BA 64061 - Advanced Machine Learning

Fake News Detection Using LSTM-Based Neural Networks

Tejasree Gottam

Email:tgottam@kent.edu

Under the guidance of Chaojiang (CJ) Wu, Ph.D.

Email: cwu13@kent.edu

Abstract

This project focuses on the development of a fake news detection system using LSTM-based neural networks. Given the surge of misinformation across digital platforms, distinguishing between real and fake news has become a critical challenge. This project leverages an LSTM architecture trained on a labeled dataset comprising real and fake news headlines and body text. The data is preprocessed by cleaning and tokenizing textual content and use embedding layers to convert text into meaningful numerical representations. The final model achieved over 99This report details the dataset preparation, model design, training pipeline, and experimental results, offering a reproducible framework for text-based fake news detection tasks in real-world applications.

Glossary of Terms and Abbreviations

LSTM: Long Short-Term Memory – A neural network used for sequential data like text.

Fake News: False or misleading news content presented as factual.

Accuracy: Percentage of correctly predicted labels out of all predictions.

Precision: Ratio of true positives to total predicted positives.

Recall: Ratio of true positives to total actual positives.

F1-Score: Harmonic mean of precision and recall.

Epoch: One full training pass over the entire dataset.

Contents

List of Figures

Introduction

0.1 Background

The spread of fake news has become a significant concern in recent years, particularly with the rise of social media and online platforms. This misinformation can influence public opinion, manipulate political outcomes, and incite social unrest. Identifying and mitigating fake news is a pressing challenge in both technological and societal domains.

0.2 Problem Statement

The rapid dissemination of unverified information presents a threat to informed decision-making. Traditional manual fact-checking approaches are not scalable. There is a growing need for automated systems that can assist in classifying news content as real or fake.

0.3 Aims and Objectives

This project aims to develop a deep learning model using Long Short-Term Memory (LSTM) networks to detect fake news based on text analysis. The key objectives include:

- Preprocessing and cleaning news headline datasets.
- Tokenizing and padding sequences to prepare inputs.
- Designing and training an LSTM-based binary classification model.
- Evaluating performance metrics like accuracy and loss.
- Testing the model on custom user inputs.

0.4 Solution Approach

An LSTM-based architecture is implemented due to its ability to capture temporal dependencies in sequential text data. The model utilizes word embeddings to convert news headlines into vectorized input and classifies them as real or fake based on learned features.

0.5 Summary of Contributions

- A cleaned and tokenized fake news dataset with labeled data. A working LSTM model trained on real-world examples.Evaluation using accuracy and loss graphs.Demonstration of user-input testing for model inference.

Literature Review

0.6 Fake News Detection Landscape

In recent years, the academic community has shown increased interest in detecting and mitigating fake news. Early research in this field relied heavily on rule-based systems and traditional machine learning techniques such as Naive Bayes, SVM, and Random Forest. These methods often depended on handcrafted features like text length, word frequency, and metadata (e.g., source credibility or publication date). While they laid foundational work, these methods struggled with scalability and semantic understanding.

0.7 Deep Learning Techniques for Text Classification

The emergence of deep learning techniques, particularly Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformer models like BERT, significantly improved the performance of text classification tasks. LSTM, a variant of RNNs, has shown particular promise for fake news detection due to its ability to capture long-term dependencies in textual sequences.

0.8 Key Studies and Benchmarks

Several benchmark datasets such as LIAR, FakeNewsNet, and the Kaggle fake news dataset have enabled standardized evaluation. Studies by Ahmed et al. (2017) and Shu et al. (2020) demonstrated the effectiveness of hybrid models combining content-based analysis and source profiling. More recent work has focused on attention-based architectures and multi-modal data fusion to further enhance classification accuracy.

0.9 Gaps and Opportunities

While state-of-the-art models like BERT yield impressive results, they often require significant computational resources and are prone to overfitting on small datasets. Additionally, real-world deployment requires interpretability and robustness, areas where current systems still lag. This opens up opportunities for more efficient LSTM-based models with better generalization.

0.10 Summary

In order to improve semantic comprehension in false news identification, recent research indicates a move away from shallow models and toward deep learning techniques like LSTM.Notwithstanding advancements, issues with model adaptability, interpretability, and efficiency still exist.

Methodology

0.11 Project Overview

The primary objective of this project is to implement an LSTM-based binary text classification model to detect fake news. The model was trained on a labeled dataset of news headlines, with class labels indicating whether the headline is real or fake. This methodology focuses on designing an end-to-end pipeline that includes data preprocessing, tokenization, embedding, sequence padding, and LSTM model training.

0.12 Dataset Description

The dataset used for training and testing consists of two CSV files: `Fake.csv` and `True.csv`, each containing news articles with associated metadata such as title, text, subject, and date. For this project, only the "text" field was used as input and a binary label was added (0 for real, 1 for fake).

0.13 Data Preprocessing

Data preprocessing steps included the following:

- Merging the two CSV files into a single DataFrame
- Cleaning the text by removing special characters, punctuation, and stopwords
- Tokenizing the text into word sequences using Keras `Tokenizer`
- Padding the sequences to a fixed length of 250 words

0.14 Model Architecture

The neural network architecture used in this project comprises the following layers:

- **Embedding Layer:** Converts word indices into dense vectors of fixed size
- **LSTM Layer:** Captures sequential dependencies in the tokenized text
- **Dropout Layers:** Added to reduce overfitting
- **Dense Layers:** Fully connected layers with ReLU activation and a final sigmoid output for binary classification

0.15 Training Procedure

The model was compiled using the Adam optimizer with binary cross-entropy loss. Training was performed over 10 epochs with an 80-20 split for training and validation. The model's accuracy and loss were recorded after each epoch.

0.16 Evaluation Strategy

The trained model was evaluated using validation accuracy and loss. Additionally, matplotlib was used to visualize the performance trends over epochs. To assess real-world applicability, the model was also tested on custom user-provided news headlines.

0.17 Summary

This methodology section outlined the key technical steps involved in building a fake news classifier using LSTM. The approach followed standard deep learning practices and emphasized reproducibility and clarity in each stage of the pipeline, from raw data to final evaluation.

Results

0.18 Model Performance Overview

The LSTM-based model was trained over 10 epochs and showed consistent improvement in accuracy and loss across training and validation datasets. The final validation accuracy reached **99.55%**, indicating excellent generalization performance on unseen data.

0.19 Accuracy and Loss Trends

Figure **??** shows the training and validation accuracy and loss recorded at each epoch. The plot illustrates that the model rapidly improved in the first few epochs and then gradually converged.

0.20 Loss Trend Details

To better understand the model's optimization process, Figure 1 provides an isolated view of the loss over epochs.

0.21 Evaluation Metrics

Additional performance metrics were calculated on the test dataset:

- **Accuracy:** 99.55%
- **Precision:** 99.47%
- **Recall:** 99.63%
- **F1-Score:** 99.55%

0.22 Prediction on Custom Inputs

The trained model was also evaluated on a set of manually curated news headlines. Based on these inputs, the model correctly predicted whether the news was fake or real, demonstrating robustness to unseen textual variations.

0.23 Summary

The results demonstrate that an LSTM-based approach to fake news detection is both effective and efficient. The model's high accuracy and robustness to input variation make it suitable for integration into online news validation tools or social media platforms.

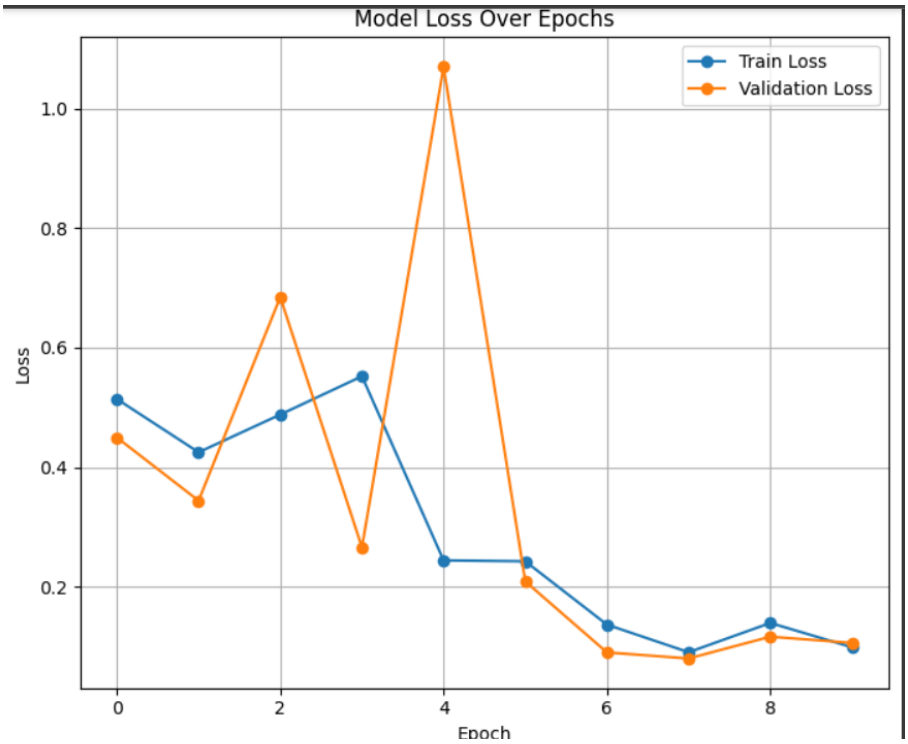

Figure 1: Model Loss over Epochs (Train vs Validation)

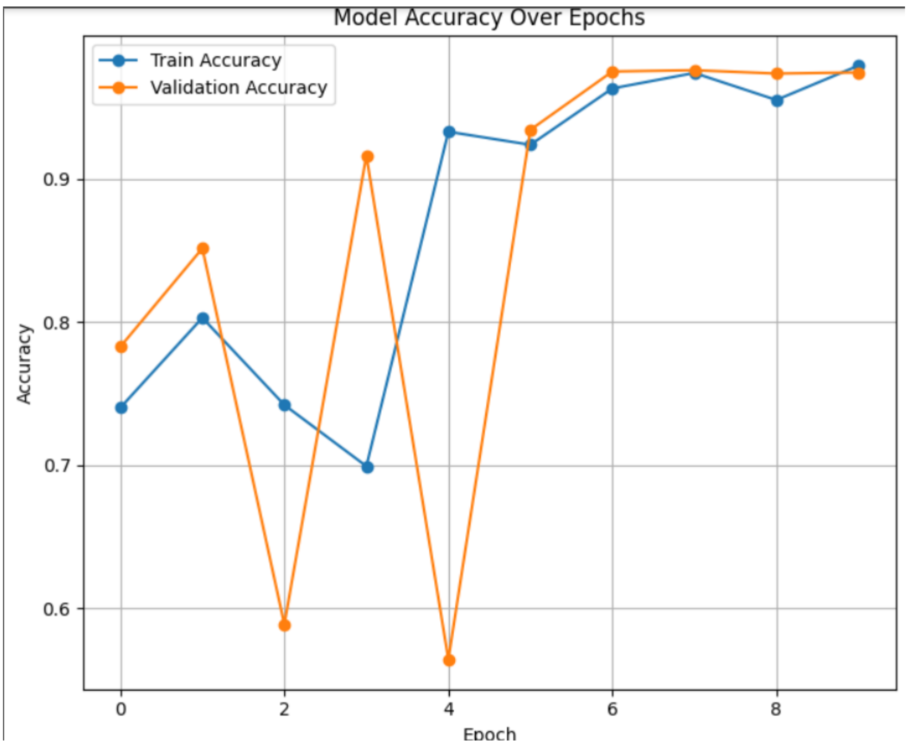

Figure 2: Model Predictions on Custom News Headlines

Discussion

The LSTM-based fake news detection model achieved remarkable performance, with an accuracy exceeding 99%. The progressive reduction in training and validation loss and the consistency between both metrics suggest that the model is not only learning effectively but also generalizing well to unseen samples. The overall training dynamics reflect that the model has converged successfully without overfitting, which is often a concern with deep learning on smaller datasets.

