Fake News Detection: A deception detection technique using Machine Learning

Surya Sudharshan
UFID:5019-3163
University of Florida, Computer Science

***Abstract*—This project explores the application of natural language processing and machine learning techniques for the detection of 'fake news', that is, misleading news stories that come from non-reputable sources. In this day and age, we have a lot of places where we get our information and news from. The internet would be one amongst the major . But sometimes sources on the internet can't necessarily be trusted as people or algorithms might put up articles that might seem original but are in lieu, completely fake. This might lead to widespread misinformation amongst a lot of readers and have profound repercussions in society.**

Keywords—Natural language processing, Machine learning, Classification algorithms, Fake-news detection.

I. Introduction

The nature of online news publication has changed owing to the abundance of content generators, as well as various formats and genres. “Fake news detection” is defined as the task of categorizing news along a continuum of veracity, with an associated measure of certainty. Said veracity is compromised by the inclusion of intentional deception. The adverse effects of fake news cannot be trivialized, the effects run the gamut of a simple prank to the potential downfall of a multinational corporation or even governments.. In this project, we seek to produce a model that can accurately predict the likelihood that a given article is fake news.

Many social media sites like Facebook, Twitter and Instagram are at center of fake news critique. Facebook even has a feature where the user can flag news indicating its veracity. But a more optimal solution would be to leverage automation and machine learning algorithms. Indeed more and more companies are investing in research on the ability to distinguish news articles automatically. Granted it's a difficult task, an effective implementation must remain politically unbiased and give equal credence to legitimate news on both ends of the spectrum. In addition, the question of credence is a difficult and variegated one. The determination of what constitutes positive and negative credence should be done in an objective manner.

In this project, the efficacy of models are compared using different features like bag of words, TF-IDF using bi-gram frequency, POS tagging and word2vec vectorizer. The outline of the project follows the existing work of Collroy, Rubin and Chen [5] and Nishit Patel[19] for deception detection through Machine Learning and Natural Language Processing, all whilst using the dataset presented by William Yang Wang[1]. The results show that while the bag of words yield considerably good results at classifying articles from unreliable sources, it can be improved using the TF-IDF bi-gram frequency. The findings suggest that, the model's efficacy is not augmented much by the bag of words.

Section II briefly describes the past work done in the field of text classification and fake news detection. Section III describes the challenges of obtaining a dataset and a description about the final LIAR dataset used for training the classifier. Section IV illustrates the feature generation methodology and pre- processing steps. Section V delineates the actual modelling procedure and compares the outputs from the different al- gorithms. Finally, Section VI presents the conclusions and briefly illustrates the potential for further improvements in the proposed methodology.

II. Related Works

There is a vast body of existing research for the use of Machine Learning and Natural Language processing in deception detection and fake-news. A majority of these are concentrated on just segregating social media articles and internet news articles in a classifier. The impetus to dwelve into this domain was provided by the 2016 American Presidential election, when one of Donald Trump's favourite quotes "You are fake news!" caught on and became the topic of future work when it came to deception detection.

Collroy, Rubin and Chen [5] delineate multiple methods that hold promising results for rightly classifying news articles from unreliable sources. They denote that simple content-related n-grams and lacklustre part-of-speech (POS) tagging were not sufficient for the tedious and challenging task that classifying is. In fact, they talk about how these methods come handy when the analysis goes deeper down the rabbit hole in terms of complexity. Another promising angle is in Deep Syntax analysis using Probabilistic Context-Free Grammar (PCFG) in combination with n-gram methods. Feng, Bannerjee and Choi [4] were able to achieve a commendable accuracy of close to 91% in fake-news detection using corpora that was just available in plain sight online.

Feng and Hirst [5] implement a semantic analysis looking at 'object:descriptor' pairs for contradictions with the text on top of Feng's initial deep syntax model for additional improvement. Rubin, Lukoianova and Tatiana [6] analyze rhetorical structure using a vector space model with similar success. Ciampaglia et al. [7] employ language pattern similarity networks requiring a pre-existing knowledge base.

