Synopsis

I have developed a novel way to identify fake news, by combining multiple NLP models trained on different datasets, enriching the result with meta-data and deploying as a chrome extension/ WhatsApp number.

After experimenting with different NLP models, it was decided to settle for a multi-model system:
1) The novelty lies in simultaneously using 3 different models, trained on different datasets, to get our prediction.
2) The 3 different models are: A bi-RNN based stance detection model, an AWD LSTM linguistic model and a pre-trained BERT based linguistic + meta-data model.
3) The multi-model system recorded a 26.67% percentage points increase in accuracy as compared to individual models.

Once an article is received through the WhatsApp number or chrome extension, a novel extraction algorithm is used on it, a combination of Named Entity Recognition and keyword finding algorithms. These extracted words/phrases are used to find 40 articles from a database and stance detect against them. The named entities extracted are used to feed the linguistic + meta-data model and it predicts one of 6 classes, based on the degree of fakeness. Lastly, the linguistic model makes a prediction. The predictions from the model are enriched using a novel combination of network analysis data and databases that contain trust scores for users, media outlets etc. Finally, a novel weighing algorithm is used to output a final prediction, from the inputs of the three individual models.

Our tool greatly improves in terms of accuracy, flexibility and real-world usefulness.

Introduction and Background research

In this study, fake news is defined as a news media that is intentionally and verifiably false (Shu et. al). Fake news is a growing problem. According to research firm Gartner, fake news is expected to overtake real news in terms of volume by the year 2022 [2]. After the US presidential elections, the interest in fake news has been on the rise, even in India.

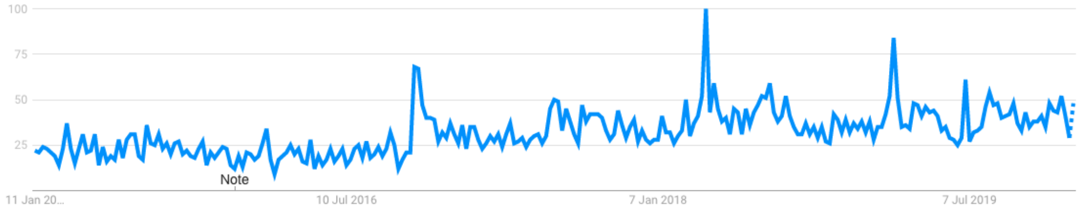


Figure 1. Google trends for the term ’fake news’ In India

A number of computational methods have been experimented in the past (Parikh et al). Figure 2 provides an overview of them.

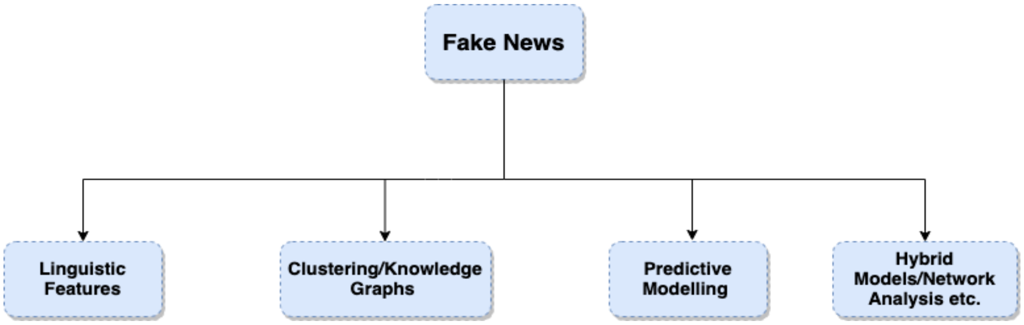

Fake News
Linguistic Features
Clustering/Knowledge Graphs
Predictive Modelling
Hybrid Models/Network Analysis etc.

Figure 2. Different methods used for fake news detection.

