Fake News Detection

Anushya Subbiah
anushya@stanford.edu

Divya Sudhakar
divyasud@stanford.edu

Kenny Hsu
shsu1@stanford.edu

Abstract

We present a comparison of models to reliably and scalably detect "fake news," defined here as articles masquerading as news containing intentional misinformation. We treat the problem as a binary classification and leverage linguistic features in the news articles and social context features such as user engagement with the news articles. We were able to achieve an accuracy of 84% and an F1 score of 0.89 with our best performing model using transfer learning with BERT encodings for the linguistic features fed to a DNN that also takes the other features as inputs. We also show that the features in the news article are sufficient to make a classification with high accuracy (>85%). The improvement in accuracy from incorporating the user tweets is marginal.

1 Introduction

"Fake news" was not a term many people used a few years ago, but it is now seen as one of the greatest threats to democracy and free debate. Fake news detection comes with several challenges:

- It is often written with the explicit intent to mislead/fool people for financial or political gain and so the problem is inherently adversarial.
- The topic is usually a current event that happened very recently, so there is not a lot of data/time to respond to new pieces of fake news.
- While a huge body of research on the problem of distinguishing misinformation exists, the work is fragmented over multiple problems (detecting satire, sensationalism, stance, etc.) and different datasets (tweets, content of news articles, user engagement with news articles on social media, etc).

There has been a lot of research on detecting whether or not a piece of text could be misinformation on it's own or what the stance of a piece of text is and whether or not it is satire or sensationalist. There has also been a fair amount of work on trying to categorize the credibility of tweets. But there hasn't been too much work done by combining the two datasets - news articles and tweets on the news articles. Shu et al. suggested that using tweets in combination with news articles might provide a stronger signal to determine if a given article is misinformation and provided a dataset, FakeNewsNet, that combined these two sources.[1]

We evaluated several models - some using just the news content features, some using just the user engagement features and some using the combined features from the news articles as well as the user engagement - on this dataset. In this paper, we present a comparison of all these models and an error analysis that show the limitations of these models as well as a discussion on future avenues that could be explored in this space.

2 Related Work

Shu et al. provide a survey of the Fake News detection landscape including the various public datasets, the methodologies, features and algorithms used to distinguish between real and fake news.[2]

Golbeck et al. examine the problem of classifying between fake news and satire and provide a dataset, including full text of articles, links to the original stories, rebutting articles for fake news, and thematic codes, for doing the same.[3]

Hanselowski et al. provide a retrospective on the 2017 Fake News Challenge to identify the stance of the title of an article compared to the body of the article. [4] They describe the top performing models including the model that won the challenge - a Deep Convolutional Neural Net (CNN) using pre-trained word2vec embeddings and with gradient boosted decision trees. The other models described in the paper include Multi Layer Perceptron (MLP) based models.

In [5], Wang provides a dataset of labeled statements collected from Politifact combined with metadata about the authors of the statements such as party affiliations, current job, home state, and credit history. Wang also provides a CNN based model that reads the embeddings of the statements and the metadata.

3 Dataset and Features

We used FakeNewsNet, a publicly available dataset that has been generated by integrating two labeled news datasets from fact checking websites PolitiFact (https://www.politifact.com/) and GossipCop (https://www.gossipcop.com/) with user context data obtained from Twitter [1].

	Category	Features	PolitiFact		GossipCop	
			Fake	Real	Fake	Real
News Content	*Linguistic*	# News articles	432	624	5,323	16,817
		# News articles with text	420	528	4,947	16,694
	Visual	# News articles with images	336	447	1,650	16,767
Social Context	*User*	# Users posting tweets	95,553	249,887	265,155	80,137
		# Users involved in likes	113,473	401,363	348,852	145,078
		# Users involved in retweets	106,195	346,459	239,483	118,894
		# Users involved in replies	40,585	18,6675	106,325	50,799
	Post	# Tweets posting news	164,892	399,237	519,581	876,967
	Response	# Tweets with replies	11,975	41,852	39,717	11,912
		# Tweets with likes	31692	93,839	96,906	41,889
		# Tweets with retweets	23,489	67,035	56,552	24,955
	Network	# Followers	405,509,460	1,012,218,640	630,231,413	293,001,487
		# Followees	449,463,557	1,071,492,603	619,207,586	308,428,225
		Average # followers	1299.98	982.67	1020.99	933.64
		Average # followees	1440.89	1040.21	1003.14	982.80
Spatiotemporal Information	*Spatial*	# User profiles with locations	217,379	719,331	429,547	220,264
		# Tweets with locations	3,337	12,692	12,286	2,451
	Temporal	# Timestamps for news pieces	296	167	3,558	9,119
		# Timestamps for response	171,301	669,641	381,600	200,531