TABLE I: Distribution of top news sources, both reliable and unreliable.

Top Five Unreliable News Sources		Top Five Reliable News Sources	
Before It's News	2066	Reuters	3898
Zero Hedge	149	BBC	830
Raw Story	90	USA Today	824
Washington Examiner	79	Washington Post	820
Infowars	67	CNN	595

III. Data Preparation

A. Dataset Challenges

Conroy, Rubin and Chen [5] outline several requirements for a helpful corpus for use in these contexts:

1. Verifiability of 'ground truth'.
2. Availability of both truthful and deceptive instances.
3. Homogeneity in lengths and writing matter.
4. Predefined timeframe.
5. The manner of delivery

Contemporary resources for deception detection are majorly crowd sourced datasets. Though they are useful datasets to an extent, data for positive labels are generated from a simulated environment. Moreover these datasets are unsuitable for fake news detection, as the fake news that is generally floated around on social media or other forms are much shorter in length than customer reviews

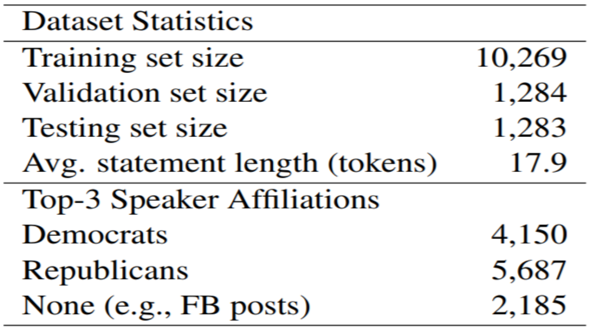
There are many impediments in the form of copyright issues and proprietary material when it comes to building a corpus of news articles. The first legitimate step in this direction was taken by Vlachos and Riedel(2014) when they built a fact-checking and fake news dataset. They were again beneficiaries of the vast collection of news articles POLITIFACT.COM and CHANNEL 4^2. POLITIFACT.COM is a Pulitzer-prize winning website that covers a broad spectrum of political news. They also provide the quintessential process of extensive fact-checking and provide fine-grained labels. Another noteworthy dataset was released by Ferreira and Vlachos (2016) which has around 300 labeled news articles from POLITIFACT.COM. The common problem with these datasets are the shortage of news articles, state of the art models require at least thousands of news articles to build and benchmark a process to implement machine learning algorithms to efficiently detect fake news

Therefore, its crucial to build a comprehensive dataset to buttress the development of machine learning based computation to efficiently and automatically detect fake news.

B. Dataset Description

The LIAR dataset that is used in this project contains close to 13K labeled succint news statements from the credible website POLITIFACT.COM's API. The credibility stems from the fact that each statement is verified by an editor from POLITIFACT.COM for its veracity. The dataset performs a normalization of sorts, by removing duplicate labels and combining the no-flip, full-flop, half-flip labels into more understandable labels such as true, false and half-true respectively. In the end, the dataset contains the final labels 6 in number and named as pants-fire, false, barely-true, half-true, mostly-true and true.

TABLE II: The LIAR dataset statistics



Dataset Statistics	
Training set size	10,269
Validation set size	1,284
Testing set size	1,283
Avg. statement length (tokens)	17.9
Top-3 Speaker Affiliations	
Democrats	4,150
Republicans	5,687
None (e.g., FB posts)	2,185

The distribution of labels in the LIAR dataset is pretty thorough, it has over 1000 instances of pants-fire and 2,063-2,638 each for other labels.

IV. Feature Generation

Our approach evaluates the performance of models trained on three feature sets:

1. Bag of words model.
2. Bigram Term Frequency-Inverse Document Frequency.

We leverage the Natural Language Toolkit(NLTK) to preprocess the data and perform feature generation. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

The choice for NLTK is that, when the data is self acquired we can train our own POS taggers. In our project also, we train our own POS tagger and this provides the added advantage of being quite fast and lightweight when compared to Spacy's prebuilt version

A. Preprocessing

For convenience, the original 6 labels are converted to 2 labels, wherein half-true, mostly-true, true are mapped to true, everything else is mapped to false. We start the data preprocessing by checking for data balance across all classes, followed by checking for missing or null values. For convenience, we have also stemmed the words using the NLTK stemmer so that plurals or different tenses do not hinder the model's learning and efficacy.