Studying the literature and running some experiments to back it up, here are some observations that were made:

- Advancements were made using the above methods but they performed poorly on real world samples (WY Wang).
- This is due to their inability to compare facts and deduce if they match or not (Figueiraa et. al 2017).
- A major pitfall of the current systems is lack of metadata integration (Figueiraa et al. 2017).
- This includes taking into account variables like reputation of the source, reputation of the writer, location, sentiment etc.
- I also wanted to know what kept users from using the current alternatives Figure 3 shows what was found.

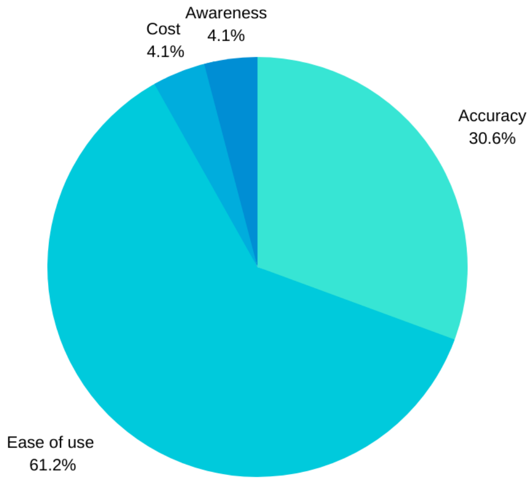


Figure 3. Results of the survey we conducted on reddit, family and friends.

- I deduce that our system will only be successful, if the end user could use it without any friction.

Dataset review

- **Fake News Corpus** - This dataset contains approximately 9.5M news articles. Imbalanced, as it contains majority of fake news. Covers upwards of 745 domains.
- **Fake News Challenge (FNC) dataset** – It does not have any labels as fake or real, instead the text is classified as agree, disagree, unrelated and discuss. This methodology can be extrapolated to identify if a dubious piece of news is in accordance with what well establish news medium are publishing about the topic.
- **Emergent dataset** – Follows a similar methodology to FNC dataset. The labels are split into for, against and observing. A similar technique to FNC is then used to check for the veracity of the claim.
- **Fake or Real dataset** – Contains headlines and body that are classified as Fake or Real.
- **Buzzfeed dataset** – The dataset labels sites based on their political tendency. While not all text in this dataset is false, there are highly biased.
- **PHEME dataset** – Contains tweets that have been retweeted a large number of times. These tweets are then classified as ‘proven to be false, ‘verified as false’ or unverified.
- **LIAR dataset** – The largest human annotated dataset for fake news analysis. Each claim is labelled with one of the six following veracity degrees: pants fire, false, barely-false, half-true, mostly-true and true.
- **LIAR PLUS** – An extension of the LIAR dataset.

From the analysis done, it is believed that the FNC dataset, Fake or Real dataset, Buzzfeed dataset and LIAR PLUS dataset has the highest potential in our study. A system combining models trained on these datasets is believed to be ideal.

Hypothesis

- A combination different text analytics models, each having its unique strengths and weaknesses, would address some of the existing limitations and outperform each of the individual model's performance. Some of the observed limitations of existing models are sensitivity to text length and inability to identify fake news for highly sentimental articles.

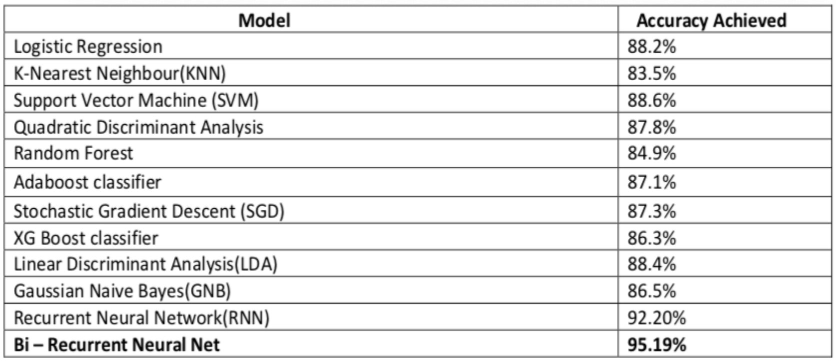
- Using metadata such as, location of origin, author and publisher credibility scores, prior user feedback on similar article, will improve the results further and reflect real-world scenarios.

