Sophia Colonello, Soomin Han, Chelsea Calalb, Adriana Schermaier

Dr. Dalia Sulieman

CS 4801-51: Deep Learning

December 9, 2024

Final Report

Project Goal

The goal of this project is to determine if a piece of news-related text is real or fake. Real news would be content that directly relates to a true event, whereas fake news is typically seen as misleading or untruthful. In a world where news is present everywhere, it is important to utilize technology to flag news that may misinform everyday people. The spread of fake news can have serious consequences such as influencing public opinion and fueling social unrest. This project aims to detect the patterns present in fake news and contribute to the reduction of misinformation.

Deep Learning Background

Deep learning is a subset of machine learning that involves training artificial neural networks to model complex patterns in data. A neural network consists of interconnected layers of nodes, or neurons, which mimic the way the human brain processes information (Holdsworth & Scapicchio). Each node takes inputs, applies weights and biases to them, processes the results through an activation function, and passes the output to the next layer. Neural networks are composed of input layers, which receive raw data, hidden layers, which perform computations, and output layers, which produce predictions. The hidden layers, often stacked deeply in deep learning models, allow the model to learn hierarchical features, making deep learning

particularly effective in fields like image recognition, speech processing, and natural language understanding.

During training, the network learns by adjusting its weights and biases to minimize the error between its predictions and the actual outcomes. This error is quantified using a loss function, a key concept in deep learning. Common loss functions include mean squared error for regression tasks, which measures the average squared difference between predictions and true values, and cross-entropy loss for classification tasks, which quantifies the difference between predicted probabilities and actual class labels (*Neural Networks 101*). The choice of loss function depends on the problem being solved and directly impacts how the model learns.

To optimize the network's parameters and minimize the loss, algorithms such as Stochastic Gradient Descent (SGD) and Adam are commonly used. Gradient descent calculates the gradient of the loss function with respect to the model's parameters and updates them iteratively to move toward a minimum (Agrawal). Variants like SGD with momentum or Adam introduce additional techniques to accelerate convergence and stabilize training. For instance, Adam adapts the learning rate for each parameter based on the estimated first and second moments of the gradient, making it robust and widely used.

Activation functions are another fundamental component, introducing non-linearity to the network and enabling it to learn complex patterns. One popular activation function is ReLU (Rectified Linear Unit), which outputs zero for negative inputs and the input itself for positive values, making it computationally efficient and effective for deep networks (Agrawal). Other functions, like sigmoid and tanh, are used in scenarios where outputs need to be bounded, such as in probability estimation or feature scaling.

Deep learning models train over multiple epochs, where each epoch represents a complete pass through the entire training dataset. To improve training efficiency and handle large datasets, data is typically processed in smaller batches during each epoch. Techniques like dropout are often employed to prevent overfitting by randomly disabling neurons during training, encouraging the network to learn more generalized representations. Similarly, batch normalization is used to normalize input data within each batch, leading to faster convergence and improved stability.

Deep learning models span a wide variety of architectures tailored to specific data types. For instance, Convolutional Neural Networks (CNNs) are ideal for image data, as they extract spatial features using convolutional layers, while Recurrent Neural Networks (RNNs) and their variants like LSTMs and GRUs are well-suited for sequential data, such as time series or text (Holdsworth & Scapicchio). These architectures, combined with the concepts of loss functions, optimization algorithms, and regularization techniques, enable deep learning to excel in diverse applications ranging from autonomous vehicles to healthcare and beyond.

Literature Review

To guide the approach in developing a model for fake news detection, the team conducted a literature review of related research that explored various machine learning and deep learning methods of fake learning classification.