B. Bag of Words Model

Bag of words(BoW) model is a simple way of extracting features from text for use in modeling in machine learning. It represents base-line features in NLP. It extracts the number of times a word occurs in the document, it does this for every word in the document. It is a base-line feature because it discards any order or structure of the words and is concerned with only known words occurring in the document and not where. This model is not sophisticated as it just counts the frequency of the words in the each input and learns which word is most frequently associated with true and which ones with false.

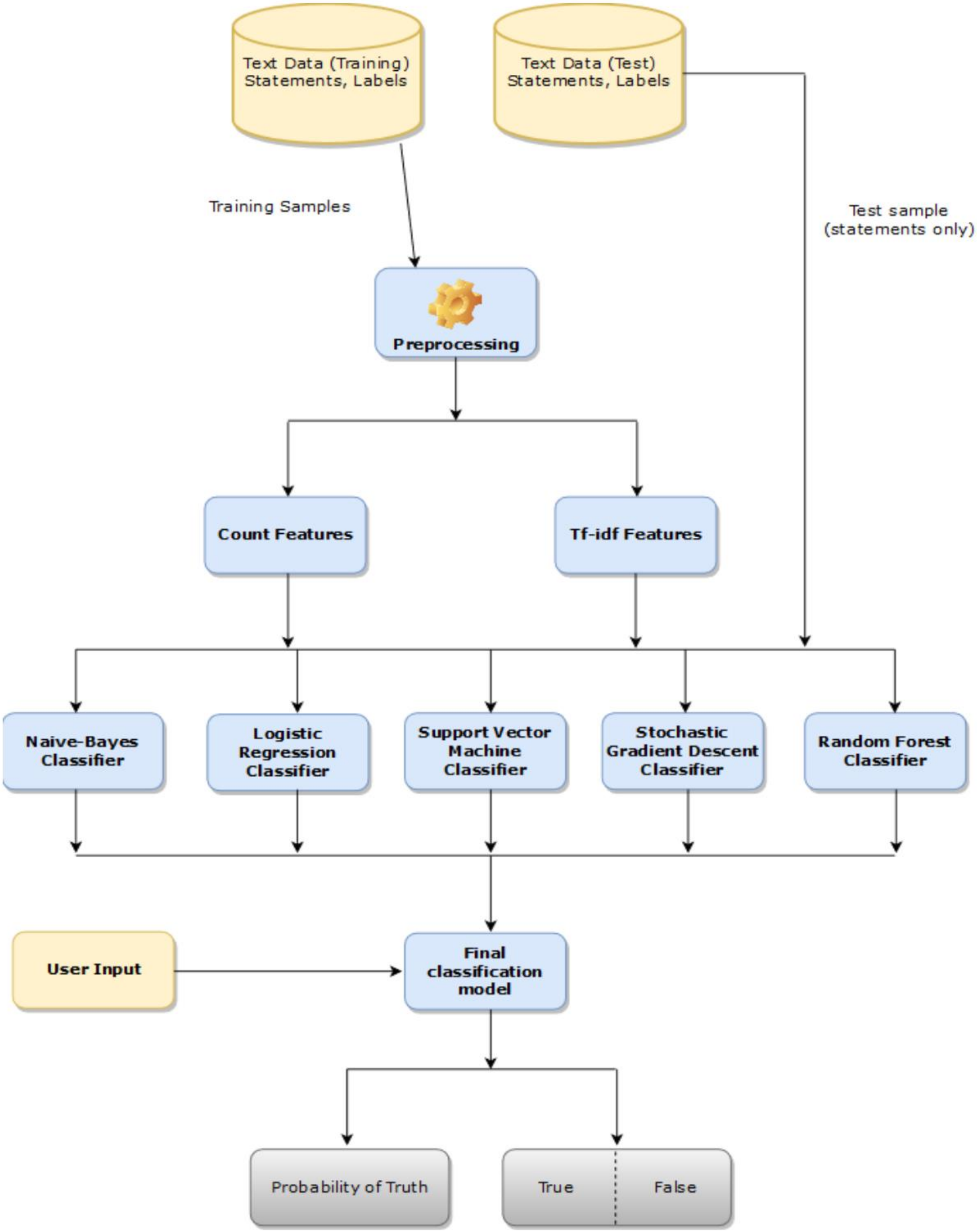


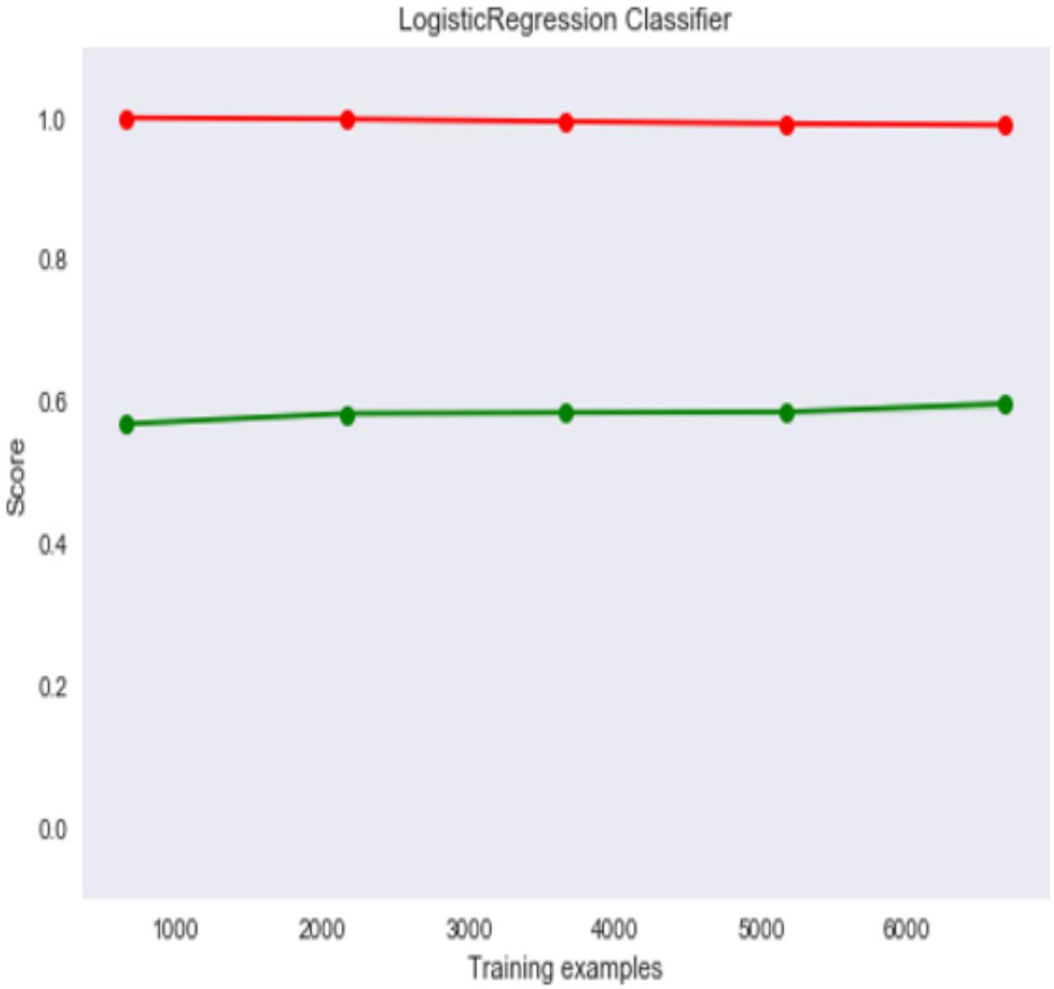
Fig. 1: Pipeline representation.

