# Detecting Fake News with Natural Language Processing: Deep Learning and Automated Machine Learning

**Zach Merritt, Deepak Nagaraj, Michael Powers**
School of Information, University of California, Berkeley
*December 2018*

## Abstract

The goal of this project is to examine different approaches to accurately classify "fake news" using natural language processing ("NLP") methods on the text of news articles. We rely on two data sets that were generated in a previous research project by Perez-Rosas et al. from the Universities of Michigan and Amsterdam. When faced with a text classification problem, such as fake news detection, a popular approach is to extract features via NLP techniques and train a series of models using machine learning. The proper implementation of this approach can require a lot from the implementer, such as expertise in linguistics, the domain of study, and machine learning optimization. In this paper, we examine two alternative approaches that allow the implementer to remain focused and specialized. We approach fake news classification with 1) automated machine learning ("AutoML") on engineered features, and 2) deep learning on word embeddings with no manually engineered features. The former approach allows linguists and domain experts to focus on feature engineering, while the latter allows deep learning scientists to focus on model architecture. We found that feature engineering alongside AutoML is a powerful combination for developing an accurate fake news classifier and is capable of outperforming deep learning neural network frameworks. With our AutoML approach, we are able to outperform the model used by Perez-Rosas et al. Nevertheless, our deep-learning-based approaches also performed well and proved to be effective at detecting fake news.

## 1 Introduction

More than ever, people are turning to social media to obtain news. This has led to an increase in people relying on news from sources that don't go through the rigorous vetting process of traditional media outlets. News accessed via social media is more likely to contain fake information,[1] because it contains a larger portion of articles from authors who want to deliver a fake message or are financially motivated to elicit more pageviews. It is challenging to automatically detect fake news because the most accurate detection method is via manual review by professional fact checkers. This process is not scalable or quick enough to keep up with a constant influx of news. Computers can address this issue by running algorithms that can predict if a news article is fake or not.

These algorithms can be built using natural language processing ("NLP"), via a process of creating systems that can "understand" language in order to perform certain tasks. This process once required extensive feature engineering, which is still useful, but often requires domain knowledge in linguistics and/or in the subject of focus. In recent years, deep learning algorithms, which don't require feature engineering or specific domain knowledge, have been shown to perform comparably.[2] Deep learning methods rely on architectures, such as neural networks, and focus on learning data representations, as opposed to task-specific algorithms. In this paper, we classify fake news using two different NLP methods: automated machine learning ("AutoML") on engineered features and deep

[1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. "Fake News Detection on Social Media: A Data Mining Perspective," https://arxiv.org/pdf/1708.01967.pdf, 2017.
[2] Samir Bajaj, "Fake News Detection Using Deep Learning," https://web.stanford.edu/class/cs224n/reports/2710385.pdf, 2017.

learning on word embeddings. We ultimately show that feature engineering with AutoML is a superior approach for the specific datasets we employed.

## 2 Background

NLP methods for fake news detection have focused on using linguistic-based features pertaining to the content of the news article.[3] Some of the most important linguistic-based features fall into four categories: lexical, syntactic, psycholinguistic, and readability. Lexical features include character and word level properties, such as total words, characters per word, and frequency of large or unique words. Syntactic features include sentence-level properties, such as frequency of words (i.e. "bag of words"), frequency of phrases (i.e. "n-gram"), punctuation and parts of speech tagging. Sentiment falls into the category of psycholinguistic features. The final group includes things like sentence readability and complexity.[4] All of these features have shown the ability to be used to detect fake news.

NLP methods for fake news detection have also focused on using deep learning.[5] Deep learning typically involves converting the words in a news article into vectors of numbers, and then using those numbers as inputs into neural network frameworks that learn patterns that are predictive of fake news. We are going to focus on implementing a few different versions of neural networks and compare their performance to models that rely on linguistic based features.