Table 1. Statistics of FakeNewsNet repository

From the raw downloaded data, we created three datasets, one with only the news content, another with tweets/user data and another with both.

The news content dataset included the article title, the domain name of the website, and the text content of the news article as features. It had 22,233 articles, roughly 75% of which are real news and 25% are fake news. The user engagement dataset consisted of approximately 1,800,000 tweets, the screen name, ID and location of the user who sent the tweet, whether the user is verified, and social features such as the number of tweets, updates, friends, followers the user has. The third dataset had a combination of the news content and the user engagement data from above.

Dataset splits

We split the news content dataset described above into 20,000 training, 1,000 development (dev) and 1,233 test set examples. We split the user engagement dataset described above into 1,500,000 training, 83,500 dev, 93,348 test examples. For the combined dataset, we experimented with two approaches:

1. Treat each tweet as an individual example. This results in a split of 1,500,000 training, 83,500 dev, 93,348 test examples.
2. Aggregate all tweets of same article into one row. This merges all the tweets pertaining to a single news article as one super-tweet and creates an average tweeter for the article by dropping all the PII of the tweeters and averaging the remaining social features. This approach results in a split of 18,000 training, 1,000 dev, 1,334 test examples.

4 Methods

News Content Models

Multi Layer Perceptron

The MLP had 5 hidden layers with a few 100s of neurons each and used the title, source and text of the news articles as features. For the title and the text, we used word embeddings available in Tensorflow that map words into 128 dimension vectors using a Feed forward Neural Net. We used a ProximalAdagradOptimizer with Batch Norm and L2 regularization.

RNN with just the title

The RNN used only the title of the article as input. It used an embedding layer as the input layer and a word encoder with vocab size of 32K. The embedding layer is connected to one or two stacked recurrent layers with either 64 GRU or LSTM units which are then connected to one of two fully connected hidden layers with 64 or 32 units and relu activation. The final output layer is a single unit with sigmoid activation. We applied L2 regularization to the fully connected and/or recurrent layers and dropout to the layers after the recurrent layers and used Adam optimizer.

BERT Transfer learning

We tried transfer learning using pre-trained Bidirectional Encoder Representations from Transformers (BERT) [6] embeddings for linguistic features in the news. We unfroze the last layer

and fed that into a softmax layer to make a real/fake classification. We used the title and the text of the articles preprocessed into the format BERT was trained on as features as described in [6]. The BERT libraries used polynomial learning rate decay and an Adam optimizer. We use dropout.

User Engagement Models

Multi Layer Perceptrons

We experimented with an MLP using just the user engagement data but the accuracy remained low so we did not pursue this further.

Combined Models

Multi Layer Perceptrons with each tweet as an individual example

This model had 3-5 hidden layers, each with anywhere from 100 to 6000 units and treated each tweet and its associated news article as an individual example. Categorical features were turned into numerical values based on how frequently they appear. We used batch norm and relu activation for most layers, along with Adam optimizer.

This model was extremely slow to train using the free Google Colab resources due to the large number of examples. It is possible that we could have achieved better accuracy if we'd had more resources and/or time to allow it to converge.

Multi Layer Perceptron aggregating all tweets for the same article

We also tried another MLP after merging all the tweets pertaining to each news article into one super tweet and creating an "average user" for these tweets. This reduced the sheer number of examples to train on allowing the model to converge faster.

The model had 8 hidden layers with a few 1000 neurons in each layer. For the linguistic features - news title, text and tweets, we used the same word embeddings available in Tensorflow described earlier. We used a ProximalAdagradOptimizer with Batch Norm and L2 regularization.

BERT + DNN

We used pre-trained BERT embeddings for linguistic features - news title, text and tweets. We unfroze the last layer and fed that into a softmax layer to make a real/fake classification. We then fed the result of that classification and the remaining non-linguistic features - news source, user verified status, user followers count, user listed count, user statuses count, user favourites count - into a DNN with 6 layers containing 718 neurons each using Leaky ReLU activation and a final softmax layer.

We used cross entropy loss for all the models specified above.

5 Results

Model	Features Used	Best Accuracy	Code

MLP	News Title, Text, Source	84%	Colab Link
RNN	News Title	84%	Colab Link
BERT	News Title, Text	87%	Colab Link
MLP	User engagement Data	37% on the training set. Model did not converge.	Colab Link
MLP	All news and user engagement data with each tweet as an individual example	60%	Colab Link
MLP	All news and user engagement data	88%	Colab Link
BERT + DNN	All news and user engagement data	87%	Colab Link

Table 2. Accuracy of various models

We picked the top performing models - MLP and the BERT+DNN - to run against the test set and compared their accuracies and F1 scores. Although the MLP outperformed the BERT+DNN in accuracy in both the dev and test sets, the BERT+DNN had higher F1 scores. Given the imbalance between the real and fake news examples in our datasets, we believe the F1 score gives a more realistic picture of model quality.

Model	Dev Accuracy	Dev F1	Test Accuracy	Test F1
MLP	88%	0.76	87%	0.76
BERT + DNN	87%	0.91	84%	0.89

Table 3. Comparison of top two models

Human Baseline

We attempted the task of distinguishing between real and fake news ourselves by manually labeling approximately 360 examples in the test set. We achieved an accuracy of 79.1% and recall of 36%.

Error Analysis

Error analysis showed that the articles that all the models struggle with the most are primarily of the Celebrity Gossip variety with hollywoodlife.com, people.com, usmagazine.com being the top three sources to be mispredicted. Given that celebrity gossip articles dominate the dataset, this is unsurprising. Even with that caveat, this performance seems reasonable given that most celebrity gossip have a similar tone and sound the same and it is hard to determine which of those are fake without additional research even as a human evaluator.

The models that just used news content features also mislabeled a few examples from source en.wikipedia.org since the GossipCop dataset contains articles from Wikipedia, some of which

are labeled real and some others which are labeled fake. Again, it wasn't immediately apparent which of these are fake without additional research even as a human evaluator. But the models that incorporated user engagement features were able to correctly classify all but one of the 35 articles from Wikipedia in the dev set.

6 Conclusion and Future Work

We were able to show that deep neural net based models were able to classify real from fake news better than an average human without additional research and the models achieved good enough accuracy using just the features from the news articles. An interesting result from our experimentation that drives home this point is that the RNN with only the title feature was able to achieve a relatively high accuracy, although we think this warrants further investigation to make sure it is not penalizing the minority class given the imbalance between real and fake news articles in the dataset. Incorporating user engagement information like tweets and user data improved the accuracy of our models but we were disappointed that the increase was only marginal.

Future work that could increase the accuracy even further is to incorporate other features available in the FakeNewsNet dataset such as the images from the news article and retweets of the original user tweets. One can also look at the timing of the tweets/retweets to extract information about the characteristics of the spread of an article. It would be worthwhile to also experiment with other ways to aggregate all the tweet data associated with an article.

7 Contributions

Anushya Subbiah:

- Exploration of relevant literature and base paper.
- Baseline DNN model with news features and distillation tests with tweet features.
- Experiments with ensembling models.

Divya Sudhakar:

- Literature review and data exploration.
- Experimented with the MLP and the BERT models using just the news content.
- Experimented with the MLP with aggregated tweets and the BERT models using the combined data.
- Error analysis.

Kenny Hsu:

- Processed the raw data into the three datasets used for experimentation by extracting the features/examples mentioned above and serializing them into panda dataframes.
- Manual labeling of subset of test dataset to establish human baseline.
- Experimented with RNN models using only news title feature
- Experimented with the multi layer perceptron using each tweet as an individual row.
- Basic data exploration of the dataset.

References

[1] FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media
[2] Fake News Detection on Social Media: A Data Mining Perspective
[3] Fake News vs Satire: A Dataset and Analysis
[4] A Retrospective Analysis of the Fake News Challenge Stance Detection Task
[5] “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection
[6] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding