Linguistic-feature based fake news detection and classification

Machine Learning final report

Btech CSE, VI sem

Machine Learning

CS351

Name	Roll. No.	Reg. No.	Email	Phone No.
Arnav Santosh Nair	**181CO209**	**181325**	arnavnair13@gmail.com	**7899361099**
Hardik L Harti	**181CO220**	**181199**	hardikharti111@gmail.com	**8867795969**

Table of Contents

1. Abstract

We have seen the evolution of media and technology in every field. We know that social media is used as a dominant source of news distribution in recent times. The world's preeminent decisions such as politics are acclaimed by social media to influence users for enclosing users' decisions in their favour. However, the adoption of social media is much needed for awareness but the authenticity of content being put up on these platforms always remains under question. To try and find an answer, we present a project based on a research work that proposes a solution to fake news detection and classification. In the case of fake news, content is the prime entity that captures the human mind towards a trust for specific news. Therefore, a linguistic model is proposed to find out the properties of content that will generate language-driven features. This linguistic model extracts syntactic, grammatical, sentimental, and readability features of particular news. Language driven models require an approach to handle time-consuming and handcrafted features problems to deal with the curse of dimensionality problems. Therefore, the neural-based sequential learning model is used to achieve superior results for fake news detection. The results are drawn to validate the importance of the linguistic model extracted features and finally combined linguistic feature-driven model can achieve the average accuracy of 78% for fake news detection and classification. The sequential neural model results are compared with machine learning-based models and LSTM based word embedding based fake news detection models as well. Comparative results show that the linguistic-features based sequential model can achieve comparable evaluation performance in discernibly less time.

2. Introduction

In recent times, social media has experienced fast and extensive growth. News from social media is prevalent these days, and people rely on social media for the latest updates, trending stories, and mutual information. This demonstrates the lack of professional competence with traditional news platforms nowadays. The major concern with this growth is that we cannot distinguish fake news and anomalous information from truthful information. It has become an obstacle for advanced computing technologies to deal with the variety of information and different meanings of the context. On the other end, a major part of the social media platforms is flooded with fake news that affects the news ecosystem, people's opinions, and stock markets. To tackle this problem researchers are working on a variety of methods. One such model has been discussed further.

This study aims to create a linguistic feature-driven deep learning model for effective fake news detection and to measure the impact of extracted linguistic features based on the comparative outcome with the deep learning model. Traditional approaches to fake news detection resort to Sequence Models like LSTM or even more advanced models like Transformers due to their superior performance when working with sequence-based input. However, in cases where data is

scarce, these models struggle to perform a satisfactory job. This study instead builds an Artificial Neural Network that works on specifically extracted linguistic features and produces superior results with smaller datasets. Lastly, we implement an SVM model and an LSTM-model on the same dataset for the sake of further comparison.

3 Work Done in the Paper

3.1 Feature Sets

The linguistic features used in the paper can be classified into 4 major feature classes as:

1) **Syntax-based Features:** The syntax-based features are composed of several linguistic dimensions which satisfy a specific pattern to classify fake news. This phase consists of the extraction of fundamental evidence - count statistical evidence, sentence sentiment evidence, and grammatical property evidence.
    - **Character Count:** Total no. of characters without spaces
    - **Word Count:** Total no. of words in a given sentence
    - **Uppercase Word Count:** Count the number of uppercase words in a given sentence
    - **Stop Word Count:** Count the total no. of stop words in a given sentence
    - **Title Word Count:** Count the number of words in a given title
    - **Lexical Density:** Ratio of lexical items to the total number of words
2) **Sentiment-based Features:** The sentiment is put forth on writing a news article that relies on decision-making factors in the process of classifying the news into fake or not fake.
    - **Polarity:** It refers to positive and negative statements. It lies in the range of [−1,1]
    - **Sentiment Score or Subjective Score:** Expressing an opinion, views, or a person's feelings. It lies in the range of [0,1]
3) **Grammatical Features:** The grammatical features are an important factor that is extracted through parts of speech (POS) tag evidence features. These features are designed to apprehend the deceiver cues in writing style to differentiate fake news.
    - **Noun Count**
    - **Verb Count**
    - **Adjective Count**
    - **Pronoun Count**
    - **Adverb Count**
4) **Readability Features:** Readability is a measure with which a reader can apprehend the written text. In simple language, the readability of a text relies on its content material i.e. the complexity of its vocabulary and syntax of its content.
    - **Flesch Reading Ease:** Evaluate the difficulty pattern of the written text.

- **Automated Readability Index:** Determines the level of understandability of English text.
- **Gunning Fog Index:** Observing the writing problem.
- **Flesch-Kincaid Score:** Analyze the level of the written text.
- **SMOG Index:** Understand the formation of writing content
- **Linsear Write Formula:** Developed for the United States Air Force to calculate the readability of their technical manuals.

