Fake News Classification Using Text-Based Features

Tim Burke and Evelyn Mwangi

Abstract

This paper reports on the detection of fake news using features scoped over the article text. The analysis is based on a corpus of 1,604 news stories collected and fact-checked by Buzzfeed News during the 2016 election. The articles are evaluated with text features in a logistic regression model as well as with CNN and LSTM models, none of which materially outperform a naive baseline. These results suggest text features alone are not viable for differentiating news stories by veracity.

1 Introduction

The identification of fake news, broadly defined as intentionally and verifiably false stories that could mislead readers (Allcot and Gentkow, 2017), represents a substantial challenge in the field of natural language processing. As a prominent issue in the 2016 U.S. presidential election, "fake news" has attracted a great deal of interest from the research community. Allcott and Gentkow found that the average voter had been exposed to at least one fake story in the months before the election, and that about half had believed the deceptive news. While the effect of partisan fake news on the outcome of the election remains uncertain, it made clear that fake news has the potential to deceive on a large scale. The implications of fake news are especially serious in the political realm, where disinformation can be used to warp political dialogue and manipulate audiences. In the example of the 2016 election, a Russian propaganda campaign spread fake news stories meant to increase partisanship and alter the outcome of the election.[1] Without the ability to quickly identify and respond to fake news, similar methods could be used to undermine democratic institutions.

The volume of misleading content creates the need for an automatic detection system (Chen et al, 2015), and substantial progress has been made in this direction using a variety of approaches. Nonetheless, a lack of quality training data has limited research on fake news detection based on article text. In this paper, we analyze text-based features for fake news detection using one of the largest and most comprehensive fake news datasets available: The Buzzfeed-Webis Fake News Corpus 2016 (Silverman, Strapagiel, Shaban, Hall, Singer-Vine, 2016). We apply a logistic regression model on the text of each article to evaluate which features are useful for differentiation fake and real news. Additionally, we apply a CNN and a LSTM model to the text, then compare our results with the logistic regression model and past work on this dataset.

[1] Timberg, C. (2016, November 24). Russian propaganda effort helped spread 'fake news' during election, experts say. Retrieved December 06, 2017

After an overview of related work in this space, section 3 discusses the composition of the Buzzfeed dataset, section 4 discusses the methodology behind each model, section 5 presents experimental results and section 6 discusses conclusions.

2 Related Work

The current body of fake news detection research can generally be classified into three main focuses: Fact checking (knowledge-based), social network analysis (context based) and style-based text analysis. Fact-checking as a form of fake news detection focuses on identifying major claims in an article and attempting to verify them using external sources. Fact checking with human experts is generally the most robust method, but cannot easily scale to cover the enormous amount of news content generated daily. To automate the process, check-worthy claims can be extracted from text based on features such as sentence length, part of speech tags and sentiment (Hassan et al, 2015). These claims must then be verified against external sources. One approach is to make a knowledge graph that encodes entities as nodes and their relationships to other entities as edges; Ciampaglia et al (2015) extract such a knowledge graph from Wikipedia and check to see if a dataset of novel claims can be inferred from the existing facts in the graph. Etzioni et al (2008) take a second approach, deploying the Text Runner tool to extract and compare facts across the web and in documents. The challenge of these approaches lies primarily in the "verification" step, as the reliability of facts from specific sites like Wikipedia or the web at large can be called into question.

Context-based approaches seek to identify fake news by investigating how users engage with or spread deceptive content on social networks. There exists a rich body of research on this topic, much of it combining a focus on the granular details of social media posts with context about the networks of users that share the articles. Ruchansky, Seo, & Liu (2017) create a fake news classification model combining information about each user's temporal engagements with an article, the text of their posts and a "score" on the trustworthiness of those users. This achieves an impressive classification accuracy of 89% against a dataset of news articles shared over Twitter. Jin et al (2017) focus on user reactions to each post, using topic extraction to mine conflicting viewpoints from posts about news events. These conflicting viewpoints are combined into a credibility network, which achieves 84% verification accuracy on a dataset of news events on Sina Weibo.

Style-based fake news detection, which focuses on finding deceptive cues in the text of the news articles, builds upon a substantial body of existing work in deception detection across other domains. These studies are typically based on a single genre (such as fake business reviews) and use a variety of natural language processing techniques to identify deception. In the domain of reviews, these have included n-grams models (Mukherjee et al, 2013) sentiment analysis (Ott et al, 2011) and syntactic stylometry (Feng et al, 2012). Fake news detection has built upon these works, but also faces unique challenges: "Fake news" encompasses multiple language domains, from long-form articles and blogs to short twitter posts or interview statements. Researchers have addressed these challenges in a variety of ways.

The 2017 Fake News Challenge, a public research contest to advance research in fake news detection, focused on identifying "fake" articles by inconsistencies between the headline and body of 50,000 headline-article pairs. Contestants sought to classify the headlines as agreeing with the body, discussing the body without a stance, disagreeing with the body or being entirely unrelated to it. Mrowca et al (2017) addressed the issue with stance detection, using a bidirectional LSTM model supplemented with global features. Chaudry et al (2017) approached the same contest with a greater variety of models including Jaccard similarity, feed-forward neural networks, and several versions of LSTM models. Both Chaudry et al and Mrowca achieved impressive accuracy in classifications of headlines that agree or are unrelated to the content, but the uneven distribution of the training data made identifying disagreements difficult. While the contest produces numerous impressive models, the operational definition of "fake news" as headline-body disagreement may address a

different domain than the more common definition of intentionally and verifiably false content.

The research on classifying verifiably deceptive news stories based on the text of the articles is more limited, in part because of the lack of sizeable training datasets. As such, much of the research uses some proxy for fake news rather than actual fake content identified in the public domain. Rubin et al (2015) use rhetorical structure analysis on a selection of fake stories created for the NPR radio show "Bluff the Listener", while Badaskar et al (2008) use syntactic and semantic features to differentiate real articles from fakes generated by a trigram language model. These approaches lay the groundwork for fake news detection in more comprehensive datasets.

The recent emergence of large-scale fake news datasets create the opportunity to train more robust text-based fake news detection models. Wang (2017) introduces the LIAR dataset, comprised of roughly 12,800 short statements from the US political fact-checking website Politifact.com. Each statement was fact-checked and manually labeled for veracity along six categories from true, mostly true, half-true, false, mostly false, and pants-on-fire. It also includes information about the speaker of the statement, such as the speaker's political party, state, and history of false or true statements. The distribution of statement classifications and speaker political affiliation is fairly even across the dataset. These data are well-suited to classification of short statements, although they do not include the full text of the article or interviews from which the statements were sourced. Wang analyzes the data as a classification task into the six Politifact categories, first using linguistic models and then supplementing them with features from the metadata associated with each statement. These include an SVM model, logarithmic regression, bidirectional LSTM, CNN, and hybrid CNNs with features from the aforementioned information about the speaker. Wang finds the hybrid models to be the most effective in classifying the data, achieving accuracies of 27%.

A second promising dataset, and the focus of this paper, is the Buzzfeed-Webis Fake News Corpus 2016. The data include the headlines and full text of news articles from nine news sources, including mainstream organizations and "hyper-partisan" websites on the left and right. The articles were collected over a week near the end of the election, and each was manually categorized as either mostly true, mostly false, mixed or "no factual content" by Buzzfeed journalists. Potthast et al (2017) enriched this dataset by adding permanent links to the relevant articles as well as attached media and associated metadata. While the scope of the data are relatively narrow, covering only a short time and a small selection of publishers, its scale and the presence of article metadata create the potential for performing interesting analyses. Potthast et al studied the Buzzfeed corpus with a stylometric approach, using Unmasking to identify the styles associated with hyper-partisan writing on the left and right. The study also examines whether fake news can be detected based on style and if satire can be differentiated from other news. Their model finds a great deal of stylistic similarity between hyper-partisan writing on the left and right, and also is able to differentiate satire from other news based on style. However, the model is not effective at identifying fake news solely based on style.

3 Data: The Buzzfeed-Webis Fake News Corpus

We use the Buzzfeed-Webis Fake News Corpus 2016 to perform our analysis. As discussed in Section 2, the data consist of a total of 1,627 news articles that had been shared over Facebook. The articles were collected from September 19-23 and September 26-27 from nine news sources: Three mainstream (CNN Politics, ABC Politics, Politico), three left-leaning (The Other 98%, Addicting Info, Occupy Democrats), and three right-leaning (Eagle Rising, Right Wing News, Freedom Daily). Data for each article include the publisher, the partisan leaning of the publisher, a link and the number of facebook comments, shares and reactions. Buzzfeed defined "fake news" as articles based on verifiably false claims, and five journalists were assigned with fact-checking the corpus. As nearly all stories contain a mix of true and inaccurate claims, Buzzfeed classifies articles on

a scale from "mostly true," "mix of true and false," "mostly false," and "no factual content" (see Table 1 for summaries of annotator guidelines for each label). The ratings of "mostly false" or "mixture of true and false" required justification by the journalist, and each "mostly false" article had to be classified as such by a second reviewer. Ambiguous cases were reviewed by a second journalist, and if there was

Table 1: Summary of Buzzfeed Label Guidelines

Mostly True	Authors may interpret info in their own way, so long as the info in the story is fact-based and not misrepresented. This does not allow for unsupported claims or speculation
Mix of True and False	Speculation or unfounded claims are mixed with real events and evidence, the headline is misleading but the text is largely accurate, or the story is based on unconfirmed info
Mostly False	Most or all of the information in the story is inaccurate, or the central claim being made is false
No Factual Content	Stories that are pure opinion, comics, satire, or do not make a factual claim.

disagreement between the two then a third journalist would evaluate the article and make the final decision. Potthast et al enriched the dataset by adding permanent links for each article, the full text with quotes and external links identified, and the author of each piece. The process of finding permanent links for all articles cut the corpus size from the original 2,282 articles to 1,627, as many of the original links had expired. A further 23 "articles" do not contain any main text on which to perform analysis - primarily because they are videos or images without associated text commentary. These images are excluded for the purpose of our analysis, bringing the total size of the usable corpus to 1,604 articles. The removed articles included fifteen "mostly true" stories, three "mixed" articles and five "mostly false" articles.

Corpus Statistics: Table 2 contains a selection of descriptive statistics relating to the Buzzfeed-Webis Fake News Corpus 2016. The data are biased towards "mainstream" and "right wing" news articles, which account for 51% and 33% of the corpus, respectively. The distribution of article veracity is also skewed - 78% of articles are "mostly true", while only 5% are "mostly false." No mainstream articles are marked "mostly false," perhaps owing to the fact that many mainstream news organizations tend to strictly fact check stories before publication. The distribution of "mostly false" stories is heavily skewed towards right-wing publishers, which accounted for 83% of such articles in the data.

Table 2: Buzzfeed Corpus Statistics

Publisher	Fact Checking Results					Key Statistics Per Article				
						Avg. Number Paragraphs	Avg. Number Links		Avg. Number Words	
	True	Mix	False	N/A	Total		External	All	Quoted	All
Mainstream	806	8	0	12	826	20.1	2.2	3.7	18.1	692.0
ABC	90	2	0	3	95	21.1	1.0	4.8	21.0	551.9
CNN	295	4	0	8	307	19.3	2.4	2.5	15.3	588.3
Politico	421	2	0	1	424	20.5	2.3	4.3	19.9	798.5
Left Wing	182	51	15	8	256	14.6	4.5	4.9	28.6	423.2
Addicting Info	95	25	8	7	135	15.9	4.4	4.5	30.5	430.5
Occupy Democrats	59	25	7	0	91	10.9	4.1	4.7	29.0	421.7
The Other 98%	28	1	0	1	30	20.2	6.4	7.2	21.2	394.5
Right Wing	276	153	72	44	545	14.1	2.5	3.1	24.6	397.4
Eagle Rising	106	47	25	36	214	12.9	2.6	2.8	17.3	388.3
Freedom Daily	49	24	22	4	99	14.6	2.2	2.3	23.5	419.3
Right Wing News	121	82	25	4	232	15.0	2.5	3.6	33.6	396.6
Total	1264	212	87	64	1627	17.2	2.7	3.7	20.6	551.0