In order to do this, we will rely on the data and models used in the paper titled "Automatic Detection of Fake News,"[6] written by researchers at the Universities of Michigan and Amsterdam, to establish a target accuracy for our algorithms. These researchers used two datasets that both contain news, that is labeled as fake or not, to train models that rely on linguistic features and can be used to predict the label of other pieces of news. They used the following features: ngrams, punctuation, psycholinguistic, readability and syntax. With these, they achieved an accuracy of 0.74 on the "FakeNewsAMT" data set, and 0.76 on the "Celebrity" data set. We will attempt to outperform the paper using AutoML on a similar set of engineered features, as well as deep learning, specifically with neural networks, on word embeddings.

## 3 Data

*Data Set 1: FakeNewsAMT*[7]

This data contains 240 records of excerpts, usually two or three paragraphs, of legitimate news from mainstream news websites including ABC News, CNN, USA Today, New York Times, Fox News, and Bloomberg. The content covers six different domains (sports, business, entertainment, politics, technology, and education), and was manually fact-checked in order to verify its authenticity. 240 records of fake news were manually generated by Amazon Mechanical Turk ("AMT") workers who were instructed to write one piece of fake news that imitates the topic, length, and journalistic style of each piece of legitimate news. The fake news had to have all proper nouns found in the real news. The resulting data set includes 480 records, consisting of an equal proportion of real and fake news.

[3] "Fake News Detection on Social Media: A Data Mining Perspective," 2017.
[4] Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea. "Automatic Detection of Fake News," http://aclweb.org/anthology/C18-1287, 2018.
[5] "Fake News Detection Using Deep Learning," 2017.
[6] "Automatic Detection of Fake News," 2018.
[7] "Automatic Detection of Fake News," 2018.

*Data Set 2: Celebrity*[8]

This data contains 500 news articles related to public figures, since they are frequently targeted by rumors and fake reports. The sources of information include Entertainment Weekly, People Magazine, and Radar Online. The data was evaluated using gossip-checking sites and collected in pairs of fake and corresponding legitimate stories. The resulting data set has an even distribution of each classification.

We split each of these data sets into three distinct sets, producing a 60:20:20 split for training, validation and testing.

## 4 Models

In order to meaningfully compare deep learning and AutoML with feature engineering, we built the following models:

1. AutoML on Manually Engineered Linguistic-Based Features
2. Long Short-Term Memory (One Layer and Two Layers)
3. Gated Recurrent Unit

In both FakeNewsAMT and Celebrity, the distribution of fake and not fake news is uniform. Therefore, our baseline model for both datasets is guessing fake or not fake news at random. On average, the resulting accuracy of this baseline model is 50% on both datasets.

Perez-Rosas et al. used a linear SVM classifier with feature engineering and achieved accuracies of 74% and 76% on FakeNewsAMT and Celebrity, respectively.[9] We use these accuracies as a benchmark when we look to 1) improve Perez-Rosas et al.'s approach with expanded feature extraction and AutoML and 2) implement deep learning models.

### 4.1 AutoML on Manually Engineered Linguistic-Based Features

We attempted to replicate Perez-Rosas et al.'s approach for feature engineering and expand on their approach with automated machine learning. For model building, we implemented AutoML with the TPOT framework, "an open source genetic programming-based AutoML system that optimizes a series of feature preprocessors and machine learning models with the goal of maximizing classification accuracy on a supervised classification task."[10] As feature engineering can often require domain expertise (especially if done to its full capacity), we wanted to explore an automated modelling framework so that future domain experts could focus more on feature building and less on model building.

For our feature engineering, we applied a similar approach to the "Automatic Detection of Fake News" paper by building five sets of linguistic-based features (ngrams, punctuation, psycholinguistic, readability, and syntax).

*Ngrams*: As in the paper,[11] we extracted unigrams and bigrams and weighted the extractions with TF-IDF.

[8] "Automatic Detection of Fake News," 2018.
[9] "Automatic Detection of Fake News," 2018.
[10] Randal S. Olson and Jason H. Moore. "TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning," http://proceedings.mlr.press/v64/olson_tpot_2016.pdf, 2016.
[11] "Automatic Detection of Fake News," 2018.

*Punctuation*: We included punctuation as complete words in the text. Therefore, the punctuation in the articles were counted and normalized in the n-gram portion. For example, the phase "ran!" would be counted as two units for the unigram and bigram extraction: "ran," "!," and "ran !."

*Psycholinguistic Features*: As in the paper,[12] we also used the Linguistic Inquiry and Word Count ("LIWC") lexicon[13] to add psycholinguistic context to each article. The LIWC software assigns words to 73 LIWC categories in a binary format. Some examples of concepts embodied in these categories include: "affiliation," "achieve," "power," "reward," and "risk."

*Readability*: As in the paper,[14] we calculated the readability metrics "Flesch-Kincaid, Flesch Reading Ease, Gunning Fog, and the Automatic Readability Index (ARI)."[15] In addition, we calculated the readability metrics "The SMOG Index," "The Coleman-Liau Index," "Linsear Write Formula," "Dale-Chall Readability Score," and a readability consensus based on all of these listed metrics. We compute all 10 of these readability metrics using a single computing library, "textstat."[16]

*Syntax*: Like our approach with punctuation-based feature building, we simply counted and normalized syntactic patterns seen in the text. We did this by tagging the part-of-speech ("POS") of each word, replacing the word with the POS, extracting unigrams, bigrams, and trigrams of the resulting "sentence," and finally weighting those extractions with TF-IDF.

We developed two models, one for each dataset (FakeNewsAMT and Celebrity). For our models, we used AutoML with the Python library, TPOT. Upon applying the TPOT algorithm to our feature-enriched datasets, we arrived at a Random Forest model with optimized hyperparameters in both instances.

### 4.2 Long Short-Term Memory

The first deep learning network we applied is a Recurrent Neural Network ("RNN") with Long Short-Term Memory ("LSTM") units. RNNs have shown to be one of the most effective algorithms for classifying sequential data, because at each step in a sequence they are able to effectively remember important patterns from previous steps.[17] News has a temporal aspect to it, because a word used in a sentence depends heavily on words used before and after. In our RNN, each news article represents an input sequence, and each word in the article is associated with a specific time step. LSTM cells have input, output, and forget gates that are particularly good at remembering patterns over arbitrary time intervals, because they regulate the flow of information in and out of the cell. LSTM cells are typically better than traditional RNN cells at encapsulating information about long range dependencies, because traditional cells are more sensitive to the distance between steps, due to exploding and vanishing gradients.[18]

---

[12] "Automatic Detection of Fake News," 2018.
[13] James W. Pennebaker, Roger J. Booth, Ryan L. Boyd, and Martha E. Francis. "Linguistic Inquiry and Word Count: LIWC2015," https://s3-us-west-2.amazonaws.com/downloads.liwc.net/LIWC2015_OperatorManual.pdf, 2015.
[14] "Automatic Detection of Fake News," 2018.
[15] "Automatic Detection of Fake News," 2018.
[16] Shivam Bansal, Chaitanya Aggarwal. https://pypi.org/project/textstat/.
[17] Sepp Hochreiter and Jurgen Schmidhuber. "Long Short Term Memory," https://www.bioinf.jku.at/publications/older/2604.pdf, 1997.
[18] Boxuan Yue, Junwei Fu, and Jun Liang. "Residual Recurrent Neural Networks for Learning Sequential Representations," https://www.mdpi.com/2078-2489/9/3/56/htm, 2018.