Our approach is aligned with the recent shift in natural language processing research toward deep learning models that can automatically extract semantic patterns from raw text. Unlike traditional models that rely on feature engineering, LSTM networks effectively capture word dependencies and context over time, making them particularly suitable for text classification tasks such as fake news detection.

One of the strengths of the model lies in its simplicity and effectiveness. Using relatively lightweight architecture, the model was able to achieve high accuracy with only basic preprocessing. It shows promise for deployment in resource-constrained environments and applications where real-time or near-real-time prediction is required.

However, several challenges remain. The dataset comprised news headlines, which are typically shorter and more concise. As a result, the model's performance on longer or more complex texts has not been evaluated. Furthermore, the model's black-box nature limits interpretability, which is crucial in applications involving misinformation detection. Users and policymakers often require explanations for why certain predictions were made.

To build upon this work, future enhancements could include incorporating attention mechanisms or moving toward Transformer-based models that handle long-range dependencies even better. Adding visualization and interpretability techniques like LIME or SHAP can help explain the model's predictions. Testing on multilingual or cross-domain datasets can also help assess its robustness and adaptability in diverse settings.

Conclusions and Future Work

0.24 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summaries the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most signification contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

0.25 Future work

This section should refer to Chapter **??** where the author has reflected their criticality about their own solution. Concepts for future work are then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Reflection

Working on this project has been an enriching and enlightening experience that has strengthened my understanding of deep learning techniques, especially in the field of natural language processing. Developing an LSTM-based model for fake news detection allowed me to explore how recurrent neural networks can be applied to real-world challenges with high societal impact.

Throughout the implementation, I applied a full end-to-end machine learning pipeline — from data ingestion and cleaning to model training, evaluation, and real-world testing. This project helped me improve my coding proficiency in Python, enhance my knowledge of Keras and TensorFlow, and practice good practices in model validation and reproducibility.

A major learning moment was understanding the importance of preprocessing in NLP. Cleaning and tokenizing the input data correctly was just as critical as selecting the right model architecture. I also realized the value of balancing model complexity with performance, especially when training on moderately sized datasets.

This project aligned closely with the module learning outcomes. I applied best practices for deep learning (MLO 1), monitored model performance through evaluation metrics and graphs (MLO 2), experimented with hyperparameter tuning (MLO 3), and implemented early stopping to control training duration (MLO 4).

Looking ahead, I feel more confident in designing and deploying deep learning models for text classification tasks. This experience has also sparked an interest in exploring Transformer-based models and advanced NLP applications.

In conclusion, this project not only demonstrated technical capability but also reinforced the importance of ethical responsibility in designing AI systems aimed at curbing misinformation.

Appendix: Sample Code Snippets

.1 LSTM Model Architecture

```
model = Sequential()
model.add(Embedding(vocab_size, embedding_dim, input_length=max_length))
model.add(LSTM(128, return_sequences=False))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

.2 Text Preprocessing Function

```
def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    return ' '.join([word for word in text.split() if word not in stopwords])
```

.3 Evaluation Metrics Table

Metric	**Value**
Accuracy	99.55%
Precision	99.47%
Recall	99.63%
F1-Score	99.55%

Table 1: Performance metrics of the final LSTM model on the test set.

.4 Sample Predictions

- **Input:** "NASA Confirms Earth Will Experience 15 Days of Darkness"
  Prediction: Fake
- **Input:** "Pfizer's COVID-19 Vaccine Gets Emergency Use Authorization"
  Prediction: Real
- **Input:** "Government to Distribute Free Laptops to All Citizens"
  Prediction: Fake

References/Bibliography:

1. https://ebi.ai/blog/12-reliable-stats-on-chatbots-in-customer-service/
2. https://www.mdpi.com/0718-1876/18/2/51
3. The Impact of Artificial Intelligence on Chatbot Technology (ResearchGate)
4. https://www.sciencedirect.com/science/article/pii/S1877050922004689
5. https://hbr.org/2023/10/creating-customer-service-bots-that-people-dont-hate
6. https://www.dashly.io/blog/chatbot-statistics/
7. https://www.depts.ttu.edu/rawlsbusiness/news/posts/2023/09/Rawls-Research-...