- Improving the ease of use would increase end user participation and enhance the effectiveness of the existing fake news detection technology

Model Training

Stance Detection

All the training is done on the Fake News Challenge (FNC) dataset [6]. For the stance detection model, several different methods are tried. Table 1 shows the results obtained after training.



Model	Accuracy Achieved
Logistic Regression	88.2%
K-Nearest Neighbour(KNN)	83.5%
Support Vector Machine (SVM)	88.6%
Quadratic Discriminant Analysis	87.8%
Random Forest	84.9%
Adaboost classifier	87.1%
Stochastic Gradient Descent (SGD)	87.3%
XG Boost classifier	86.3%
Linear Discriminant Analysis(LDA)	88.4%
Gaussian Naive Bayes(GNB)	86.5%
Recurrent Neural Network(RNN)	92.20%
Bi – Recurrent Neural Net	**95.19%**

Table 1. Accuracies achieved with different models

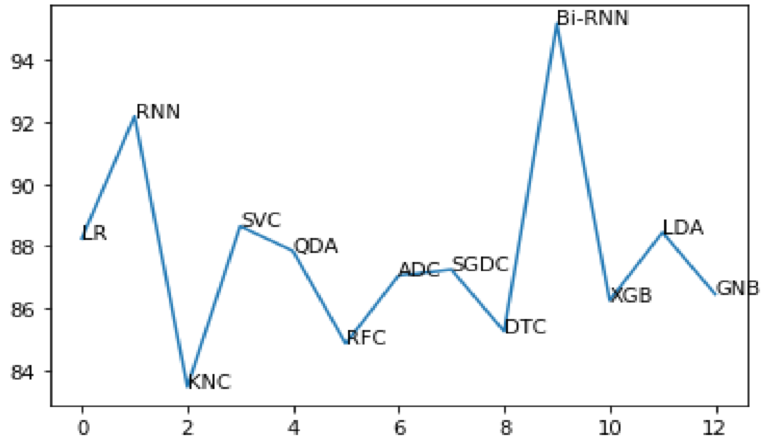


Figure 4. Accuracies achieved relative to each other

From our experiments it was clear that the bi-RNN model got the highest accuracy. Below, we detail more about the model and the hyperparameters used.

- The model is based on the architecture proposed by Borges et al. with modifications such as, KL Divergence between the headline and the body, cosine similarity and other similarity scores.
- The final model is trained with a bi-RNN architecture. All the training was done using Keras with a Tensorflow backend.
- The model also pre-trained on the SNLI[8] and NLI[9] datasets to better improve performance.
- The model architecture is depicted by Figure 6 and the hyperparameters used is depicted by Table 2. The final output of the model is one of four: Agree, Disagree, Discuss or unrelated.

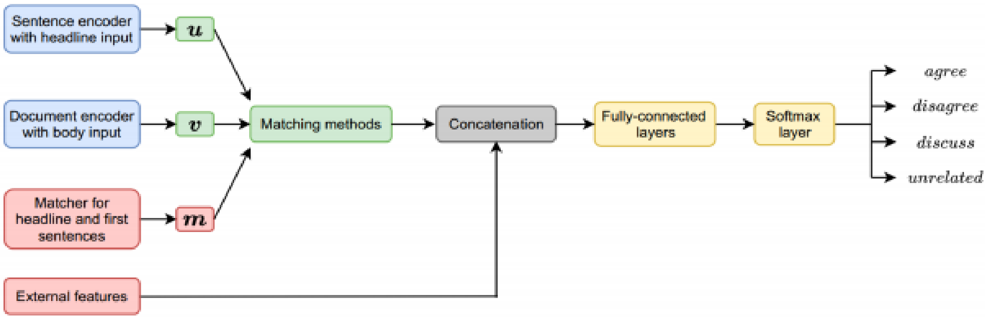


Figure 5. Model Architecture of the stance detection model

Optimizer	Adam
Loss	Categorical Cross Entropy
Metric	Accuracy
Batch size	64
Epochs	100
Early Stop	Enabled