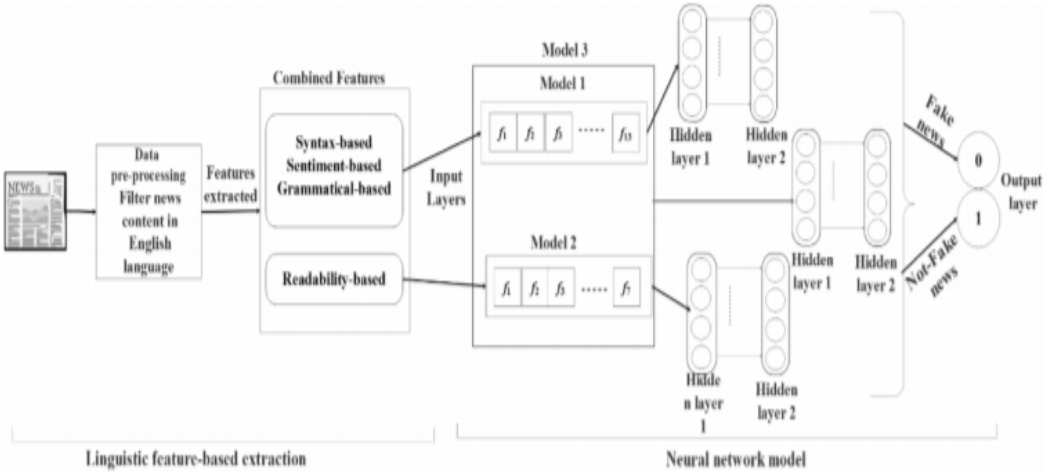
Feature	**Formula**
Flesch Reading Ease	206.835 − (1.015 * ASL) − (84.6 * ASW)
Automated Readability Index	4.71(xcc/xwc) + 0.5 (xwc/xst)
Gunning Fog index	0.4(ASL + PHW)
Flesch-Kincaid score	3 + square root of Polysyllable Count
The SMOG Index	(0.39 * ASL) + (11.8 * ASW) − 15.59
Linsear write formula	[(100 − 100*n(wsy<3)/ nw) + (3*100(n*wsy≤3)/nw)] / (100*nst /nw)

3.2 Dataset

Description of the dataset:

Dataset name	**News categories**	**News Count(Title and Description Each)**
Buzzfeed political news	Fake	48
	Real	53
Random political news	Fake	75
	Real	75

3.3 Models and Architecture



3.3.1 Models

We will be separately training and testing the data on three separate models based on the use of feature sets as shown in the figure:

a. **Model 1**: This will evaluate the three important types of linguistic features: Syntax based, Sentiment based and Grammatical evidence features. The feature extraction stage has been set up to extract a total of 13 features from these feature classes. Thus, Model 1 works on this feature set.
b. **Model 2**: This will evaluate the readability features, consisting of a total of 6 features from the extraction stage.
c. **Model 3**: This model behaves like a superset of the ones discussed so far. It works on all 19 features extracted from all the different feature classes, so as to obtain a holistic look at the input data. Thus, it works with 19-dimensional feature vectors.

3.3.2 Neural Network Architecture

- The Neural network architecture is the same for all the models.
- Every model has only **2** hidden layers, each with **4** units.
- The **ReLu** function will be used as the activation function at every hidden layer.
- The final classification is obtained using a **Sigmoid** activation function.

3.4 Methodology

- **Stage 1:** This stage involves sending the dataset into the preprocessing stage. Here, the

application will begin by tokenizing and removing all punctuations from the provided tuples. This pre-processed data is now ready for feature extraction.

- **Stage 2:** This stage extracts the aforementioned linguistic features from Section 3.1. These extracted features are compiled into their own input object.
- **Stage 3:** This involves the actual Neural Network architecture. It will accept the input object of extracted features and run on them to perform the desired classification. The differences in feature classes accepted in this stage produce Models 1-3 as described in Section 3.3.1.

  Finally, the performance of the 3 models in terms of their accuracy is compared so as to establish the best set of feature classes for this task.