TABLE III: Average model performance with only BoW and then the average performance with TF-IDF

Model	Precision	Recall	F1-Score
Naïve-Bayes	58.5%	23.3%	66.96%
Logistic Regression	41.0%	22.3%	64.69%
Support Vector Machine	82.9%	25.3%	61.04%
Stochastic Gradient Descent	88.8%	45.3%	64.08%
Random Forest Classifier	81.3%	48.1%	70.26%

Model	Precision	Recall	F1-Score
Naïve-Bayes	66.9%	37.9%	70.11%
Logistic Regression	40.2%	16.1%	71.80%
Support Vector Machine	84.2%	18.4%	67.90%
Stochastic Gradient Descent	74.1%	71.7%	70.50%
Random Forest Classifier	80.9%	44.5%	66.50%

Fig 2.Learning Curve of our best candidate model



C. Term Frequency-Inverse Document Frequency

The next feature that we want to utilize is an improvement on the Bag of Words model, this is the Term Frequency - Inverse Document Frequency. It weighs down the frequency of the words based on how less frequent they are. This is done to make sure that common words like "The" that are so frequent across all documents are penalized. Rare words get more weightage as they provide more context to the article.

Our project is mainly concerned with political articles, the model will be bound to learn the names, institutions and related truths. They will be highly reactive to that particular news domain. A simple solution to this problem can be introduced during tokenization where we render these words invisible, so to speak. This can be overcome by replacing proper nouns with placeholders.TF-IDF with bigrams is done using scikit-learn. A sparse matrix of the TF-IDF is built.

There is much to explore using this feature, that is by selecting different thresholds for selecting terms from the vocabulary. The n in n-grams can be varied for better results and different methods of restricting the vocabulary. All these options are not explored in the project but may be possible roads to explore in the future works on this concept..

V. Modeling and Evaluation

A. Our Pipeline

Once we are done with our preprocessing and feature extraction, we feed the training and validation data to our pipeline with an 80:20 split. Our pipeline tries to fit the data for 5 machine learning based models ranging from a Naive-Bayes classifier to a Random Forest Classifier. We run the pipeline twice, once with BoW feature and on the second time we enhance the BoW model with TF-IDF features. We check the confusion matrix and F-1 score using both features and we see that the enhanced features does well for the accuracy of the models.

Next, the pipeline chooses the 2 best performing models and performs parameter tuning using GridSearchCV method on these selected models to select the best performing parameters for them. Out of these two candidates, the best performing model is saved and are tested on a 20% holdout to evaluate and understand its performance. The final model predicts whether a given news statement is true or not and also provides a truth probability score.

B. Baseline Models for Comparison: BoW model

The baseline model for initial understanding and performance judgement is the BoW model. As we know the BoW model is quite simple in that in just takes into consideration the count of each word that occurs in a news article. The baseline model gives us an understanding of how accurate each of our models are and which ones will be the forerunners in contention for the final model selection, we see that random forest, logistic regression and naive-bayes are performing quite well in just the baseline model and are providing decent F1-scores and confusion matrices.

C. Combining BoW and TF-IDF bi-gram features

The next step demands that we augment our base model with the TF-IDF feature and true to its word, the models all perform well and above to their baseline performances. We can conclude that our best candidate models for the final process are Logistic Regression and Random Forest Classifier. It is a given that logistic regression will perform well with data that is highly dimensional and sparsely populated. To elucidate on the previous statement, they are fore runners in precision and do no sacrifice on high recall and that they will identify fake news effectively and prioritize the news articles if need and objective be.