Table 2. Parameters used for the stance detection model

- The performance metric used during training was accuracy. The model achieved an accuracy of 95.19% and an FNC accuracy of 83.56%. FNC accuracy is a weighted accuracy, given by the challenge organizers to benchmark the results.
- My modified implementation marginally improves the model performance as compared to the model proposed by Borges et al.

Linguistics model

The model is trained on the Fake or Real dataset [10]. This dataset was found to be ideal for the task because of the diversity in its sentiments and structure of the data. The steps followed in training the model are as follows.

- I started out with training a language model on the Wiki-text 103 [14] dataset. This was done to improve the 'English' understanding of the model.
- Then the model was trained on the Fake or Real dataset. The model uses a Fast.ai implementation of an AWD_LSTM.

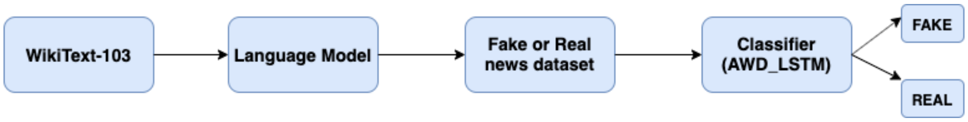


Figure 6. Model architecture of the linguistics model

Optimizer	Adam
Loss	Cross Entropy
Metric	Accuracy
Batch size	32
Epochs	12
Learning rate	1e - 2

Table 3. Parameters used for the linguistics model

- I obtained an accuracy of 99.34% for the classifier, state-of-the-art for the dataset [8].

Linguistics + metadata model

The model was trained on the LIAR PLUS [11] dataset. This dataset was found to be an ideal combination of linguistic data and metadata. Metadata included speaker, affiliation, venue, organization etc. The steps followed in training the model are as follows:

- Using a pre-trained BERT model as the base model and fine-tuning it further for the classifiers was found to be ideal for this task [12].
- Figure 8 shows the model architecture and its predictions.

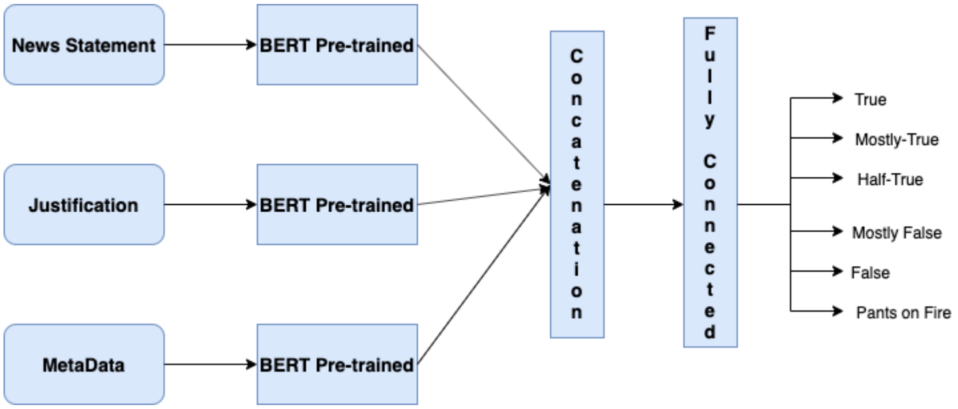


Figure 7. Model architecture of the linguistc + metadata model

Optimizer	Adam
Loss	Focal loss
Metric	Accuracy (6-class)
Batch size	16
Epochs	20
Learning rate	1e - 3

Table 4. Parameters used for linguistic + metadata model

- I achieved an accuracy of 37.2% on this dataset, better than the accuracy mentioned in the paper [9].

