JOURNAL OF
EMERGING INVESTIGATORS

Article

An explainable model for content moderation

Steven Cao[1], Matt Johnson[1]
[1] Gretchen Whitney High School, Cerritos, California

SUMMARY

The spread of fake news on social media has eroded trust in traditional news outlets and institutions. In response, social media platforms have incorporated machine learning, algorithms that can learn patterns in data without explicit programming, into their content moderation systems to help remove fake posts. However, these algorithms often misinterpret language and make poor moderation decisions. We need to better understand how machine learning algorithms interpret language. Explainability is a challenge with neural networks, a popular machine-learning model inspired by how neurons communicate in the brain. In this study, we sought to develop an explainable model for content moderation comparable in accuracy to a traditional neural network, focusing on classifying real and fake news articles. We hypothesized that by identifying keywords and quantifying their contribution to an article's credibility, we could predict credibility by summing the contributions of all the keywords in an article. We trained a convolutional neural network to classify articles using a 3,000-keyword vocabulary, achieving 85.8% accuracy. We used the Shapley additive explanations algorithm to calculate each keyword's median contribution to the model's predictions. We created a linear model that summed the median contributions of keywords in an article, achieving a comparable 81.0% accuracy. We then examined keywords with the largest median contributions. Clickbait and COVID-19 terms correlated with fake news. Legal and political terminology correlated with real news. Our results demonstrate the potential for explainable models to improve our understanding of content moderation algorithms and fake news linguistics.

INTRODUCTION

The popularity of social media enables information sharing at an unprecedented speed and scale. Half of Americans today regularly get their news from social media (1). However, social media's wide reach has made spreading false or misleading information as factual news easier. Fake news has surged as political and social divisions deepen in the United States (2). This disinformation has lowered trust in mainstream media to an all-time low, with just 20% of Democrats and 8% of Republicans having high confidence (3). Fake news has also damaged the credibility of public institutions. Notably, false stories of election fraud have interfered with the integrity of the 2020 U.S. presidential election and continue to be shared by tens of thousands of people on social media (4).

Social media platforms have faced growing scrutiny over the spread of fake news. In March 2021, Congress called Google, Facebook, and Twitter to a hearing on handling disinformation (5). The companies cited the problem as a lack of content moderation. Human moderators struggle to keep pace with the sheer size of social media platforms, where hundreds of millions of posts are created daily (6). Social media companies turned to machine learning as their solution. They promoted their new algorithms as capable of proactively removing false information and scaling the work of their human moderators across the entire platform (7). However, these state-of-the-art algorithms are far from perfect, often misinterpreting the context of posts (8). In one widely-reported example, the YouTube algorithm blocked chess videos as hate speech because it confused the game's terminology of "white" and "black" pieces "attacking" and "defending" each other with racial violence (9). Such mistakes only inflame the fake news problem by lowering users' trust in content moderation algorithms. Users become more likely to distrust the algorithm's decisions even when it correctly identifies fake news.

Social media companies have taken steps to improve their content moderation systems. For instance, machine learning models usually train on existing human-cultivated datasets, limiting their information past a certain date. This can be problematic for fact-checking current events. Facebook recently introduced Reinforcement Integrity Optimizer (RIO), which updates its content moderation algorithm with recently reviewed content, helping to close this gap (10). However, these companies have not significantly addressed the explainability of their models. By better understanding how machine learning models interpret language, we can identify possible biases in their decision-making that we can address in future model versions (11). We can achieve explainability relatively easily for simple machine-learning models. For instance, we can represent the output of a linear regression model as a linear equation where the variables are the input features. However, the deep learning models used in content moderation pose a problem. Deep learning models use neural networks, algorithms composed of computational units called nodes connected in layers (12). Through a technique known as backpropagation, the network can adjust the weights of its nodes to improve its performance, learning patterns in data. The output of a large network is the result of many connections across millions or billions of nodes. Neural networks can achieve state-of-the-art performance on language tasks, but it is far more challenging to identify how a single feature influences the network's output (13).