VI.Conclusion and Future Scope

The results obtained are quite impressive. This project demonstrates the efficiency of a few machine learning models in deception detection. It shows that Term Frequency - Inverse Document Frequency can be an effective feature in news detection and may also hold promise to other NLP applications. We can see that the best performing models are the Logistic Regression and Random Forest models in terms of overall Recall, Accuracy and F1-Scores, using the feature set of TF-IDF. We note that the BoW model does not augment much and can pretty much serve as the base-line model and every other feature can be compared against this for selection. TF-IDF on the other hand is a potent feature in terms of prediction and we tested this even while ignoring name tags and other bias-inducing entities. We need to take this with a grain of salt though, cause we do not know how it behave under rapidly changing news cycles and a huge torrential downpour of just named articles. This can be tested only using a more comprehensive dataset which may be available in the future. Food for thought.

When it comes to future scope of this project, we can see that despite such promising results from our models, especially the Logistic Regression with TF-IDF, there is still scope for a lot of improvement. For one angle on future scope we can explore another feature that is the Probabilistic Context-Free Grammar (PCFG), as they are known to help improve Recall and Precision scores when it comes to NLP applications. They can also function as effective fake-news filters.

Given that TF-IDF is a good performing feature, there may also be the possibility that we are over-fitting to specific topics extremely pertinent to the topical news cycle. There is also the problem of being unable to pinpoint the most important features, because we are taking a vectorized approach to our analysis. All of these issues combined can cap our efficiency and analysis leading to a wide generalization. These are areas that can definitely be looked into in the future and may yield better deception detection and fake news filter than we have now.

References

[1] "Liar, Liar pants on fire": A new benchmark dataset for fake news detection, William Yang Wang

[2] S. Maheshwari, *How fake news goes viral: A case study*, Nov. 2016. [Online]. Available: https://www.nytimes.com/2016/11/20/business/media/how-fake-news-spreads.html (visited on 11/08/2017).

[3] A. Mosseri, *News feed fyi: Addressing hoaxes and fake news*, Dec. 2016. [Online]. Available: https://newsroom.fb.com/news/2016/12/news-feed-fyi-addressing-hoaxes-and-fake-news/ (visited on 11/08/2017).

[4] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, Association for Computational Linguistics, 2012, pp. 171–175.

[5] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[6] V. W. Feng and G. Hirst, "Detecting deceptive opinions with profile compatibility.," in *IJCNLP*, 2013, pp. 338– 346.

[7] V. L. Rubin and T. Lukoianova, "Truth and deception at the rhetorical structure level," *Journal of the Association for Information Science and Technology*, vol. 66, no. 5, pp. 905–917, 2015.

[8] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, *D.* Menczer, and A. Flammini, "Computational fact checking from knowledge networks," *PloS one*, vol. 10, no. 6, e0128193, 2015.

[8] *Opensources*. [Online]. Available: http://www.opensources.co/ (visited on 11/08/2017).

[9] D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?" In *Proceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016.*, 2016, pp. 42–47. [Online]. Available: http://ceur-ws.org/Vol-1568/paper8.pdf.

[10] *Business financial news, u.s international breaking news*. [Online]. Available: http://www.reuters.com/ (visited on 11/08/2017).

[11] Explosion, *Spacy*, Sep. 2017. [Online]. Available: https://github.com/explosion/spaCy (visited on 11/08/2017).

[12] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. Seljebotn, and K. Smith, "Cython: The best of both worlds," *Computing in Science Engineering*, vol. 13, no. 2, pp. 31–39, 2011, ISSN: 1521-9615. DOI: 10.1109/MCSE.2010.118.

[13] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[14] J. D. Choi, J. R. Tetreault, and A. Stent, "It depends: Dependency parser comparison using a web-based eval- uation tool.," in *ACL (1)*, 2015, pp. 387–396.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[17] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1373–1378. [Online]. Available: https://aclweb.org/anthology/D/D15/D15-1162.

[18] Fake news detection on social media: A data mining perspective, Kai Shu Amy Sliva, Suhang Wang, Jiliang Tang, Huan Liu

[19] Nishit Patel, "Fake news detection using python", https://github.com/nishitpatel01/Fake_News_Detection