System Architecture

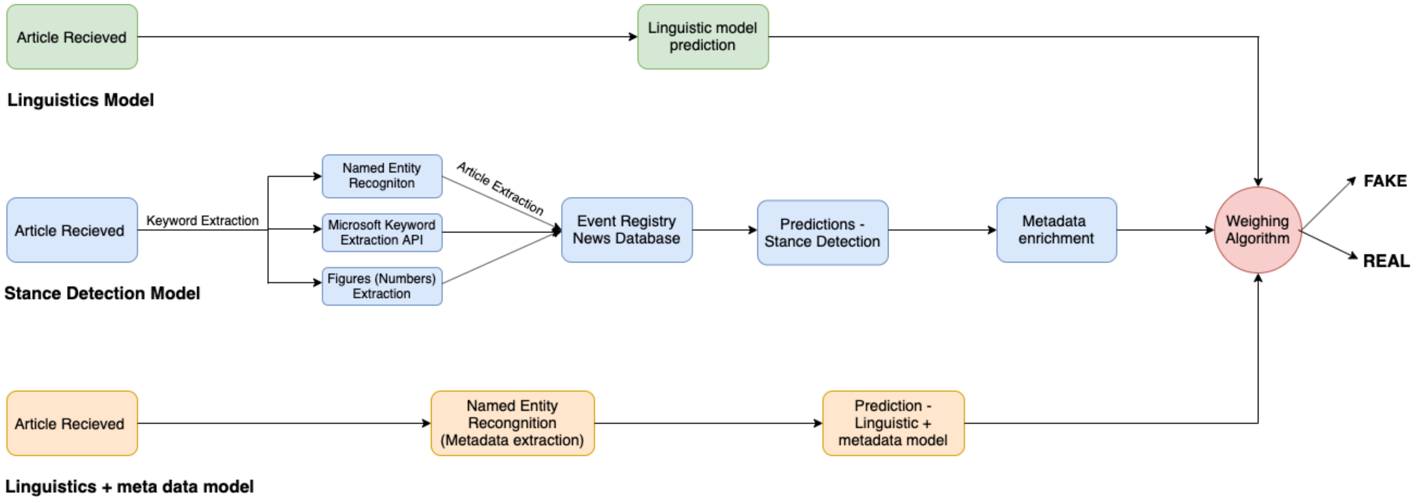

Article Recieved
Linguistic model prediction
Linguistics Model
Named Entity Recogniton
Article Extraction
Article Recieved
Keyword Extraction
Microsoft Keyword Extraction API
Event Registry News Database
Predictions - Stance Detection
Metadata enrichment
Weighing Algorithm
FAKE
REAL
Figures (Numbers) Extraction
Stance Detection Model
Article Recieved
Named Entity Recongnition (Metadata extraction)
Prediction - Linguistic + metadata model
Linguistics + meta data model

Figure 8. Combination and flow of all the models

Figure 9 above describes the processes flow for the three models put together. Once an article is received, it is simultaneously sent to all the three models for prediction. The resultant outputs from each models is then weighed based on a weighing algorithm to arrive at the final classification. The details are as discussed below:

Stance Detection model

- The first step for the stance detection model is to find similar articles to stance detect against.
- This is done by first extracting keywords/phrases using a novel algorithm comprising of Named Entity Recognition (NER) to extract people, organizations, locations etc. , a keyword/phrase extraction API and extraction of numerical figures.
- The above keywords/phrases are used to search for similar articles from Event Registry, a database containing millions of articles. We retrieve 40 articles from each query. This number was decided after intensive experimenting.
- Next, using our stance detection model, each of the article retrieved are classified into one of four categories: agree, disagree, unrelated or discuss.
- Each of the four categories are assigned a weight. If the sum of this weight is postive then the news is real, else it is fake.
- At this step, metadata is incorporated to further improve the performance. metadata used are: credibility score of the source (obtained from our database), a trust score for the writer (crowdsourced from users and independent verifiers), a trust score for the user. Credibility and trust scores are dynamically updated to reflect the latest information available. Usage of metadata, significantly imporived the results of the model.