4 Methodology

We analyze the dataset in two separate approaches - one using the entire dataset and one using only partisan news sources. The full dataset is likely more representative of the real news environment - the majority of content is true, and mainstream news organizations are more represented than smaller partisan outlets. However, mainstream outlets make up 50% of the data and publish effectively no fake news - none of their articles are "mostly false" and only eight are a "mix of true and false." If it holds true in reality that mainstream publishers almost never make false content, then the task of fake news identification can be simplified by first excluding any mainstream publishers. Following this logic, we also train a classifier for the more limited problem of identifying fake news exclusively among partisan sources. In keeping with the process of Potthast et al, we also exclude articles with 'no factual content' as they are not relevant to the domain of fake news. This reduces the dataset to a size of 730 articles, in which 61% of articles are "mostly true," 28% are a "mixture of true and false" and 11% are "mostly false." This dataset is also randomly split 80/20 into training and test sets. Both models operationalize fake news using the Buzzfeed definitions and seek to classify articles to one of the Buzzfeed veracity ratings, with the

exception of removing articles with no factual content for one analysis. Potthast et al's approach to the same dataset classified "mixture of true and false" and "mostly false" articles as "fake" and "mostly true" as "real". We use the Buzzfeed labels because we believe it is important to keep the distinction between degrees of "fake" news.

Logistic Regression: We use a logistic regression model to analyze text features for detecting fake news. For the logistic regression models, news article text is represented using two separate methods - TF-IDF vectors and the element-wise average of the Google News word embeddings over the text. The Google News embeddings[2] are a selection of 3 million dense representations of words trained over a corpus of over 100 billion words from Google News. Both logistic regression models use L2 regularization, the strength of which is selected with stratified k-fold cross validation run over the training dataset. Additional tests evaluate the use of different scoring metrics including accuracy, recall, precision, F1 score and numerous values of F-beta score. The final model is fit using F1 score.

Many of the features used in the logistic regressions have been effective in other deception detection tasks - Affron, Brennan and Greenstadt (2012) use a selection of these to detect hoaxes and forgeries in writing style. The features used are:

1) Number of words, number of long words and number of unique words
2) Average number of syllables per sentence as well as the number of monosyllabic and polysyllabic words
3) Flesch readability score
4) Number of first, second and third person pronouns
5) Number of conjunctions and modal verbs
6) Number of hedge words (Words that show vagueness)
7) Number of weasel words (Words that show uncertainty)
8) Number of quotes
9) Number of links

To obtain the count of modal verbs in each article, the text is first run through the NLTK part of speech tagger[3] before counting the number of modal verb tags. The hedge word and weasel word features are designed using publicly available dictionaries of such words.[4] Details on the top features are presented in the appendix section.

CNN and LSTM: For the purpose of comparison with the logistic regression model, we also apply a convolutional neural net and Long-Short Term Memory model to the data. These models have produced impressive results in similar domains - Wang et al apply a CNN to detect false statements, and Chaudry et al apply multiple LSTM models to the task of detecting article-headline mismatch. In the case of our models, article text is represented using the pre-trained Google News word embeddings. This is because the Buzzfeed corpus is too small to train meaningful embeddings. To account for differing article word counts, which range from 6 to 6,570, each article representation is either cut down or "padded" to a uniform maximum length of 700 words. The CNN model includes an embedding layer, three convolutional layers that are each followed by max pooling and dropout layers, and a final two dense layers (see Appendix, Figure 1). All convolutional and dense layers use ReLU activation functions with the exception of the final softmax dense layer. Both the LSTM and CNN use categorical cross entropy to calculate loss. To evaluate the hyperparameters of the CNN, the training data are again randomly split 80/20 into training and validation sets. Grid search is performed to evaluate dropout rate, kernel size, number of filters, batch size, number of training epochs, class weighting and maximum word length. Additional validation tests are performed with other CNN architectures, including shallower networks, fewer dropout layers, L2 regularization and no regularization. The final model was selected with a kernel size of 5, dropout rate of 50%, maximum article length 700, 300 filters, one training epoch and a batch size of 300.

[2] https://code.google.com/archive/p/word2vec/

[3] http://www.nltk.org/book/ch05.html

[4] https://github.com/words/hedges, https://github.com/words/weasels

The LSTM is a straightforward model feeding through an embedding layer, an LSTM layer and then into a dense sigmoid layer. Similar to the CNN, the LSTM uses dropout layers both before and after the LSTM layer to prevent overfitting. The model uses the same validation set as the CNN to select hyperparameters including dropout rate and the output size of the LSTM layer. The final model was selected with an output size of 300, dropout rate of 50%, maximum article length of 700, batch size of 300 and a single training epoch.

Baseline: To set a baseline for fake news classification accuracy, we classify all articles as the majority class of "mostly true." In addition to this naive baseline, we also examine the fake news classification results achieved by Potthast et al in their analysis of the same dataset.

5 Results and Analysis

The results of the baseline majority classifier model evaluated on the held out test set are described in Table 3.

Table 3: Naive Baseline Results

Evaluation Method	Score (Full Corpus)	Score (Partisan Only)
Accuracy	81.3%	63.0%
Avg. Precision	0.66	0.40
Avg. Recall	0.81	0.63
Avg. F1 Score	0.73	0.49

The imbalance in the data creates deceptively high average results - across the test set of 321 articles, 81% are mostly true. Results for the logistic regression, CNN and LSTM models are summarized below.

Table 4: Summary of Classification Results

Model (Full Corpus)	Accuracy	Avg. Precision	Avg. Recall	Avg. F1
Logistic Regression (TF-IDF)	82.2%	0.77	0.82	0.75
Logistic Regression (Embeddings)	73.8%	0.68	0.74	0.70
Logistic Regression (TF-IDF, features)	81.3%	0.73	0.81	0.74
Logistic Regression (Embeddings, features)	81.3%	0.66	0.81	0.73
CNN	81.3%	0.66	0.81	0.73
LSTM	81.3%	0.66	0.81	0.73
Model (Partisan Sources Only)	**Accuracy**	**Avg. Precision**	**Avg. Recall**	**Avg. F1**
Logistic Regression (TF-IDF)	65.1%	0.72	0.65	0.53
Logistic Regression (Embeddings)	45.9%	0.49	0.46	0.47
Logistic Regression (TF-IDF, features)	62.3%	0.50	0.62	0.50
Logistic Regression (Embeddings, features)	62.3%	0.50	0.62	0.50
CNN	63.0%	0.40	0.63	0.49
LSTM	63.0%	0.40	0.63	0.49

Note that both the CNN and LSTM models have the exact same scores as the naive baseline - this is because both models learn to classify all articles as "mostly true" due to the class imbalance of the dataset. The imbalance proves to be remarkably difficult to overcome. Adding class weighting to increase the relative importance of the minority veracity weightings is ineffective - the model can learn to stop classifying all articles as "mostly true" but instead classifies nearly all cases as one of the minority classes, tremendously decreasing accuracy without adding any real power to differentiate fake news. The tendency of both models to act as majority classifiers is remarkably resilient to changes in the neural network structure or hyperparameters. Using different loss functions, shallower networks, training epochs, or regularization still results in the same outcomes. This issue persists when the scope of the problem is limited to strictly partisan news sources, despite "mostly false" or "mixture of true and false" stories accounting for a higher proportion of the training data in this case. The combination of a strong class imbalance and relatively small dataset appear to make simple CNN and LSTM models impractical for this problem.

Logistic regression results only fare marginally better than the neural models. Surprisingly, the best results are achieved with logistic regression models that only include a TF-IDF or word embedding representation of the article text and no features. Unlike the CNN and LSTM models, the logistic regressions can be tuned to not always choose the majority class by fitting with F1 as the scoring metric. The TF-IDF logistic regression reaches the highest overall accuracy by getting perfect recall for "mostly true" articles as well as correctly classifying 20% and 6% of "no factual content" and "mix of true and false" articles, respectively. Interestingly, the TF-IDF logistic regression misses all "mostly false" articles while the embeddings version reaches an F1 score of 0.14 for "mostly false" stories (though at the cost of lower precision and recall across the other veracity ratings). The results are similar when both models are trained on the partisan-only dataset with "no factual content" articles removed. The TF-IDF logistic regression again achieves the highest accuracy, correctly classifying all "mostly true" articles and getting an F1 score of 0.12 for "mix of true

and false" articles. The embeddings logistic regression outperforms for "mix of true and false" storeies with an F1 score of 0.25, but again at the cost of lower F1 scores across the other veracity ratings. Neither logistic regression model correctly identifies any "mostly false" articles in the partisan-only dataset.

Adding features to the model actually causes a slight decrease in performance of both logistic regression models, which holds for the full corpus and the partisan-only dataset. Adding features does not affect the model's ability to identify "mostly true" articles, but greatly reduces their ability to identify any of the minority classes. Neither featurized model correctly identifies any of the minority labels in the full corpus, and both only achieve F1 scores of 0.04 for "mix and true and false" articles in the partisan dataset. It appears that the features are not useful for differentiating news article veracity, and therefore only serve to add noise to the model.

Overall, the performance of the logistic regression models is disappointing. Both models are generally effective at identifying "mostly true" articles, but cannot surpass a recall of 0.24 for any of the minority veracity ratings. Performance was especially weak for "mostly false" articles, suggesting these models would be ineffective at identifying "fake news." It it worth noting the differences in performance between the TF-IDF and Google News embeddings versions of the logistic regression: We expected that pre-trained embeddings from a large and similar domain would offer better overall performance by encoding a richer representation of the article text, but TF-IDF achieved higher accuracies in all cases. The embeddings did outperform the TF-IDF model with respect to classifying "mostly false" and "mix of true and false" articles, but always at the cost of lower accuracy. It is possible that representing an article as the element-wise average of its constituent word embeddings causes some of the embeddings' information to be lost, weakening their representational power. Taking the element-wise minimum and maximum across each article's word embeddings does not change this result.

These test results fall short of the classification accuracy reached by Potthast et al's model, but ultimately neither approach is able to successfully differentiate fake news. Potthast et al does not attempt to classify the full Buzzfeed corpus, focusing only on the partisan dataset and combining "mixture of true and false" and "mostly false" into a single label of "fake". Their models are more successful in differentiating "fake" articles, with a recall of 0.4-0.5. Nevertheless, their accuracy suffers with regards to true articles, and the models only reach accuracy of at most 58% with f1 scores of 0.58 to 0.66. The results of this paper and Potthast et al's research suggest that fake news is resistant to detection with stylistic features, even in the limited scope of identifying fake news in partisan writing.

Comparing the distribution of feature values across the different veracity ratings gives some insight into the difficulty of this problem: There is little variation in any of the features when looking across classes. For example, the average numbers of hedge words and weasel words are nearly identical across "mostly true" and "mostly false" or mixed articles. Features that do show variance between veracity ratings, such as number of sentences, appear to mostly reflect the "mostly true" ratings' association with longer articles published by mainstream outlets. In the case of the Buzzfeed Corpus, fake news is surprisingly similar to real stories in most respects. This may be by design - fake stories are intended to mislead readers, and thus seek to emulate the style of legitimate reporting as closely as possible.

6 Conclusions

In this paper, we examine the viability of detecting fake news in the Buzzfeed Webis 2016 corpus using stylistic features and simple CNN and LSTM models. Our research shows that the problem of fake news remains resistant to detection through text-based features scoped over the article body. Furthermore, the simple CNN and LSTM used in this paper do not appear to be a good fit for this problem. Fake news stories are looking more and more like real news, and differentiating the few false stories from the majority of factual reporting is a difficult task. Text-based features alone do not appear sufficient to tackle the problem, and future

research may benefit from pursuing other directions. Incorporating metadata about the analysis has offered a promising start in other research; Wang et al's hybrid CNN using statement text and metadata is one such example, as is Ruchansky, Seo, & Liu's approach of using metadata about articles shared over Twitter. The neural models used in this paper were not effective, but more complex models, or those trained over a larger dataset, may be able to overcome the challenges in identifying fake news.

7 References

Afroz, S., Brennan, M., & Greenstadt, R. (2012, May). Detecting hoaxes, frauds, and deception in writing style online. In Security and Privacy (SP), 2012 IEEE Symposium on (pp. 461-475). IEEE.

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election (No. w23089). National Bureau of Economic Research.

Badaskar, S., Agarwal, S., & Arora, S. (2008). Identifying Real or Fake Articles: Towards better Language Modeling. In IJCNLP (pp. 817-822).

Chaudhry, A. K., Baker, D., & Thun-Hohenstein, P. (2017). Stance Detection for the Fake News Challenge: Identifying Textual Relationships with Deep Neural Nets.

Chen, Y., Conroy, N. J., & Rubin, V. L. (2015). News in an online world: The need for an "automatic crap detector". Proceedings of the Association for Information Science and Technology, 52(1), 1-4.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., & Flammini, A. (2015). Computational fact checking from knowledge networks. PloS one, 10(6), e0128193.

Craig Silverman, Lauren Strapagiel, Hamza Shaban, Ellie Hall, Jeremy Singer-Vine. (n.d.). Hyperpartisan Facebook Pages Are Publishing False And Misleading Information At An Alarming Rate. Retrieved September 26, 2017, from https://www.buzzfeed.com/craigsilverman/partisan-fb-pages-analysis?utm_term=.dyjjZJ4zM#.stPlrxgbW

Feng, S., Banerjee, R., & Choi, Y. (2012, July). Syntactic stylometry for deception detection. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2 (pp. 171-175). Association for Computational Linguistics.

Hassan, N., Li, C., & Tremayne, M. (2015, October). Detecting check-worthy factual claims in presidential debates. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 1835-1838). ACM.

Jin, Z., Cao, J., Zhang, Y., & Luo, J. (2016, February). News Verification by Exploiting Conflicting Social Viewpoints in Microblogs. In AAAI (pp. 2972-2978).

Mrowca, D., Wang, E., & Kosson, A. (2017). Stance Detection for Fake News Identification.

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. S. (2013, July). What yelp fake review filter might be doing?. In ICWSM.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011, June). Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 309-319). Association for Computational Linguistics.

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017). A Stylometric Inquiry into Hyperpartisan and Fake News. arXiv preprint arXiv:1702.05638.

Rubin, V. L., Conroy, N., & Chen, Y. (2015, January). Towards news verification: Deception detection methods for news discourse. In Hawaii International Conference on System Sciences.

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A Hybrid Deep Model for Fake News. arXiv preprint arXiv:1703.06959.

Wang, W. Y. (2017). " Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. arXiv preprint arXiv:1705.00648.

8 Appendix

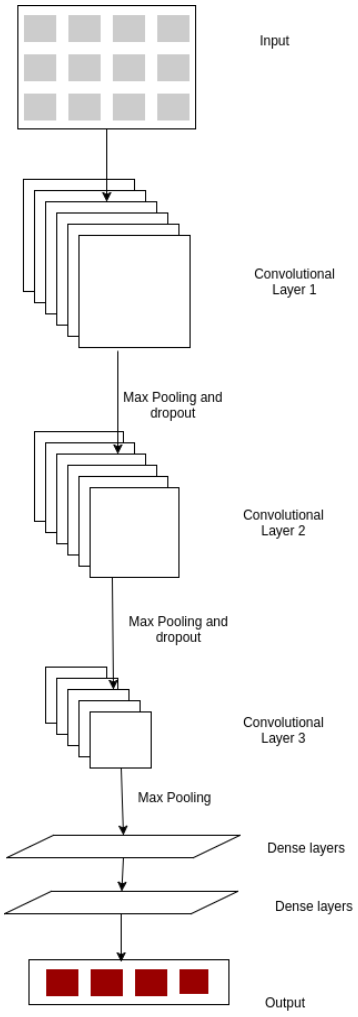

Fig1 . CNN model

Table 5: Top 5 TF-IDF Features Per Class

	Word	Coefficient
Mostly True		
	clinton's	-0.213
	endorsed	-0.215
	shared	-0.217
	are	-0.220
	there's	-0.220
Mixture of True and False		
	democratic	-0.110
	know	-0.110
	we	-0.110
	bring	-0.113
	real	-0.118
Mostly False		
	via	-0.215
	whether	-0.219
	scott	-0.220
	violence	-0.227
	gop	-0.228
No Factual Content		
	u.s.	-0.143
	before	-0.145
	white	-0.146
	out	-0.146
	them	-0.146