FAKE NEWS DETECTION USING NATURAL LANGUAGE PROCESSING (NLP)

Name: Prarthana B R

Reg.No: 513521104037

Department: CSE

Year: III

NM ID: au513521104037

E-Mail:prarthuravi13@gmail.com

Phase 2 (INNOVATION)

INTRODUCTION:

The spreading of fake news causes many problems in the society. It easily deceives people and leads to confusion among them. It has the ability to cause a lot of social and national damage with destructive impacts. Sometimes it gets very difficult to know if the news is genuine or fake. Therefore it is very necessary to detect if the news is fake or not.

Natural Language Processing (NLP) and deep learning techniques are powerful tools for fake news detection. In this documentation, we will explore the use of an **innovative LSTM (Long Short-Term Memory) model** for fake news detection.

The following are the steps that are to be performed sequentially in the process of fake news detection.

1. DATASET PREPARATION :

To build and train our LSTM model, we need a labeled dataset of real and fake news articles. We need to ensure that the dataset is balanced and appropriately labeled to ensure reliable model training and evaluation. So we choose a dataset that matches our requirements from the Kaggle.

The link for the dataset we will be using for this project is

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

2. DATA PREPEROCESSING:

i. Tokenization:

Tokenization is the process of splitting text into individual words or sub-words. This step breaks down the text into its constituent elements, making it more manageable for further processing. For example, the sentence "I love learning AI" would be tokenized into ["I", "love", "learning", "AI"].

ii. **Lowercasing:**

Convert all the text to lowercase. This ensures that words with different capitalizations are treated as the same word. For instance, "Apple" and "apple" becomes "apple".

iii. **Removing Stop Words:**

Remove common words like "and," "the," "is," etc., that may not carry much meaning.

iv. **Removing Special Characters:**

Eliminate punctuation and non-alphanumeric characters.

Proper data pre-processing ensures that the LSTM model receives clean, structured, and meaningful input data, which is essential for effective learning and improved model performance in fake news detection and other NLP tasks.

3. FEATURE EXTRACTION:

In the context of LSTM (Long Short-Term Memory) models for natural language processing (NLP) tasks like fake news detection, feature extraction primarily involves converting textual data into numerical representations that can be fed into the LSTM model.

i. **Word Embedding:**

Word embeddings are **dense numerical representations of words that capture semantic relationships** between words in a continuous vector space. These embeddings provide a way to represent words in a way that the model can understand.

Common pre-trained word embeddings include **Word2Vec, GloVe,** and **FastText.** Alternatively, we can train your own word embeddings during model training.

ii. **Padding and Truncation:**

LSTMs require input sequences of the same length. However, natural language text often comes in variable-length sequences. To handle this, you may need to pad shorter sequences with zeros or truncate longer sequences to ensure uniformity.

Padding adds zeros to the end of sequences until they reach a specified length, while truncation removes excess words from sequences that are too long.

4. LSTM MODEL ARCHITECTURE:

The following are the possible layers that are present in the LSTM architecture.

i. **Embedding Layer:**

Convert numerical vectors (word embeddings) into continuous representations.

ii. **LSTM Layers:**

Stack multiple LSTM layers to capture sequential dependencies effectively.

iii. **Dropout Layers:**

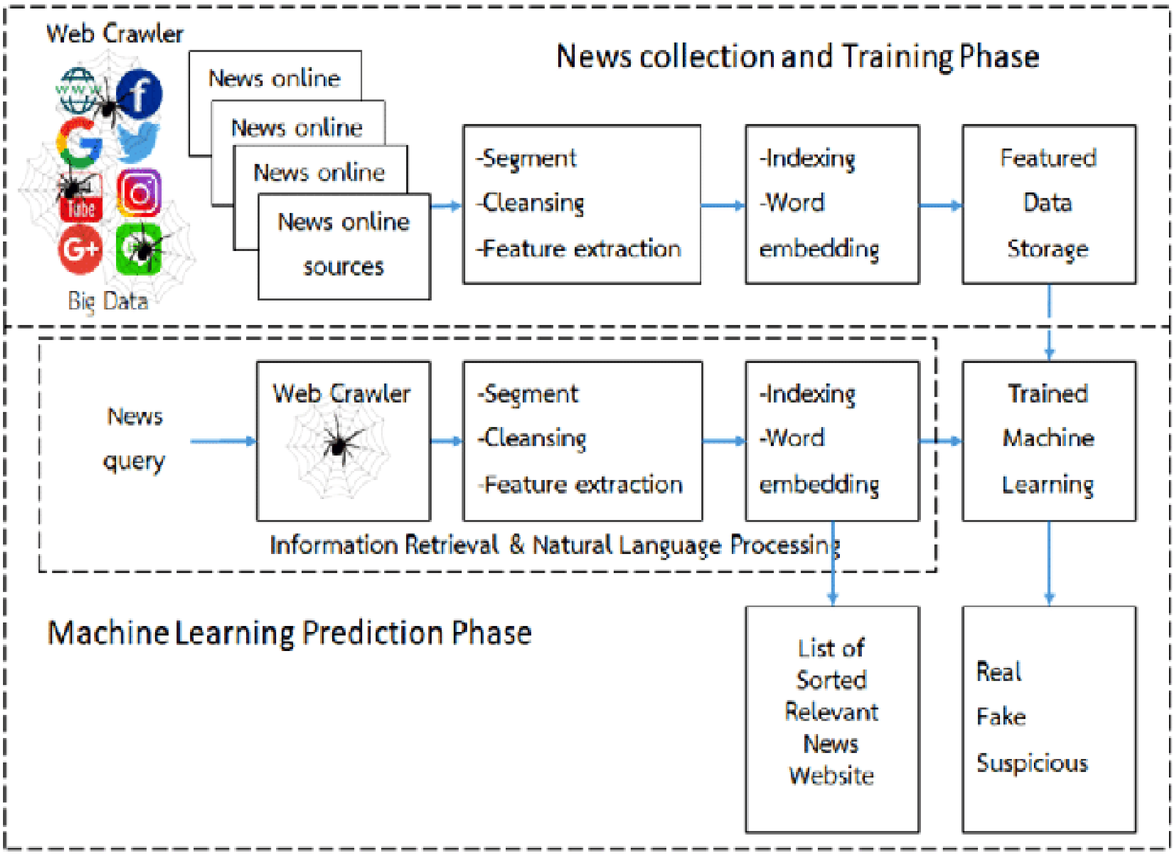
Prevent over-fitting by randomly dropping out a fraction of neurons during training.

iv. **Dense Layers:**

Add one or more dense layers for classification.

v. **Output Layer:**

Use a sigmoid activation function to produce a binary classification result (fake or real).



5. MODEL TRAINING:

The following parameters are used in training the model.

i. **Data Splitting:**

Split the dataset into training, validation, and test sets (e.g., 70% for training, 15% for validation, 15% for testing).

ii. **Model Compilation:**

Compile the LSTM model with an appropriate loss function (e.g., binary cross-entropy) and optimizer (e.g., Adam).

iii. **Training:**

Train the model on the training data and monitor its performance on the validation set. Use techniques like early stopping to prevent over-fitting.

6. MODEL EVALUATION:

i. **Evaluation Metrics:**

Assess the model's performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

ii. **Confusion Matrix:**

Visualize the model's performance using a confusion matrix.

CONCLUSION:

We saw the detailed steps of using the LSTM model in the Fake News Detection. Using these steps it will be easy for us to convert the design in the project. The next step is to implement in terms of the PYTHON code.