Linguistic + Metadata model

- Named entity recognition is used to extract from the article information regarding the speaker, place and organizations involved. A trust score is also maintained for the speakers that keep changing based on their performance (how many false claims they present) across multiple articles.
- Finally, the model makes one of 6 prediction depending on the factuality of the article.

Linguistics Model

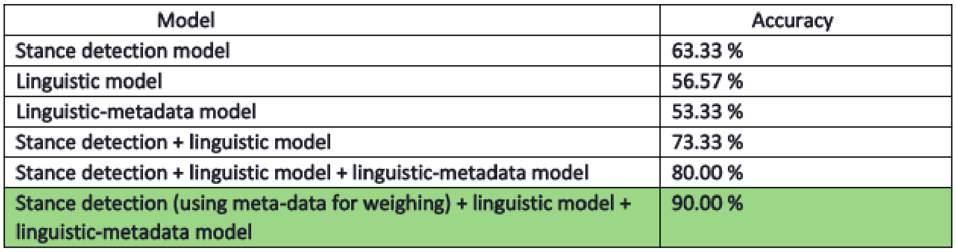
- The output of the linguistic model is used without further modifications as it inherently classifies a given article into Real or Fake.

The output of the above three models namely. Stance detection, linguistic + metadata and linguistic model are sent to the final weighing algorithm to determine the final classification of the article i.e. real or fake

Weighing Algorithm

- The predictions from the three models serves as the inputs to our custom weighing algorithm.
- The algorithm assigns weights based on various factors: length of the article, number of keywords/phrases extracted, number of similar articles found and sentiment present in the article.
- Each of the factors is assigned a weight and they were iteratively improved upon, optimizing for higher accuracy.

All the experiments are performed on the RealNet dataset [13]. The dataset has been released for external evaluation. This dataset is a combination of long and short articles and WhatsApp/Instagram/Facebook posts. It is representative of the real-world and has diversity in terms of length, genre, source of extraction and sentiment.



Model	Accuracy
Stance detection model	63.33 %
Linguistic model	56.57 %
Linguistic-metadata model	53.33 %
Stance detection + linguistic model	73.33 %
Stance detection + linguistic model + linguistic-metadata model	80.00 %
Stance detection (using meta-data for weighing) + linguistic model + linguistic-metadata model	90.00 %

Table 5. Accuracies achieved using different methods

Table 2 shows the results obtained after performing different sets of experiments. Few observations are as follows:

- The individual models struggled since each one was good at some parts of the datasets but struggled in the other parts.
- For example, the stance detection model performed extremely well with long articles but struggled with short articles and social media posts.
- Combining the models along with the metadata the highest accuracy of 90% was achieved.

- From our research, it is clear that a combination of models, trained on different datasets performed much better than single models.
- The dataset, code and weights are released on Github[14] for reproducibility.

Result

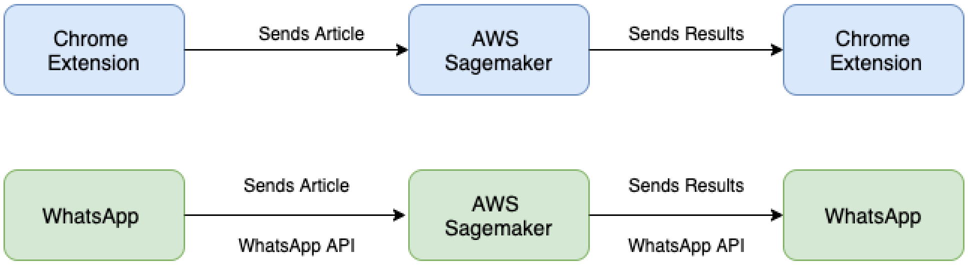


Figure 9. Deploymeny of the models.

- The model is deployed both as a chrome extension and as a WhatsApp number.
- The chrome extension automatically retrieves every article the user is reading and sends it via an API endpoint hosted on AWS Sagemaker which returns the results.
- The models are also accessible through a WhatsApp number. The user has to just forward the news/post he/she wants to be fact-checked which is then transmitted to AWS Sagemaker via the WhatsApp API which then returns the results.

Both of the deployment methods are highly accessible and easy to use

Novelty

- Used new features to improve results of the stance detection model.
- Achieving state-of-the-art results using a pre-trained language model for the linguistics model.
- **Presenting a novel way to combine multiple models such that they complement each other, significantly improving results.**
- Integrations of metadata leading to improvement in accuracy and flexibility of the model.
- Unique deployment methods - Chrome Extension/WhatsApp number.

Ongoing and future work

- Creating a new dataset for stance detection that is more representative of the real world. (***Ongoing***)

- Expanding the RealNet dataset. (***Ongoing***)
- Introducing new models to the multi-model system. (***Ongoing***).
- Creating a dataset and training a custom deep learning model to extract keywords and phrases. (***Future***)
- Using logistic regression to optimize weights in the weighing algorithm. (***Future***)
- Increasing the use of metadata in the system and adding more network analysis components . (***Future***)

Conclusion

This study successfully proves the validity of the first two out of the three hypothesis, which are improving model performance through a combination of existing models and incorporating metadata to further improve the performance. The new approach lead to a performance improvement, when tested on a real-life representative dataset, of 26.67% as compared to the best performing individual model.

The work on setting up the IT infrastructure to test the third hypothesis (increased effectiveness of the fake news detection system by improving the ease of use) is in progress and is expect to be rolled out for user testing in the next 45 days.

References

1. [Shu et al] Shu, Kai & Sliva, Amy & Wang, Suhang & Tang, Jiliang & Liu, Huan. (2017). Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter. 19. 10.1145/3137597.3137600.
2. Panetta K, (2017 Oct 17) Gartner Top strategic predictions, 2018 and beyond, retrieved from www.gartner.com/smarterwithgartner/gartner-top-strategic-predictions-for-2018-and-beyond
3. [Parikh et al] Parikh, Shivam & Atrey, Pradeep. (2018). Media-Rich Fake News Detection: A Survey. 10.1109/MIPR.2018.00093.
4. [WY Wang] Wang, William. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. 422-426. 10.18653/v1/P17-2067.
5. [Figueira et al] Figueira, Álvaro & Oliveira, Luciana. (2017). The current state of fake news: challenges and opportunities. Procedia Computer Science. 121. 817-825. 10.1016/j.procs.2017.11.106.
6. Fake News Challenge dataset, (2016 Dec), retrieved from www.fakenewschallenge.org, accessed 3 September 2019

7. {Borges et al] Borges, Luís & Martins, Bruno & Calado, Pável. (2018). Combining Similarity Features and Deep Representation Learning for Stance Detection in the Context of Checking Fake News.
8. Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
9. Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. arXiv preprint arXiv:1704.05426 (2017).
10. Fake or Real https://github.com/GeorgeMcIntire/fake_real_news_dataset, accessed 20 November 2019
11. Alhindi, Tariq & Petridis, Savvas & Muresan, Smaranda. (2018). Where is Your Evidence: Improving Fact-checking by Justification Modeling. 85-90. 10.18653/v1/W18-5513.
12. Singh M, retrieved from github.com/manideep2510/siamese-BERT-fake-news-detection-LIAR
13. RealNet dataset, github.com/cabhijith/iris
14. github.com/cabhijith/iris