Detect fake news using Spark NLP and deep learning models.

Trupesh Prajapati
240031105151010

Introduction

- Fake News is a challenging problem in today's times.
- Social Media websites are flooded with much misinformation, which can prove fatal.
- Twitter particularly struggles with the fake news problem.
- However, there is a certain regular pattern in fake news. Some individuals are more likely to spread fake news.
- We can use Machine Learning to identify such patterns and try to predict fake news.

Introduction

What is Apache Spark?

Apache Spark is an open-source **unified analytics engine** designed for **large-scale data processing**. It was originally developed at UC Berkeley and is now one of the most widely used big data frameworks.

🚀 Key Features of Spark

Feature	Description
In-Memory Computing	Stores intermediate results in memory (RAM), making it much faster than Hadoop MapReduce.
Distributed Processing	Automatically distributes data and computation across multiple nodes in a cluster.
Multi-language Support	Supports Python (PySpark), Scala, Java, and R.
Fault Tolerant	Automatically recovers from node failures using lineage and DAGs.
High-level APIs	Simplifies working with big data using DataFrames, Datasets, and SQL.
Versatile Workloads	Handles batch processing, streaming, machine learning, and graph processing.

Introduction

Core Components of Apache Spark

1. **Spark Core**
   - The foundational engine for basic I/O, scheduling, task distribution, etc.

2. **Spark SQL**
   - Supports structured data processing with SQL queries and DataFrames.

3. **Spark Streaming**
   - Processes real-time data streams.

4. **MLlib (Machine Learning Library)**
   - Provides scalable ML algorithms like classification, regression, clustering.

5. **GraphX**
   - For graph computation and analysis (less commonly used today).

Project Objective

- To develop a deep learning model that can accurately detect fake news.
- Use spark natural language processing (spark NLP) techniques to analyze the content of news articles.
- Demonstrate the application of Python, data preprocessing, and LSTM-based deep learning.

Dataset Description

- Dataset includes labeled news articles (Real/Fake).
- Fields include: id, title, and author.
- Source: Kaggle or a similar public dataset repository.
- Data Cleaning: removed null values and duplicates.

Data Preprocessing

- Converted text to lowercase.
- Removed punctuation, special characters, and stopwords.
- Applied tokenization and padding.
- Optional: Lemmatization or stemming for word normalization.

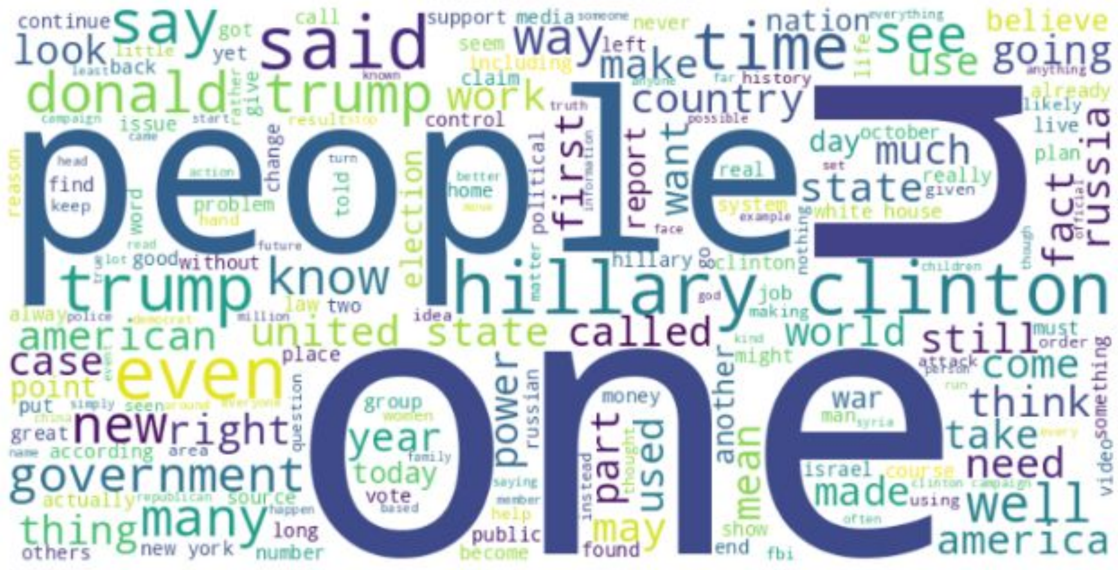
Exploratory Data Analysis (EDA)

- Visualized the number of real vs fake news articles.
- Identified most frequent words in each class.
- Analyzed text length distributions.
- Optional: Word clouds or bar graphs to show patterns.