In this study, we hypothesized that by identifying keywords and quantifying their contribution to a news article's credibility, we could predict credibility by summing the contributions of all

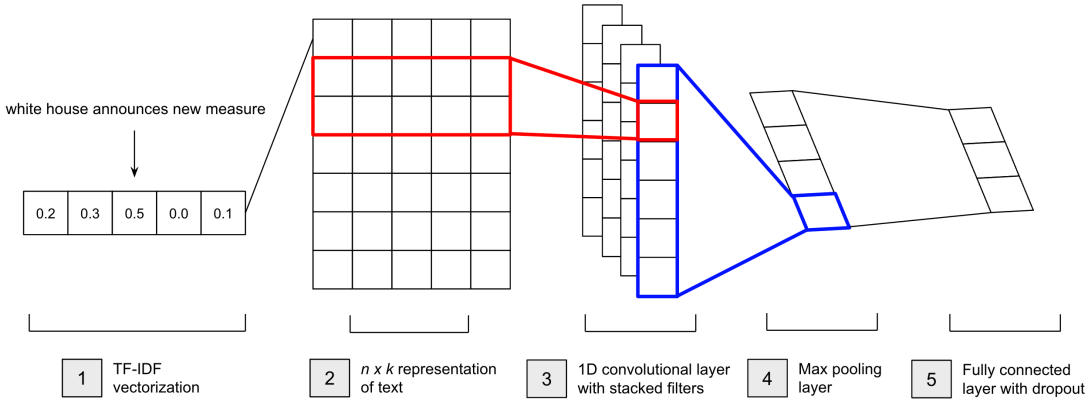

Figure 1: CNN architecture. The diagram shows the layers of the CNN in order. [1] The input text is vectorized using TF-IDF. **[2]** The vectors are represented in the shape (200000, 3000) as there are 200,000 news articles, each with 3,000 keywords. **[3]** The data is then processed through 16 stacked convolutional layers. **[4]** The maximum value of each convolutional layer is taken by the pooling layer. **[5]** The output of the pooling layer is flattened and processed through a fully connected layer of 16 nodes and a final sigmoid layer, with a dropout of 0.5 to reduce overfitting.

keywords in an article. We developed an explainable model with comparable accuracy to a traditional neural network in classifying real and fake news. This demonstrates the viability of explainable models to approximate and therefore explain content moderation algorithms. We then examined the keywords with the largest influence on our explainable model. Clickbait and COVID-19 keywords correlated with fake news, while legal and political terminology correlated with real news. These trends highlight how explainable models can improve our understanding of fake news linguistics.

RESULTS

Data collection and preprocessing

We trained our machine learning model to classify real and fake news articles using the News Landscape Ground Truth (NELA-GT) dataset. NELA-GT is composed of 1.8 million news articles published in 2021 from 367 sources (14). Each source has a credibility score from the fact-checking organization Media Bias Fact Check (MBFC). MBFC rated sources based on their history of political bias, the factuality of information, and credible sourcing (15). We labeled news articles from low-credibility sources as fake news and articles from high-credibility sources as real news, creating a balanced dataset of 100,000 real and 100,000 fake news articles. Since we labeled credibility by source, we also wanted to verify that our model was not simply memorizing source names and was generalizable to all news articles. We utilized a second dataset: Information and Security and Object Technology (ISOT), a smaller corpus of 40,000 news articles labeled as real or fake by human fact-checkers from Politifact (16). ISOT contained different sources from NELA-GT.

We then preprocessed the datasets to increase efficiency for fake news classification. We removed linguistic elements of the text that our study did not focus on: punctuation, capitalization, word stemming, and stop words (common words that provide little information, such as the articles "a," "an," and "the"). We then used Term Frequency Inverse Document Frequency (TF-IDF) vectorization to transform the text into numerical values that our machine-learning model could process (17). We limited the vectorization to a 3,000-keyword vocabulary to only retain the most important keywords.

Training and testing the convolutional neural network

We split NELA-GT into an 80:20 ratio of training to test data. We trained a convolutional neural network (CNN) to classify articles in the NELA-GT training dataset as real or fake. The CNN utilized a standard architecture of convolutional, pooling, and fully connected layers (**Figure 1**) (18). In training, the CNN converged at a validation accuracy of 86% after 20 iterations, at which we halted training (**Figure 2**). The training and validation accuracy curves stayed relatively similar, so the CNN had minimum overfitting (**Figure 2**). We then tested the CNN on the NELA-GT testing dataset for 85.8% accuracy and the ISOT dataset for 79.4% accuracy (**Table 1**). The CNN achieved a high accuracy on both datasets,