We tested many variations of hyperparameters and adopted networks that resulted in the best performance on the validation data. Our final networks capped the sequence length of the FakeNewsAMT data at 200 words, and the Celebrity data at 500 words, since most of the articles were under these lengths, apart from some outliers. We converted each word to a 50-dimensional vector by training word embeddings that minimized the loss function of the entire network. This method was more effective than using pre-trained GloVe embeddings, which are trained in a manner that aims to capture word semantics. We used a hidden layer of 100 neurons. We believe smaller hidden layers performed worse because they could not detect certain data patterns, and larger layers performed worse due to overfitting the training data. We decided to use Adam Optimizer with a learning rate of 0.0001. We initially used a larger rate, which resulted in the undesirable behavior of divergence from the loss function minimum, meaning there was a quick drop, followed by an increase, in loss. Finally, we used a dropout rate of 0.5 which resulted in a network that is capable of better generalization and is less likely to overfit the training data. Dropout is a technique where randomly selected neurons are ignored during training.

### 4.3 Gated Recurrent Unit

A Gated Recurrent Unit ("GRU") is another unit that can alleviate the vanishing gradient problem inherent in a standard RNN. A GRU is like a LSTM in the way it works, as well as in its performance at classifying sequences of data.[19] A GRU is different because it uses update and reset gates. Our implementation of the GRU network hyperparameters is the same as the LSTM network.

## 5 Results

Table 1: Hyperparameters and Performance of Applied Models: FakeNewsAMT Dataset

| Model | Embedding Size | Sequence Length | Learning Rate | Optimizer | Hidden Memory Size | Dropout Rate | Epochs | Accuracy |
|---|---|---|---|---|---|---|---|---|
| Automated Machine Learning on Engineered Features | | | | | | | | 0.77 |
| LSTM - 1 Layer | 50 | 200 | 0.0001 | Adam | 100 | 0.5 | 25 | 0.57 |
| LSTM - 2 Layer | 50 | 200 | 0.0001 | Adam | 100 | 0.5 | 25 | 0.63 |
| GRU | 50 | 200 | 0.0001 | Adam | 100 | 0.5 | 30 | 0.55 |

Table 2: Hyperparameters and Performance of Applied Models: Celebrity Dataset

| Model | Embedding Size | Sequence Length | Learning Rate | Optimizer | Hidden Memory Size | Dropout Rate | Epochs | Accuracy |
|---|---|---|---|---|---|---|---|---|
| Automated Machine Learning on Engineered Features | | | | | | | | 0.81 |
| LSTM - 1 Layer | 50 | 500 | 0.0001 | Adam | 100 | 0.5 | 35 | 0.69 |
| LSTM - 2 Layer | 50 | 500 | 0.0001 | Adam | 100 | 0.5 | 25 | 0.64 |
| GRU | 50 | 500 | 0.0001 | Adam | 100 | 0.5 | 45 | 0.68 |

The two tables above show the performance of each model on each of the two data sets.

[19] Junyoung Chung, Caglar Gulcehre, Kyung Hyun Cho, Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," https://arxiv.org/abs/1412.3555, 2014.

AutoML on manually engineered linguistic-based features delivered the best performance with accuracies of 0.77 and 0.81 on FakeNewsAMT and Celebrity, respectively. This method outperformed the original project which achieved 0.74 and 0.76 accuracies using a linear SVM classifier. Since we used similar features, this increase in performance is mainly attributed to using automated machine learning, which allowed us to efficiently apply many variations of models and hyperparameters in order to determine the best model.

The best performing deep learning models were a two-layer LSTM on FakeNewsAMT with an accuracy of 0.63, and a single layer LSTM on Celebrity with an accuracy of 0.69. In the cases of both datasets, deep learning proved to be less effective than using feature engineering. However, these are still useful models, and in the case of the Celebrity data, achieved good accuracy that was only 0.07 less than the original project. Deep learning fared worse on FakeNewsAMT. This is because this data was manually created by Mechanical Turk workers who deliberately wrote fake articles that had the same topic, length and style as the corresponding real articles. We believe the Celebrity data contains a more accurate representation of fake news, since all these articles were collected from the web. Generators of fake news typically prioritize writing articles that incite emotional responses over writing articles that replicate real news content. Another reason deep learning method performed worse on FakeNewsAMT is because the articles were cut down to excerpts of around two or three paragraphs. We were able to use models with a longer sequence length of 500 words on the Celebrity data, which allowed the models to detect more patterns.

We believe our deep learning models were also restricted by the number of examples in each dataset. Each set contained around 500 total articles, with which we used around 300 for training. This is not a lot of data, and neural networks tend to learn better with more data. Larger data sets are available but are classified based on the source (i.e. source A is not reputable therefore every piece of news from source A is fake). This concerned us because non-reputable sites could produce real news, and reputable sites could produce fake news. The field of fake news detection could benefit from a larger corpus of individually labeled articles being made publicly available.

## 6 Conclusion

This project shows that feature engineering alongside AutoML is a powerful combination for developing an accurate fake news classifier and is capable of outperforming deep learning neural network frameworks. We have shown that future linguists and/or domain experts can use automated machine learning, which will allow them to focus more exclusively on feature engineering. Furthermore, linguists and/or domain experts with less experience in machine learning should be freed to leverage their expertise to help tackle the most pressing problems in text classification and, therefore, fake news detection.

This does not suggest deep learning should be ignored. It can detect patterns that may not have not been picked up by our engineered features. It also has the added benefit of not requiring domain expertise in linguistics, which was necessary to identify useful features. We believe the most robust models will combine both approaches and would like to explore this combined NLP approach in our future work.

The future of fake news detection will involve other important methods that can be used in addition to NLP. These methods consist of social context and fact-checking approaches.[20] Social context revolves around user-driven engagements of news on social media platforms (e.g. credibility of users who post and interact with an article, content of comments on an article, etc.). Fact-checking is a knowledge-based approach that aims to verify the truthfulness of the major claims in an article, through vehicles such as experts, crowdsourcing and algorithms. The future of fake news detection will not be one-size fits all. The most effective fake news classifiers will enlist all these approaches.

[20] “Fake News Detection on Social Media: A Data Mining Perspective,” 2017.

## References


[1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. *Fake News Detection on Social Media: A Data Mining Perspective*, https://arxiv.org/pdf/1708.01967.pdf, 2017.

[2] Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H.. *Fake News Detection on Social Media: A Data Mining Perspective*, https://web.stanford.edu/class/cs224n/ reports/2710385.pdf, 2017.

[3] Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea. *Automatic Detection of Fake News,* http://aclweb.org/anthology/C18-1287, 2018.

[4] Samir Bajaj. *Fake News Detection Using Deep Learning* https://web.stanford.edu/class/cs224n/reports/2710385.pdf, 2017.

[5] Shivam Bansal, Chaitanya Aggarwal. https://pypi.org/project/textstat/.

[6] Sepp Hochreiter and Jurgen Schmidhuber. *Long Short Term Memory,* https://www.bioinf.jku.at/publications/older/2604.pdf, 1997.

[7] Boxuan Yue, Junwei Fu, and Jun Liang. *Residual Recurrent Neural Networks for Learning Sequential Representations*, https://www.mdpi.com/2078-2489/9/3/56/htm, 2018.

[8] Randal S. Olson and Jason H. Moore. *TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning*, http://proceedings.mlr.press/v64/olson_tpot_2016.pdf, 2016.

[9] James W. Pennebaker, Roger J. Booth, Ryan L. Boyd, and Martha E. Francis. *Linguistic Inquiry and Word Count: LIWC2015*, https://s3-us-west-2.amazonaws.com/downloads.liwc.net/LIWC2015_OperatorManual.pdf, 2015.

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, https://arxiv.org/abs/1412.3555, 2014.

[11] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. N*atural Language Processing (almost) from Scratch*, https://arxiv.org/pdf/1103.0398v1.pdf, 2011.

[12] Yoon Kim. *Convolutional Neural Networks for Sentence Classification*, https://arxiv.org/abs/1408.5882, 2014.