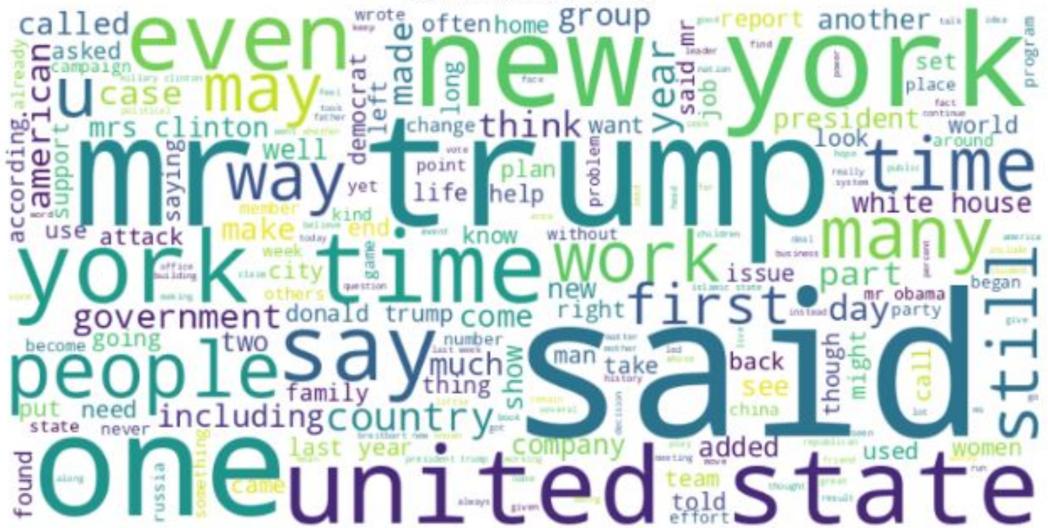
All News Word Cloud

Exploratory Data Analysis (EDA)

Fake News Word Cloud



Real News Word Cloud



Text Vectorization

- Used Tokenizer to convert words to numeric sequences.
- Applied padding to ensure equal input length for deep learning.
- Optional: TF-IDF or Word2Vec for feature extraction.

Model Architecture

•Used an **LSTM (Long Short-Term Memory)** network.
•Model includes:
•Embedding layer (for word vectors)
•LSTM layer (for sequence learning)
•Dense layer with sigmoid activation
•Dropout layer used to prevent overfitting.

Model Compilation

- Loss function: **Binary Cross-Entropy** (since it's a binary classification task).
- Optimizer: **Adam** (adaptive learning).
- Metrics: Accuracy, Precision, Recall, F1-score (optional).

Model Training

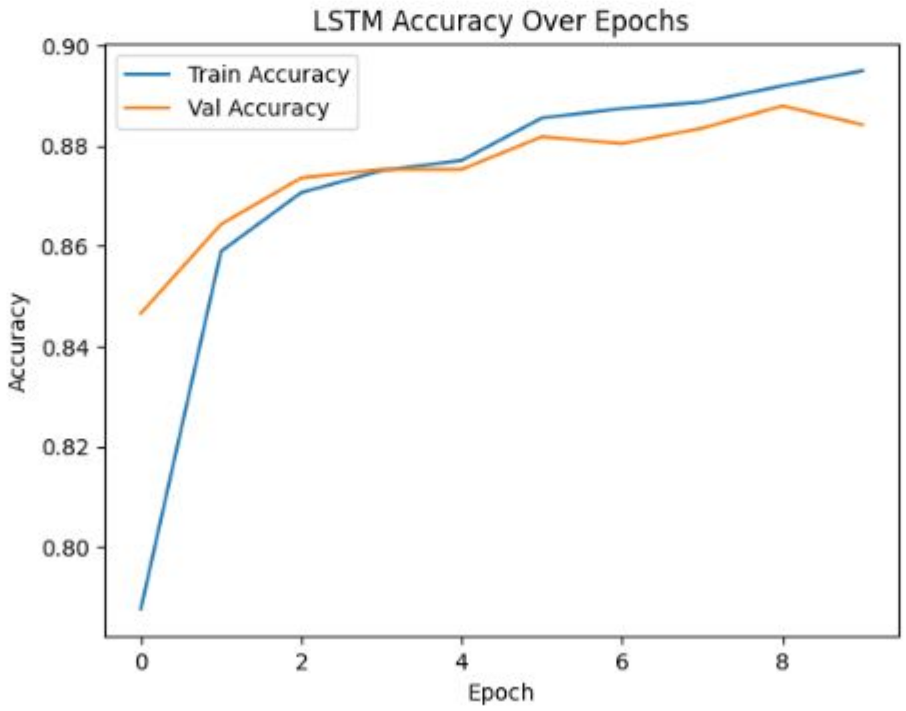
- Split data into training and validation sets.
- Trained model for 10 epochs with a batch size of 366.
- Observed accuracy/loss curves for overfitting/underfitting.
- EarlyStopping callback used to stop training when validation loss stops improving.

Model Evaluation

- Evaluated performance using test data.
- Metrics: Accuracy, Precision, Recall, F1-Score.
- Visuals: Confusion Matrix to show true/false positives/negatives.
- Optional: ROC curve to evaluate classification threshold.

Results and Observations

- Achieved 89% accuracy on test set.
- The model performs better on Real/Fake (mention any bias).
- Misclassifications typically involve ambiguous or satire content.



Conclusion

- Successfully implemented a deep learning model for fake news detection.
- Highlighted importance of data preprocessing and model tuning.
- Showcased Python, Spark NLP, and deep learning integration.

Future Work

- Improve dataset size and diversity.
- Use advanced models (e.g., BERT, GPT-based transformers).
- Implement real-time fake news detection tool.
- Explore multilingual fake news detection.


SONY
SAB
THANK YOU