In the proceedings of the 6th International Conference on Electronic Information Technology and Computer Engineering, Lu Huang (2022) presented a comprehensive study on fake news detection, emphasizing the limitations of traditional machine learning approaches like Support Vector Machines and Naïve Bayes classifiers. These models are critiqued for their reliance on manual feature selection, which is data-specific and dependent on the expertise of

individuals, leading to variability in data quality. Huang's research employs deep learning strategies to address these issues, proposing a novel model called CSI (Capture, Score, and Integrate) that leverages neural networks to improve accuracy and consistency in fake news detection. Additionally, the study explores the use of Latent Dirichlet Allocation (LDA) for sentiment analysis, demonstrating improved model performance when sentiment is incorporated. Huang also provides an in-depth discussion of the challenges posed by the diverse styles and patterns of fake news, especially in the context of social media, and underscores the importance of comprehensive datasets and robust evaluation metrics in advancing this field.

Mishra and Sadia (2023), in the paper "*A Comprehensive Analysis of Fake News Detection Models: A Systematic Literature Review and Current Challenges*", provide a systematic review of fake news detection models, categorizing them into supervised, unsupervised, and hybrid learning approaches. Their analysis highlights the limitations of supervised models, such as the need for well-labeled datasets, which are often resource-intensive to produce, and the challenges posed by evolving misinformation patterns. Unsupervised methods, while more adaptable, generally offer lower accuracy due to the absence of labeled data during training. Hybrid methods aim to balance these trade-offs but face difficulties with multimodal and multilingual content. The authors also evaluate popular NLP techniques, such as Word2Vec, BERT, and TF-IDF, which play crucial roles in the semantic analysis of fake news. The study identifies scalability, linguistic diversity, and cross-lingual applicability as key challenges for future research.

Padalko, Chomko, and Chumachenko (2023) explore the use of Bidirectional Long Short-Term Memory (BiLSTM) models for fake news classification in their paper titled, "*A Novel Approach to Fake News Classification Using LSTM-based Deep Learning Models"*. The

BiLSTM architecture, which processes input sequences in both forward and backward directions, is noted for its ability to capture contextual dependencies and nuanced language more effectively than traditional LSTM models. The authors report high accuracy rates for BiLSTM but highlight potential challenges in scaling the model for larger datasets and real-time applications. This study provides valuable insights into the potential of BiLSTM models while acknowledging the need for addressing scalability and real-time applicability in future implementations.

In the paper "*Detection of Fake News Using Deep Learning CNN–RNN Based Methods",* Sastrawan, Bayupati, and Arsa (2022), focus on the integration of CNN and RNN architectures, enhanced with pre-trained word embeddings like Word2Vec, GloVe, and fastText, for fake news detection. The study emphasizes the importance of data preprocessing and augmentation, employing techniques like back-translation to improve class balance and model performance. The experiments demonstrate that BiLSTM combined with embeddings such as GloVe and fastText achieves superior accuracy across multiple datasets. The findings suggest that combining deep learning architectures with robust pre-processing and data augmentation techniques significantly enhances model consistency and effectiveness.

This research showed the value of techniques such as data augmentation and the tackling of issues such as scalability and adaptability. These findings guided the project and influenced the approach to developing a more effective model.

Data Selection

The dataset chosen for this fake news detection project was found on Kaggle. The dataset, titled "Fake News Classification" consists of 72,134 news articles. 35,028 of those articles are real, while 37,106 are fake. The authors of this dataset combined data from four popular news datasets, those being Kaggle, McIntire, Reuters, and BuzzFeed Political. Each row

consists of the article title, the article content, and the real/fake classification label. It is important to note that determining whether an article is real or not is subjective. Therefore, the group used trusted news sources with professional journalists to ensure the data was labeled to the best of its ability (Shahane).

Implementation & Methodology

An important part of any machine learning task is, of course, the data. Ensuring that the data being fed into the model is "clean" is vital. The title and text columns of this dataset were first examined for NA values, and it was found that there were 558 instances of the title missing, and 39 instances of the text missing. There were no instances of both title and text missing, so no NA values were removed in the first version of this model. However, the question still remained if any of the NA values were affecting the model and its training, so a second iteration of this model was created. While the first model combined the title and text columns regardless if one was empty, the second version dropped all rows that had the text portion missing. This was chosen over removing all NAs as the text portions are much longer, and not having them significantly limits the words the model has to make a prediction with, while titles are shorter and may have less of an effect.

The title and text columns were then combined into a single content column to represent the full article. If there were rows with an NA value in one column, it was transformed into an empty string before being concatenated with the other column. Next, the dataset was tokenized using the Natural Language Toolkit (NLTK), splitting each article into words to create a vocabulary of unique tokens. This vocabulary was used to assign integer IDs to words, enabling text encoding for model consumption. Each article was then encoded as a sequence of integers. To prepare the data for the model, sequences were padded to a fixed length of 100 words,

ensuring consistency across input samples. The length of 100 was also chosen due to some outliers in the dataset containing extremely long text values, causing the kernel to crash.

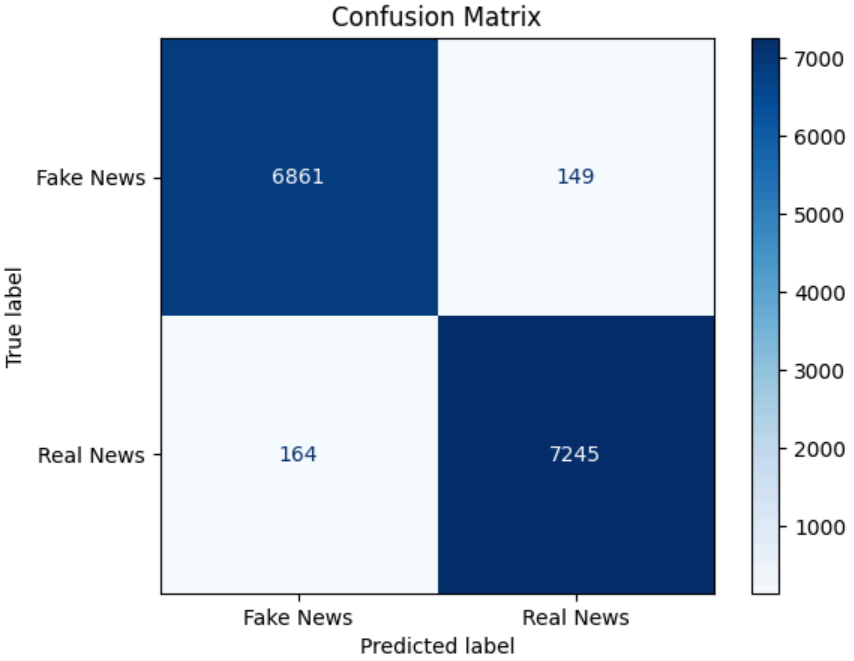
To detect misinformation and fake news, the group decided that the most appropriate model to use would be the BiLSTM model. A BiLSTM model is a sequence model which includes two LSTM layers. One layer processes input data in a forward direction and the other processes the input data in a backward direction. This approach allows a model to understand the relationship between sequences (GeeksForGeeks). Other models were also considered throughout the model selection process such as CNN, BERT, and RoBERTa, but ultimately, the BiLSTM was selected as opposed to others due to its high accuracy and practical applications.

The dataset was split into training and testing sets, with 80% (57,676 rows) going into training and 20% (14,419 rows) going into testing. A custom PyTorch Dataset class was created to structure the data for loading and batching. The BiLSTM model structure was then defined. It included an embedding layer for mapping integer word IDs to dense vector representations, a bidirectional LSTM layer for processing sequences in both forward and backward directions to capture contextual dependencies, and a fully connected layer for predicting binary labels. The model was optimized using the Adam optimizer, and a binary cross-entropy loss function with logits was used for classification.

While selecting and building a model is an important step, tailoring the model to make accurate predictions is arguably even more important. With this in mind, the team made the conscious decision to assign the model with specific hyperparameters. These hyperparameters included 50 embedding dimensions, 64 hidden dimensions, a learning rate of 0.001, and five epochs. Training involved iterating over batches of data for these 5 epochs, calculating loss, and updating model weights through backpropagation.

Results & Analysis

For evaluation, the model predictions were compared against the true labels to calculate accuracy on the test set. The first model, which did not remove any NA values, achieved an accuracy of 97.8%. This indicates that the words within articles have strong predictive power in determining whether or not a piece of text is real or fake news, and that the team's BiLSTM model was highly successful in making those predictions. The second model, which dropped all rows that had the text portion missing, achieved a very slightly higher accuracy of 97.83%. To further understand these results, a confusion matrix was created.



It can be seen that the model predicted 6861 true positives, 7245 true negatives, 164 false positives, and 149 false negatives. This resulted in 98.0% precision, 97.8% recall, and an F1 score of 0.979. The F1 score is the harmonic mean of precision and recall, and having a score close to 1 affirms good model performance.

While removing rows with missing text did not seem to make the most difference, it would be likely that this would have a much larger effect if there was more missing text data. For

example, if the data contains many rows of just the title, which contains only a few words, and the model is trained on these limited examples, it may overly rely on those words as being disproportionately important. This happens because the model has limited context to learn from and thus assigns higher weights to the few available words, even if they are not truly predictive. Therefore, even if it lowers model accuracy, it is still a good step to include in order to avoid possible bias in the vocabulary and to allow for more optimized use on different datasets.

Although the accuracy of the initial model was high, one improvement that the group thought may enhance performance further was the removal of “stopwords”. These are common English words such as “like”, “as”, or “the”. These words do not tend to add much value to the model’s classification and end up increasing the length of the input, so they can safely be removed. This also helps to reduce the amount of data that the model needs to process during the predictions. So, another iteration of the model was created. However, not all stopwords were removed from the dataset. Certain words such as 'no', 'never', 'not', 'nor', 'neither', 'but', 'therefore', 'why', 'how', 'always', 'could', and 'would' were kept. These words were kept because they carry important contextual meaning and the inclusion of these words can improve the model's ability to distinguish between real and fake news. When stopwords were removed from the model, the model's accuracy only slightly decreased from 97.8% to 97.5%, indicating that the removal of stopwords had a minimal impact on performance.

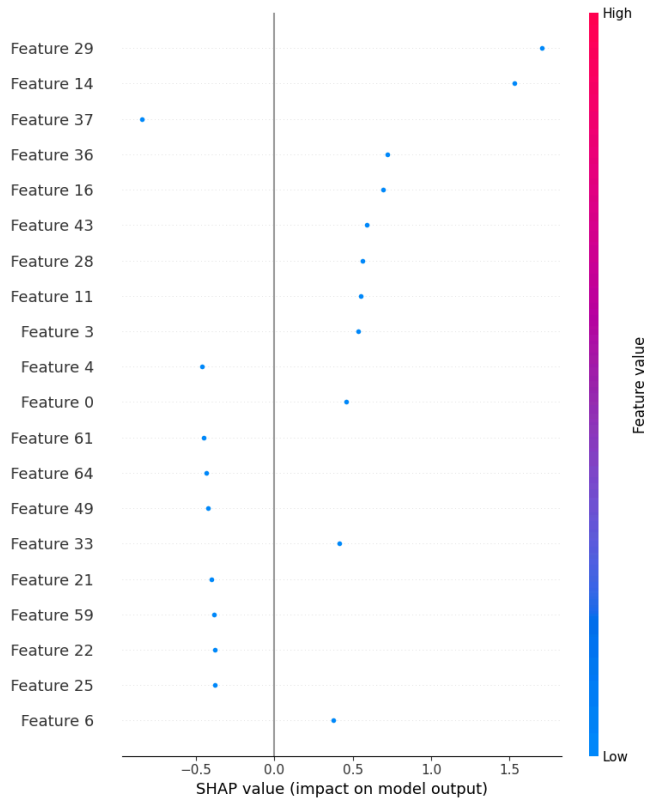
While having a model that can classify something with high accuracy is valuable, it is also incredibly important to understand how it is able to do so. Conducting some exploratory data analysis can aid in interpreting and recognizing patterns, in this case, between what would be deemed real news and fake news. One consideration in determining the validity of a piece of text could be the number of spelling or grammatical errors. For simplicity, only English content