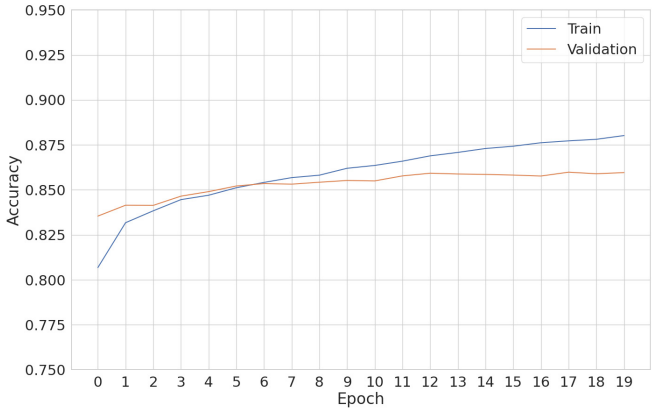

Figure 2: CNN training and validation curves. Training and validation accuracy of CNN over 20 epochs (iterations) on the NELA-GT training dataset. The training curve (blue) evaluates the CNN's performance on the training dataset. The validation curve (orange) evaluates CNN's performance on new data it was not trained on. The training curve rises to almost 88%, while validation accuracy plateaus converges at 86%.

	NELA-GT Accuracy	ISOT Accuracy
CNN	0.858	0.794
Explainable Model	0.810	0.801

Table 1: CNN and explainable model accuracies. Accuracy of CNN and explainable model on classifying real and fake news articles in NELA-GT and ISOT datasets.

based on industry standards which generally required a 70% accuracy and performance by other fake news classification neural networks (19-21). Therefore, we could confidently quantify how individual keywords influenced the CNN and, by extension, news credibility.

Determining keyword contributions for the CNN

We used the Shapley additive explanations (SHAP) algorithm to explain the CNN's prediction as the sum of the contributions of the keywords in an article (22). For a news article, SHAP calculated each keyword's Shapley value, representing its contribution to the article's credibility (23). A positive value indicated that the keyword increased the article's credibility, while a negative value indicated that the keyword decreased credibility. By calculating the Shapley values for all 3,000 keywords across a sample of 10,000 news articles, we determined each keyword's median Shapley value. We used this value to estimate each keyword's median contribution to credibility.

Creating a linear model for explanation

We created a linear model that summed the median Shapley values of all the keywords in a news article. A positive sum predicted real news, and a negative sum predicted fake news. Our linear model provided simple explainability as we could directly quantify each keyword's contribution to credibility as its median Shapley value. We tested our linear model on our two datasets, achieving 81.0% accuracy on NELA-GT and 80.1% accuracy on ISOT. Given the comparable accuracy to the CNN, we could confidently examine the keywords with the largest contributions to identify possible keyword trends in the explainable model. Since the explainable model approximated the CNN's predictions, these keyword trends would also reflect the CNN's behavior.

DISCUSSION

We hypothesized that by identifying keywords and quantifying their contribution to an article's credibility, we could predict credibility by summing the contributions of all the keywords in an article. Based on our results, our hypothesis was supported. Our explainable linear model achieved an accuracy of 81.0% on NELA-GT and 80.1% on ISOT, comparable to the CNN's accuracy of 85.8% on NELA-GT and 79.4% on ISOT (**Table 1**).

We quantified each keyword's median contribution to credibility as its median Shapley value. We determined the median to be a better measure than the mean because of extreme outliers in Shapley value distributions (**Figure 3**). All median Shapley values were relatively small values (**Figure 4**). We considered this to be reasonable as it is unlikely for a single keyword to determine the credibility of an article. Rather, the explainable model needed many keywords leaning in one

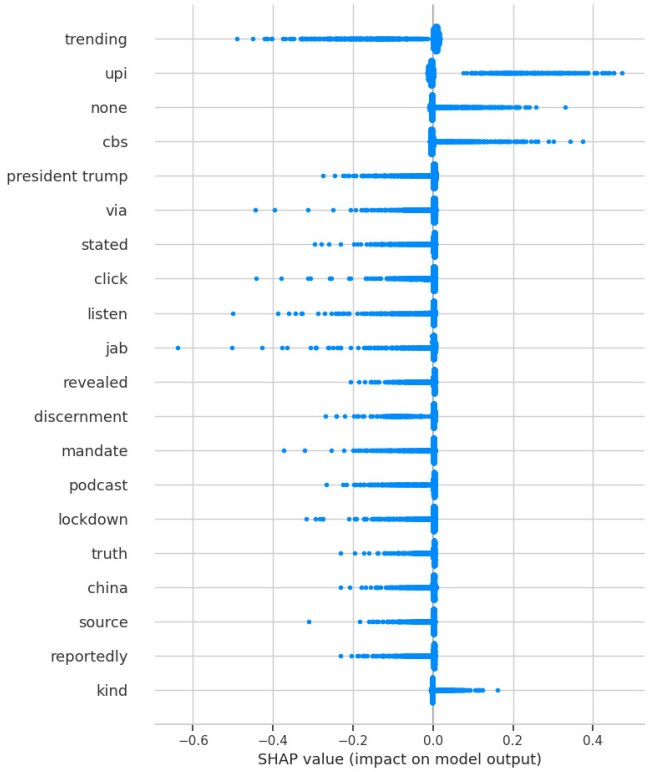

Figure 3: Shapley values for the 20 keywords with the largest influence on CNN. The 20 keywords displayed have the largest median Shapley values when all keywords are ordered by magnitude. For each keyword, its distribution of Shapley values is from 10,000 articles in the NELA-GT dataset. Each dot represents the Shapley value of a keyword in one article. Negative values (left) are associated with fake news, and positive values (right) are associated with real news.

direction to confidently classify an article as real or fake. We examined the 20 keywords with the largest median Shapley values for real and fake news. Their distribution of Shapley values was skewed to one side, indicating a consistent positive or negative contribution to credibility (**Figure 4**). We categorized the keywords into groups based on similar meanings, references, or intentions. We did not include keywords that described source names because they would not be generalizable trends. Of the fake news keywords, we identified the clickbait terms "trending," "click," "revealed," "listen," and "donate," suggesting that clickbait is a common tactic used to capture the attention of potential readers (**Table 2**). There were also references to COVID-19 in the keywords "jab," "lockdown," "mandate," "china," and "fauci" (referring to Dr. Anthony Fauci, a chief US medical advisor during the pandemic) (**Table 2**). Misinformation surrounding COVID-19 has been a frequent topic for fake news. Words such as "via" and "reportedly" also hinted at less credible sources of information (**Table 2**). In contrast, the real news keywords included legal copyright terminology "right reserved material," "rewritten redistributed," and "press right reserved," which may indicate more professional standards used by credible news organizations (**Table 2**). We also identified political terms such as "capitol," "minister," "prosecutor," "senate," and "lawmaker," suggesting that politics is a common topic in real news but addressed in more formal language (**Table**

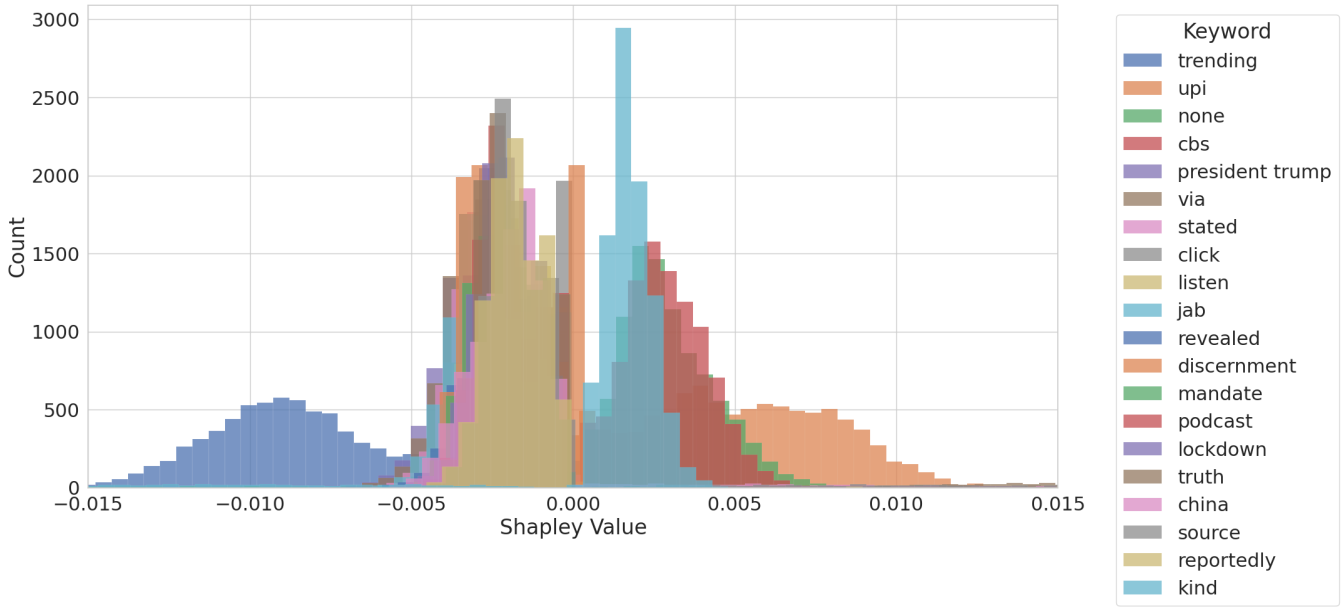

Figure 4: Shapley values for the 20 keywords with the largest influence on CNN. The 20 keywords displayed have the largest median Shapley values when all keywords are ordered by magnitude. For each keyword, its distribution of Shapley values is from 10,000 articles in the NELA-GT dataset. The distribution of Shapley values for most keywords is unimodal and with a slight skew. Two outlying distributions are "trending" on the left and "upi" on the right. Almost all Shapley values fall between -0.015 and 0.015 (extreme outliers not shown).

2). Words such as "updated" and "said statement" may also demonstrate the accountability present in real news (**Table 2**). However, we must emphasize that we have not determined these correlations to be causal. Rather, these are a series of observations from our explainable model. We recommend further analysis of these correlations in future work to determine their strength.

Our study required several important design considerations. When collecting data, we needed to determine what qualified as real and fake news. An early limitation of our study was the quality of existing fake news datasets created by human-fact checkers (16, 24, 25). Since the data authors had to check each article individually, these datasets contained a relatively small number of articles from a few sources. We were concerned that training our machine learning model on such data would be problematic. The model would likely learn the writing styles of these few specific sources to achieve the highest accuracy rather than more general linguistic trends in real and fake news. We took an alternative approach, determining real and fake news based on the source's credibility. We argued that we could reasonably ascertain the credibility of an article by examining the source's track record. We could also increase the confidence in our approach by using sources at the extreme ends of credibility, either very credible or uncredible. We conceded that we might introduce some inconsistency since we could not know whether every article from a credible source was real and every article from an uncredible source was fake. However, our approach enabled us to create a larger and more diverse dataset, as not every article had to be fact-checked by a human. With NELA-GT, we could train our model on 200,000 articles from 154 very credible or uncredible sources. We could then use a smaller existing fake news dataset, ISOT, to test the generalizability of our model.

We also considered the possible biases created by preprocessing the text. First, we may have removed linguistic features that provided rhetorical context to the text, such as exclamation marks or fully capitalized words. However, we determined this not to be a significant issue since our study focused on individual keywords. Second, TF-IDF vectorization tries to capture a word's distinctiveness in the text based on its frequency, which can result in two extremes. If a word was obscure and only appeared in a few fake news articles, the CNN may directly associate it with fake news regardless of its meaning. If a word was too common, its frequency difference in real and fake news could simply result from random chance, and the CNN could create an artificial association with credibility. Due to these potential issues, we restricted TF-IDF vectorization only to include keywords with a frequency between 1-10% of all articles.

One major limitation of our study was the computational power of our hardware. We utilized convolutional neural networks in favor of more text-specific deep learning models, such as recurrent neural networks and long short-term memory networks. CNNs offer much greater efficiency in comparison, which was a crucial requirement given the large size of our datasets (18). Additionally, we would have ideally calculated the Shapley values for all 200,000 articles for the best approximation of the CNN using our linear model. However, the number of calculations SHAP requires grows exponentially with data size, and our moderate 10,000-article sample took more than 2 hours to calculate.

The results of this study demonstrate that explainable content moderation models can achieve comparable accuracy to deep learning models. The benefits are twofold. First, this demonstrates that explainable models can approximate the original model and explain its functionality. While explainable models may not replace deep learning models as content moderation algorithms, they can be helpful diagnostic tools. If the explainable model shows an unusual correlation between

Keyword	Median Shapley value	Keyword	Median Shapley value
trending	-2.87E-03	upi	4.80E-03
president trump	-2.84E-03	cbs	2.74E-03
via	-2.68E-03	none	2.60E-03
jab	-2.52E-03	kind	1.59E-03
stated	-2.51E-03	cbs news	1.53E-03
discernment	-2.44E-03	associated press	1.42E-03
click	-2.30E-03	updated	1.41E-03
revealed	-2.17E-03	capitol	1.37E-03
truth	-2.13E-03	minister	1.35E-03
lockdown	-2.08E-03	said statement	1.34E-03
mandate	-2.07E-03	right reserved material	1.31E-03
podcast	-1.96E-03	press right reserved	1.31E-03
listen	-1.79E-03	rewritten redistributed	1.26E-03
reportedly	-1.79E-03	writer	1.18E-03
donate	-1.78E-03	prosecutor	1.12E-03
source	-1.77E-03	authority	1.11E-03
illegal	-1.76E-03	democratic	1.06E-03
freedom	-1.75E-03	late	1.05E-03
china	-1.66E-03	senate	1.05E-03
fauci	-1.65E-03	lawmaker	9.86E-04

Table 2: Median Shapley values of the 40 keywords with the largest influence on CNN. The 20 keywords with the most positive median Shapley values and the 20 keywords with the most negative median Shapley values. Positive Shapley values are associated with real news, and negative Shapley values are associated with fake news.

a keyword and credibility, we can test the deep learning model for possible biases. Second, the keyword trends that explainable models identify can improve our understanding of fake news linguistics. We can further examine these trends to determine their strength and potentially incorporate them into future content moderation models as "red flag" words to better identify fake news. Given these potential benefits, we recommend further research on leveraging the keyword trends provided by explainable models to improve content moderation systems.

MATERIALS AND METHODS

Datasets

This study utilized two datasets. NELA-GT is a large dataset of 1.8 million news articles published in 2021 from 367 sources (14). Each article consisted of the text, source name, and credibility score. The credibility score ranged from 0 (very low credibility) to 5 (very high credibility) and is based on research by the fact-checking organization Media Bias / Fact Check (MBFC). MBFC rated sources based on their history of political bias, information factuality, and sourcing credibility. Although their credibility scores could not be completely objective, a *Scientific Reports* study found MBFC suitable for most scientific studies (26). Researchers at the University of Michigan also utilized MBFC to examine disinformation in social media (27). For our study, we categorized sources with credibility scores of 0-1 as fake news, such as Infowars, which commonly promoted conspiracy and pseudo-science. We categorized sources with scores of 4-5 as real news, such as the Associated Press, with a long track record of factual content at high journalistic standards. We excluded sources with scores of 2-3 because of their mixed credibility. We created a balanced dataset of 100,000 real and 100,000 fake news articles from NELA-GT. We also used a second smaller dataset, ISOT, composed of 20,000 real and 20,000 fake news articles labeled by human fact-checkers from Politifact (16).

Data preprocessing

We preprocessed both datasets to reduce their complexity while focusing on the goal of text classification. We first removed linguistic elements of the text that did not affect our study: punctuation, capitalization, and word stemming. We then removed stop words, commonly used words that carry little contextual information, such as the articles "a," "an," and "the." This left a string of important keywords that carried the meaning of the text. The machine learning model can treat each unique keyword as a single feature in the input.

We then vectorized the datasets to transform the text into values the machine learning model could process. This study used TF-IDF vectorization, transforming keywords into numerical weights based on their frequency in the text. TF-IDF is a popular technique for text classification tasks because it captures the distinctiveness of a keyword by comparing its term frequency (number of appearances in the current document) with its inverse document frequency (number of appearances in all documents) (17). The algorithm is as follows:

$$tf(w,d) = log(1 + f(w,d))$$
$$idf(w,D) = log(\frac{n}{f(w,D)})$$
$$tfidf(w,d,D) = tf(w,d) \times idf(w,D)$$

Where f(w, d) is the term frequency of the word w in the document d and f(w, D) is the inverse document frequency of the word w in the total collection of documents D.

By comparing the distinctiveness of a keyword in different articles, the machine learning model could learn how present it is real or fake news. We also only wanted to retain the most important keywords, so we limited TF-IDF to a maximum vocabulary size of 3,000.

Training and testing the CNN

We used a standard convolutional neural network as our deep learning model. The CNN consisted of three main layers: a convolutional layer, a pooling layer, and a fully connected layer (**Figure 1**) (18). The convolutional layer applied filters to the input data to extract relevant features. The pooling layer then reduced the dimensionality of the data by taking the maximum value across subregions. The data was flattened and fed into the fully connected layer to produce the final classification. We also included a dropout layer to reduce overfitting. We split the NELA-GT dataset into an 80:20 training and testing data ratio. We trained the CNN on the training data for 20 epochs (iterations) and stopped because the validation accuracy had converged around 86% (**Figure 2**). We then tested the CNN on the NELA-GT test data for 85.8% accuracy and on the ISOT dataset for 79.4% accuracy.

Using Shapley values to explain the CNN

To explain our CNN, we utilized the game theory concept of Shapley values. The Shapley value is the average marginal contribution of a player in a team (22). Consider a team *N* of *n* players that cooperated to finish a task. The contribution an individual player *i* made is the difference in the value of the team with and without *i*, known as *i*'s marginal contribution.

The Shapley value of a player *i* is its mean marginal contribution across all possible subsets of the team. This can be expressed mathematically as:

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} (v(S \cup \{i\}) - v(S))$$

Where the function v gives the value the contribution provided by a subset of S players.

Shapley additive explanations (SHAP) is an algorithm that explains the predictions of a machine learning model as the sum of its features (23). In the context of this study, we can consider the keywords in an article like players on a team. The value of this team is the credibility of the article. For a specific article, SHAP assigns each keyword a Shapley value representing its contribution to the article's credibility. A positive Shapley value indicates that the keyword increases the article's credibility, while a negative article indicates a decrease in credibility. To calculate these Shapley values, SHAP uses a background dataset of articles for reference. By comparing how keyword differences between an article and those in the background dataset affect the CNN's prediction, SHAP calculates how individual keywords affect the article's credibility. Our study used a variation of the SHAP known as DeepSHAP, which leverages the internal structure of neural networks to explain them more efficiently (23). We calculated the Shapley values for all 3,000 keywords for each article in a 10,000-article sample with a 5,000-article background dataset. To find each keyword's median contribution, we calculated its median Shapley value across the entire 10,000 article sample.

Achieving explainability with a linear model

We then created a linear model that classified real and fake articles by summing the median contributions of the keywords found in an article. A positive sum predicted real news, and a negative sum predicted fake news. A linear model provided clear explainability by allowing us to examine the keyword contributions directly. For a text input, our linear model transformed it into a list *x* of 0s and 1s representing whether each of the 3,000 keywords is present. Each value in the list is then multiplied by the median contribution of its corresponding keyword Φ and summed. Our linear model can be represented mathematically as:

$$f(x) = \sum_{i=1}^{3000} \Phi_i x_i$$

Where $x \in \{0,1\}^{3000}$

We tested the linear model on our two datasets, achieving 81.0% on NELA-GT and 80.1% on ISOT (**Table 2**). Given the high accuracy, we could confidently use the keyword contribution Φ as a measure of the keyword *i*'s influence on credibility.

Received: December 16, 2022
Accepted: April 24, 2023
Published: August 16, 2023

REFERENCES

1. Walker, Mason. "News Consumption across Social Media in 2021." *Pew Research Center*, 20 Sep. 2021.
2. Lazer, David M. J., et al. "The science of fake news". *Science*, vol. 359, no. 6380, 9 Mar. 2018, pp. 1094–1096, https://doi.org/10.1126/science.aao2998.
3. Brenan, Megan. "Media Confidence Ratings at Record Lows." *Gallup*, 5 Jun. 2023.
4. Miller, Greg. "As U.S. election nears, researchers are following the trail of fake news." *Science*, 26 Oct. 2020.
5. Wakefield, Jane. "Google, Facebook, Twitter grilled in US on fake news." *BBC News.* 25 Mar. 2021.
6. Dormehl, Luke. "Humans Can't Stop Online Hate Speech Alone. We Need Bots to Help." *Digital Trends*. 16 Jul. 2020.
7. Seetharaman, Deepa, et al. "Facebook Says AI Will Clean Up the Platform. Its Own Engineers Have Doubts." *Wall Street Journal.* 17 Oct. 2021.
8. Simonite, Tom. "Facebook Is Everywhere; Its Moderation Is Nowhere Close". *Wired*. 25 Oct. 2021.
9. Knight, Will. "Why a YouTube Chat About Chess Got Flagged for Hate Speech". *Wired*. 1 Mar. 2021.
10. Condon, Stephanie. "Facebook shares AI advancements improving content moderation." *ZDNET*. 18 Aug. 2021.
11. Savage, Neil. "Breaking into the black box of artificial intelligence". *Nature*. 29 Mar. 2022.
12. "What are Neural Networks?" *IBM*. Accessed 15 Jun. 2023.
13. Castelvecchi, Davide. "Can we open the black box of AI?." *Nature News* 538.7623. 2016.
14. Gruppi, Maurício, et al. "NELA-GT-2022: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles." *arXiv*, 2022, https://doi.org/10.48550/arXiv.2203.05659.
15. "Methodology." *Media Bias/Fact Check*. 11 May 2023.
16. Ahmed, H, et al. "Detecting opinion spams and fake news using text classification." *Journal of Security and Privacy*. Volume 1, Issue 1. Jan 2018.
17. Ramos, Juan. "Using tf-idf to determine word relevance in document queries." *Proceedings of the first instructional conference on machine learning*. Vol. 242. No. 1. Dec 2003.
18. Zhang, Ye and Byron, Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification." *arXiv*, 2015, https://doi.org/10.48550/arXiv.1510.03820.
19. "Which is more important: model performance or model accuracy?" *Fiddler.* Accessed 15 Jun. 2023.
20. Reis, Julio CS, et al. "Supervised learning for fake news detection." *IEEE Intelligent Systems* 34.2 (2019): 76-81.
21. Pérez-Rosas, Verónica, et al. "Automatic detection of fake news." *arXiv*, 2017, https://doi.org/10.48550/arXiv.1708.07104.
22. Roth, Alvin E. "Introduction to the Shapley Value." *The Shapley Value*, 1988, pp. 1–28., https://doi.org/10.1017/cbo9780511528446.002.
23. Lundberg, Scott M. and Su-In, Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).
24. Wang, William. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection." *arXiv*, 2017, https://doi.org/10.48550/arXiv.1705.00648.
25. Shu, Kai, et al. "FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media."

arXiv, 2018, https://doi.org/10.48550/arXiv.1809.01286.
26. Choloniewski, Jan, et al. "A Calibrated Measure to Compare Fluctuations of Different Entities across Timescales." *Scientific Reports*, vol. 10, no. 1, 2020, https://doi.org/10.1038/s41598-020-77660-4.
27. Resnick, Paul, et al. "Iffy quotient: A platform health metric for misinformation." *Cent Soc Media Responsib* 17 (2018): 1-20.

APPENDIX

Text preprocessing

```
# lowercase text
nela['text'] = nela['text'].apply(lambda x: x.lower())

# remove \n
nela['text'] = nela['text'].str.replace(r'\n', ' ')

# remove numbers
nela['text'] = nela['text'].str.replace('\d+', ' ')

# remove puncutation
nela['text'] = nela['text'].str.replace(r'[^\w\s]+', ' ')

# remove stopwords
stopwords = nltk.corpus.stopwords.words('english')
nela['text'] = nela['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stopwords)]))

# remove 1-2 letter words
nela['text'] = nela['text'].apply(lambda x: ' '.join(word for word in x.split() if len(word)>2))

# remove word stemming
lemmatizer = nltk.stem.WordNetLemmatizer()
nela['text'] = nela['text'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for word in x.split()]))
```

Data train-test split and vectorization

```
# split NELA-GT into train and test datasets
x = nela['text']
y = nela['credibility']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# text vectorization
vec = TfidfVectorizer(min_df=0.01, max_df=0.1, ngram_range=(1,3), max_features=3000)
xv_train = vec.fit_transform(x_train)
xv_test = vec.transform(x_test)
```

CNN Model

```
# create CNN
input_dim = xv_train.shape[1]
model = Sequential([
    layers.Conv1D(16, 3, activation='relu', input_shape=(input_dim, 1)),
    layers.MaxPooling1D(2),
    layers.Flatten(),
    layers.Dense(16, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(1, activation='sigmoid')])
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.summary()

# train CNN on NELA-GT train dataset
checkpoint = ModelCheckpoint('cnn.hdf5', monitor='val_loss', verbose=1, save_best_only=True, mode ='auto')
callbacks_list = [checkpoint]
history = model.fit(xv_train, y_train, epochs=20, validation_data=(xv_test, y_test), callbacks = callbacks_list)

# test CNN on NELA-GT test dataset
nela_results = model.evaluate(xv_test, y_test)
```

SHAP explanation

```
# input CNN and 5,000-article background dataset into DeepSHAP
sample_x_train = xv_train[:5000].reshape(5000, input_dim, 1)
explainer = shap.DeepExplainer(model, sample_x_train)

# calculate Shapley values for 10,000-article sample
sample_x_test = xv_test[5000:10000].reshape(10000, input_dim, 1)
shap_values = explainer.shap_values(sample_x_test)

# organize keywords by median Shapley value
feature_names = vec.get_feature_names_out()
shap_df = pd.DataFrame(shap_values, columns=feature_names)
vals = shap_df.median(axis=0)
shap_importance = pd.DataFrame(list(zip(feature_names, vals)), columns=['word', 'median_shapley_value'])
shap_importance = shap_importance.sort_values(by=['median_shapley_value'])
```

Explainable model

```
# create explainable model
def linear_model(x):
  sum_contribution = 0
  for keyword in range(len(x)):
    if x[keyword] != 0:
      sum_contribution += shap_importance['median_shapley_value'][keyword]
  if sum_contribution > 0:
    return 1
  return 0
```