"Detecting Fraudulent News in Immigration and Travel Agencies using LSTM Neural Networks"

Introduction

In today's digital age, the proliferation of fake news has become a significant concern, especially when it comes to immigration and travel agencies. Many agencies resort to spreading fraudulent information, such as fake university admissions, immigration offers, and discounted scholarships, to lure unsuspecting individuals. The objective of this machine learning project is to develop a robust fake news detection system using recurrent neural networks (RNNs). By leveraging Long Short-Term Memory (LSTM) networks and natural language processing (NLP) techniques, we aim to create a model capable of accurately identifying and classifying fake news propagated by these agencies. The dataset used for this project includes reports of fraudulent activities, detailing the deceptive tactics employed by these agencies, thus providing a means to safeguard individuals from falling victim to false promises and deceptive schemes.

Data preprocessing

Data collection and source: The data is collected from two CSV files, one containing true news and the other containing fake news. The sources of these news articles are not explicitly mentioned. The dataset is sourced from Kaggle because a specific dataset on fraud immigration news could not be found. Therefore, a random news dataset was utilized for this project.

Data cleaning and handling missing values: The Pandas library is used to read the CSV files. Before performing any analysis, the data frames are checked for missing values. In this case, both data frames (`DF_true` and `DF_fake`) do not contain any missing values.

Data exploration and visualization: The data frames are explored to understand the structure of the data and its distribution. Techniques such as `info()` are used to check memory usage and the presence of null elements. Additionally, the number of data samples per class is calculated to understand the class distribution. The Seaborn and Matplotlib libraries are used for data visualization to explore different aspects of the data.

Feature Engineering

Combining columns: Two additional columns, "is_fake" for classifying news as fake (1) or real (0) and "original" for combining the title and text of the news articles, are added to the data frame. This feature engineering step prepares the data for the subsequent tasks of data cleaning and model training.

Dropping unnecessary columns: The "date" column, which is deemed unnecessary for the classification task, is dropped from the data frame.

Text preprocessing: The "original" column is created by combining the "title" and "text" columns. In the next steps, this column will undergo text preprocessing tasks such as removing stop words and tokenization to prepare the text data for model training.

Data Cleaning

Stop Words Removal: Downloaded and removed stop words (common words like 'the', 'is', 'at') from the text data using the NLTK library, enhancing the dataset by eliminating words that do not add significant value to text analysis.

Function Definition for Text Preprocessing: Defined a function pre_process that uses Gensim's simple_preprocess to tokenize text data. The function ensures that only meaningful words (excluding stop words and words shorter than three characters) are kept.

Applying Preprocessing to Dataset: Applied the pre_process function to the combined 'Original' column, creating a new column 'Clean' in the DataFrame that contains the cleaned text data without stop words and insignificant words.

Joining Cleaned Words: Created a new column 'Clean_Joined' where all words in the 'Clean' column were joined together into one string per entry, facilitating further analysis and visualization.

Data Visualization

Visualizing Subject Distribution: First, we visualized the distribution of subjects in our dataset using a count plot. This plot helps us understand the frequency of different news subjects. The count plot showed that the "Politics News" category had the highest count, indicating that our dataset is predominantly composed of political news.

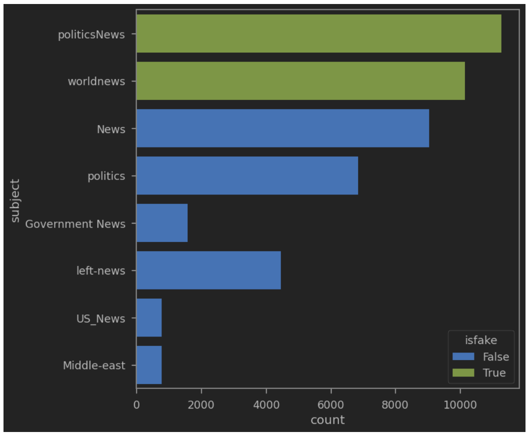

Image: Subjective Distribution

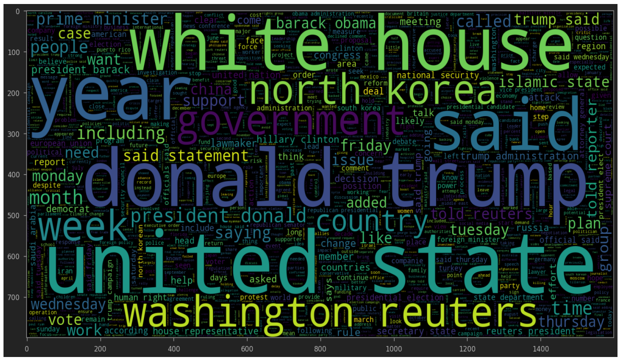

Image 1: For True data

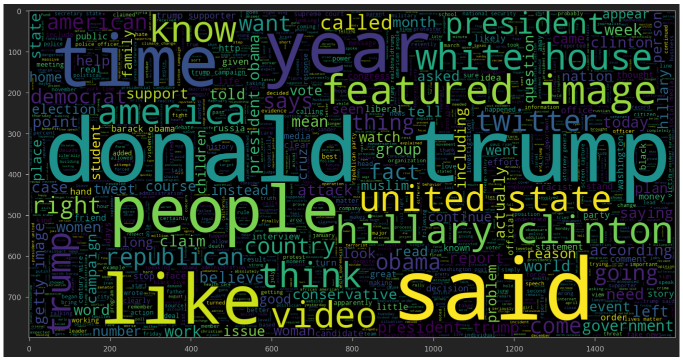

Image 2: For fake data

Visualizing Fake vs. Real News: Next, we visualized the distribution of fake and real news articles. This step helps ensure that our dataset is balanced. The resulting plot confirmed that our dataset is nearly balanced, with a comparable number of fake and real news articles.

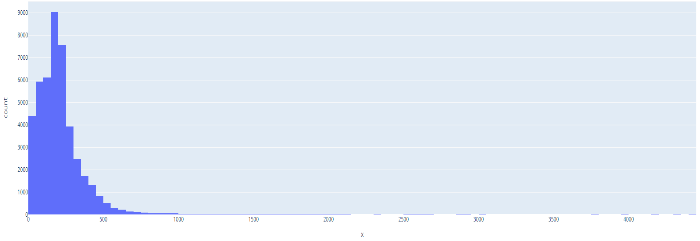

Image: Word Visualization

Word Cloud Visualization
To gain insights into the most frequently used words in fake and real news articles, we created word clouds. Word clouds display words in varying sizes based on their frequency in the text, providing a visual representation of the most common terms.
For fake news, the word cloud revealed terms such as "Trump," "Clinton," "Republican," and "Democrat," which were prominent, reflecting the political nature of our dataset. Similar terms appeared in the word cloud for real news, indicating a common focus on political topics.

Model Selcetion and Training

Splitting the Data: We used the train_test_split function from Scikit-learn to divide our dataset into training and testing sets. Specifically, 80% of the data was allocated for training, and 20% was allocated for testing. This is crucial to ensure that our model can generalize well to new, unseen data and not just memorize the training data.

Tokenization: Tokenization is the process of converting text into sequences of integers. This transformation is essential because machine learning models cannot directly process raw text. We utilized the Tokenizer from the Natural Language Toolkit (nltk) library for this purpose. The fit_on_texts method was used to create a vocabulary index based on the training data, and the texts_to_sequences method transformed the text into sequences of integers.

Padding Sequences: To ensure that all input sequences have the same length, we used padding. Padding is the process of adding zeros to the sequences to make them uniform in length. We set the maximum length to 4405, which is the maximum number of words found in any document in our dataset. This step is crucial for feeding the data into our LSTM network, which requires input sequences of the same length.

Results and Analysis:

The project aimed to develop a model capable of detecting fake news. It started with understanding the problem statement and business case, followed by importing necessary libraries and the dataset. Feature engineering and data cleaning were performed to prepare the dataset for modeling, accompanied by data visualization to gain insights. The data was then prepared by tokenization and padding, and the project explored Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Using Keras and TensorFlow 2.0, an LSTM model was built and trained. The model's performance was assessed using testing data, and it achieved a high accuracy of 99%. This indicated the model's effectiveness in distinguishing between real and fake news articles.

Discussion

In this guided project, we developed a fake news detection model using LSTM (Long Short-Term Memory) networks. We started by understanding the problem statement and the business case, which involved developing a model capable of determining whether a news article is real or fake.

We then proceeded to import the necessary libraries and datasets. Feature engineering and data cleaning were performed to prepare the data for modeling. Visualization techniques such as word clouds and Plotly were utilized to gain insights into the data.

The core of the project involved understanding recurrent neural networks, specifically LSTM networks, and their ability to handle time-series data. We learned about the limitations of feedforward neural networks and how LSTM networks overcome the vanishing gradient problem.

Following this, we built and trained our LSTM model using Keras and TensorFlow. We achieved impressive results with an accuracy of around 99% on the testing data. We also visualized the model's performance using a confusion matrix to understand the misclassifications.

Conclusion

In conclusion, we successfully developed a fake news detection model using LSTM networks. The model demonstrated high accuracy in classifying news articles as real or fake. This project highlighted the power of LSTM networks in handling sequential data and demonstrated their effectiveness in natural language processing tasks like fake news detection. Additionally, we gained insights into the challenges of training recurrent neural networks and learned how to overcome them using techniques like LSTM networks. Overall, this guided project provided valuable hands-on experience in developing and training deep learning models for text classification tasks.

References:

1. https://delphieducation.co.uk/protecting-yourself-from-study-abroad-scams-a-comprehensive-guide-for-aspiring-students/
2. https://cis.org/North/Victims-MarriageRelated-Immigration-Fraud-Tell-Their-Stories
3. https://www.business-standard.com/finance/personal-finance/foreign-job-scams-how-to-outsmart-fraudsters-targeting-job-seekers-124051400913_1.html