was evaluated. This was determined using the Python package languagedetect. Out of all of the rows, only 549 entries were not in English, so they were excluded from the grammatical analysis. Of the 71,546 English rows, a small subset of the data prelabeled real and a small subset of fake were iterated through to count the number of entries with at least one spelling or grammatical error. To do this, a Python wrapper of LanguageTool was used to detect grammar or spelling errors. This tool categorizes the issues found from typos to grammar, punctuation, and more.

First, out of 500 rows labeled fake, the full combination of the title and article was iterated through. 498 of those rows had spelling or grammatical issues, and 9,931 were found overall. The top three types of errors detected were typos (4,240), typography (3,910), and punctuation (1,098), with an average of about 20 errors per combined title and article. In 500 rows labeled real, the full combination of the title and article was also iterated through. Of those 500, 482 rows had spelling or grammatical issues, and 12,112 issues were found overall. The top three types of errors detected were typography (6,554), typos (3,871), and punctuation (562), with an average of about 24 errors per entry. Although these results seem counterintuitive, as one would expect fake news to have more spelling or grammatical errors, it is important to note that 99.6% of the subsetted fake rows had at least one issue present, while only 96.4% of the true rows did. It is an interesting fact that both true and fake data subsets had the same top three issue types, those being typographical and punctuation errors.

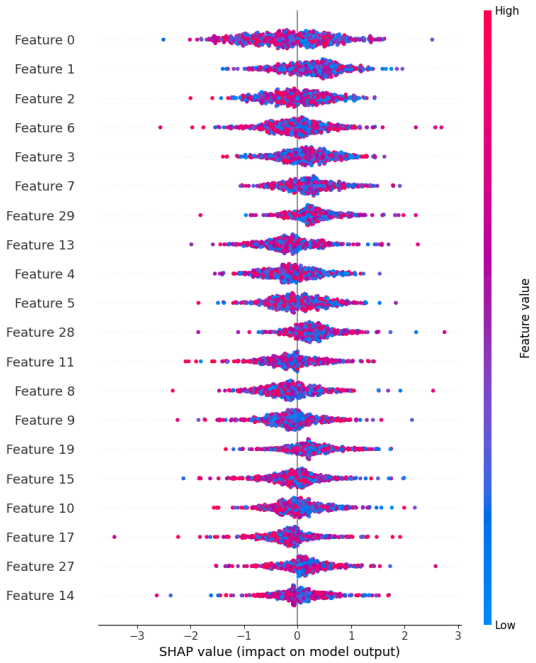
To further understand how the model makes decisions, the SHAP (SHapley Additive exPlanations) technique was applied. SHAP is a method that helps explain the output of machine learning models by attributing each feature's contribution to the final prediction. It provides insights into how individual features influence the model's prediction, offering a clearer understanding of model behavior.

SHAP analysis was used on the first data point using this text "*Elon Musk's Tesla stock up $2 billion since joining Trump's team. Although Tesla CEO Elon Musk shocked Silicon Valley by breaking ranks to become an official member of the White House manufacturing jobs initiative, the value of his Tesla stock is up by over $2 billion since the election of Donald Trump as president."* The model emphasizes the words **official(Feature 29)'**, **'initiative(Feature 37)'**, **'jobs(Feature 36)'**, and **'Tesla(Feature 16)'.** These terms were key contributors to the prediction. Words like 'official' and 'initiative' suggest government-related actions, while 'jobs' and 'Tesla' highlight the business and economic impact of Musk's role in the Trump administration, aligning with the increase in Tesla's stock.



Additionally, words like 'official', 'jobs', and 'Tesla' were associated with a real news (positive SHAP value) prediction. These words conveyed credibility through ties to verified information, economic relevance, and real-world entities. In contrast, 'initiative' was associated

with a fake news prediction (negative SHAP value), possibly due to its association with speculative claims about policies or plans which can invoke skepticism in certain narratives.



SHAP was applied to a larger dataset using K-means (K=10) to summarize the background data. The results were visualized and the features deemed important were: **['<UNK>', 'photochemical', 'consequences.last', 'процентов', 'rant.burrell', 'forever.', 'barbarism.', 'подготовили', 'dismal.', 'ground.stripling', 'bwah-ha-ha', 'pissed.featured', 'damascus']**.

Conclusion

In the effort to determine if a piece of news-related text was real or fake, the team was able to find success. Prior to development, the group researched similar projects to take note of ways to perform natural language processing. From there, each member of the team developed their own initial and different version of a classification model. In the end, the team decided to continue working with the BiLSTM model and further tuned hyperparameters. Hyperparameters within the model were edited, naturally, but manipulating the input data was also considered.

Dropping rows that were missing the article content and removing unnecessary “fluff” words from the text was done in an effort to improve the accuracy of the model. After each iteration, the model’s accuracy hovered right around 97%, with the data manipulation not affecting the final accuracy much. In addition to the different iterations of the model, some exploratory data analysis was done. The SHAP technique was used to determine what features play the largest role in the final classification, while the difference in spelling/grammatical errors present in subsets of real and fake data were also looked into. Although neither of these explorations yielded intuitive explanations when it comes to classifying a text as real or fake, it speaks to the “black box” nature of these Deep Neural Networks. To improve upon this iteration, more data could be collected from other sources to diversify the input further. Ultimately, attempting to determine whether news is real or fake is extremely important, and it seems that using deep learning techniques can garner high-quality results.

References

Agrawal, A. (2017, September 29). *Loss Functions and Optimization Algorithms. Demystified.* Medium. https://medium.com/data-science-group-iitr/loss-functions-and-optimization-algorithms-demystified-bb92daff331c

GeeksforGeeks. (2023, June 8). *Bidirectional LSTM in NLP* https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/

Holdsworth, J., & Scapicchio, M. (2024, June 17). *What is Deep Learning?*. IBM. https://www.ibm.com/topics/deep-learning

Huang, L. (2022, October). *Deep Learning for Fake News Detection: Theories and Models.* In 2022 6th International Conference on Electronic Information Technology and Computer Engineering *(EITCE 2022), October 21-23, 2022, Xiamen, China*. ACM, New York, NY, USA, https://doi.org/10.1145/3573428.3573663

Mishra, A., & Sadia, H. (2023, October 4–5). *A Comprehensive Analysis of Fake News Detection Models: A Systematic Literature Review and Current Challenges*. Presented at the International Conference on Recent Advances on Science and Engineering (RAiSE-2023), Dubai, United Arab Emirates. Eng. Proc., 59(1), 28. https://doi.org/10.3390/engproc2023059028

New York Institute of Technology. (2023, December 12). *Neural Networks 101: Understanding the Basics of This Key AI Technology*. New York Institute of Technology. https://online.nyit.edu/blog/neural-networks-101-understanding-the-basics-of-key-ai-technology

Padalko, H., Chomko, V., & Chumachenko, D. (2023, December 18). *A Novel Approach to Fake News Classification Using LSTM-based Deep Learning Models*. Frontiers. https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2023.1320800/full

Sastrawan, I. K., Bayupati, I. P. A., & Arsa, D. M. S. (2022, September 5). *Detection of Fake News Using Deep Learning CNN–RNN Based Methods*. ScienceDirect. https://www.sciencedirect.com/science/article/pii/S2405959521001375

Shahane, S. (2023, October 8). *Fake news classification*. Kaggle. https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification