterns among fake types.
- Training models separately ensures that each LSTM captures task-specific temporal dependencies without interference.
- This setup improves feature disentanglement, reduces overfitting, and yields better generalization on both classification stages.

Implementation Details and Libraries

Libraries:

- `torch`, `torch.nn`, `torchtext` – model and vocab handling
- `GloVe (6B, 100d)` – static embeddings via `torchtext.vocab`
- `scikit-learn, seaborn` – evaluation metrics (accuracy, F1, confusion matrix)

Model Design (Shared):

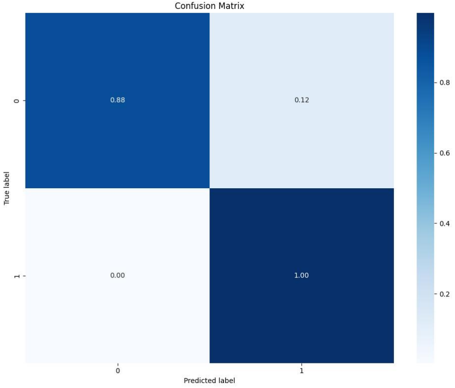
- `Embedding:` GloVe-initialized, frozen
- `CNN:` Three 1D conv layers (kernel sizes 3, 5, 7), 100 filters each
- `BiLSTM:` Hidden size = `hidden_dim`, bidirectional
- `Feature Vector:` Concatenated CNN + LSTM outputs → 400-D

Results

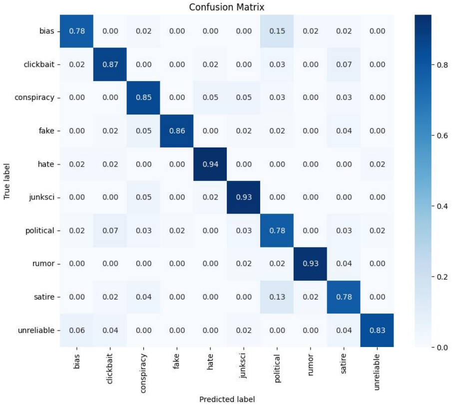
- **Primary Classifier:** Achieved a high accuracy of **98%**.
- **Secondary Classifier:** Achieved **86%** accuracy in identifying one of the 10 fake news types.
- **Single Unified Model Attempt:** We also experimented with a single model trained jointly to perform both tasks.
  - However, it could only reach around **80%** accuracy.
  - This underperformance highlights the difficulty in jointly optimizing for both coarse (real/fake) and fine-grained (type) classification.

Confusion Matrices

Primary Classifier



Secondary Classifier



Model Analysis

- **Justification for Two Separate Models:**
  - Primary Model: 97% accuracy
  - Secondary Model (Subclass Classification): 86% accuracy
  - Combined accuracy of the two models:

$$100 - 12 \times \left(\frac{1}{11}\right) - 14 \times \left(\frac{10}{11}\right) = 86.7\%$$

    which is higher than 80%. This justifies the need for separate models as the features required for the primary classifier (real vs fake) and the secondary classifier (subcategories) may differ in the LSTM part.

Note: 1/11 and 10/11 represent the fraction of real and fake data respectively. 12% is the false negative accuracy of primary classsifer and 14% is the inaccuracy of secondary model

Model Analysis

- **Primary Classifier Confusion Matrix Analysis:**
  - Some true news articles are misclassified as fake (12%) (False Negative).
  - This could be due to the skewed ratio of the training data.
  - No false articles are classified as true (True Negatives).
  - This behavior is intentional, as the model is designed to be *cynical* and err on the side of tagging content as fake.
- **Secondary Classifier Confusion Matrix Analysis:**
  - The main source of inaccuracy in the secondary classifier is in classifying articles into political, bias, and satire categories.
  - These categories are more challenging for the model, as they share overlapping features and often involve subjective interpretation.

Model Analysis

- The Primary classifier achieves an accuracy of $\sim$97%.
- This roughly matches the performance reported in the original paper (on the `WeLFake` dataset).
- Despite using a different dataset, similar results indicate strong generalization ability of the model.

Improvements Over Original Model

- **Baseline (Paper's Model):**
  - Performs only binary classification – detecting real vs fake.
  - No capability to classify different types of fake samples.
- **Our Two-Stage Architecture:**
  - **Stage 1:** Binary classifier (Primary) detects fake vs real.
  - **Stage 2:** Multi-class classifier (Secondary) classifies fake samples into subcategories.
- **Advantages:**
  - Specialized models improve precision and generalization.
  - Better handling of fake subcategories due to focused training.
  - Achieved $\sim$97% accuracy in Stage 1 and $\sim$86% in Stage 2.

What We Learned

- Understood the theory behind LSTMs and CNNs.
- Gained experience on how to improve model performance through trial and error and analysis, like
  - testing SVM on the features from LSTM outputs,
  - trying a single model for all 11 classes (fake + real), which led to inferior accuracy compared to the two-stage approach
- Got comfortable with PyTorch and torchtext for building NLP models.
- Used scikit-learn for evaluation; accuracy, F1, confusion matrix.
- Built a simple UI with Streamlit to test the model live.