4. Additional Dataset

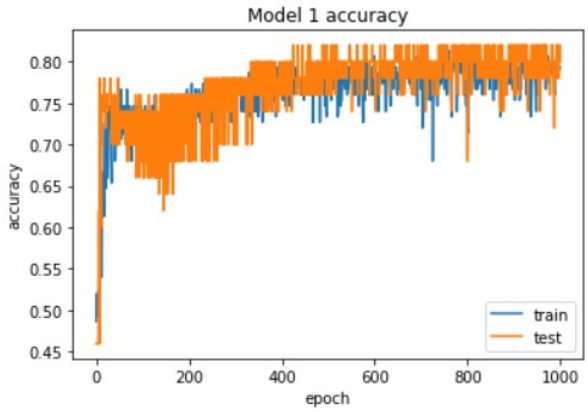
The datasets used in the paper are considered quite small when working with Neural Network architectures. Thus, to ensure that the results obtained in the paper are not simply a fallout of scarcity of data, we repeat the same task with a new dataset on all 3 models and compare the results. For this task, we used a Fake New Dataset from Kaggle. The dataset has a total of 20800 tuples with the following attributes:

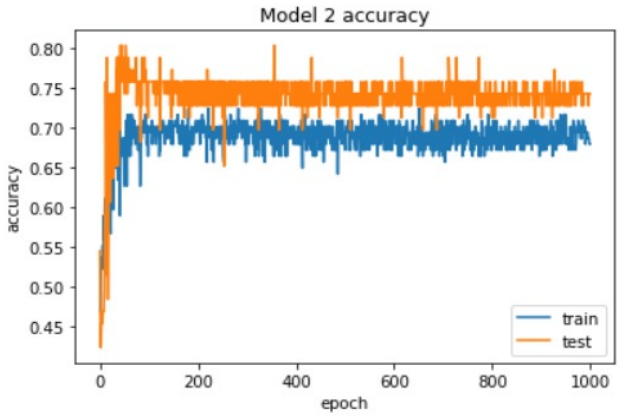
- **id**: unique id for a news article
- **title**: the title of a news article
- **text**: the text of the article; could be incomplete
- **label**: a label that marks the article as potentially unreliable, must be either:-
  1: fake
  0: real

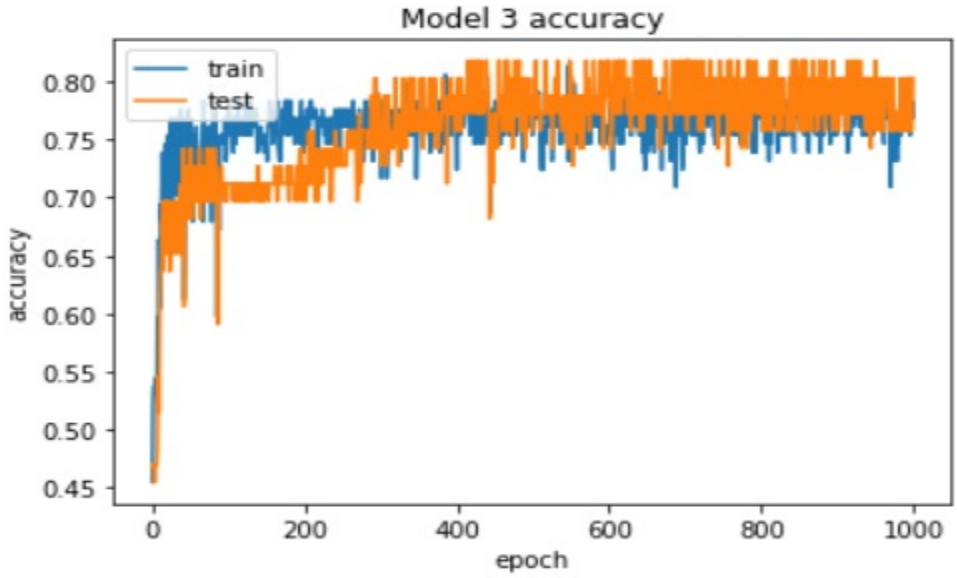
5 Results

5.1 Original Dataset

The following results were obtained on the 2 original datasets mentioned in the paper, namely - Buzzfeed political news and Random political news. Each model was run on Mini-Batch Gradient Descent with a batch size of 10 and number of epochs as 1000. The results were as follows:





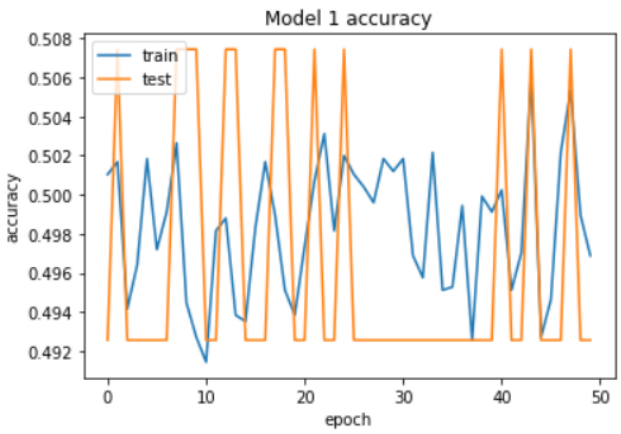


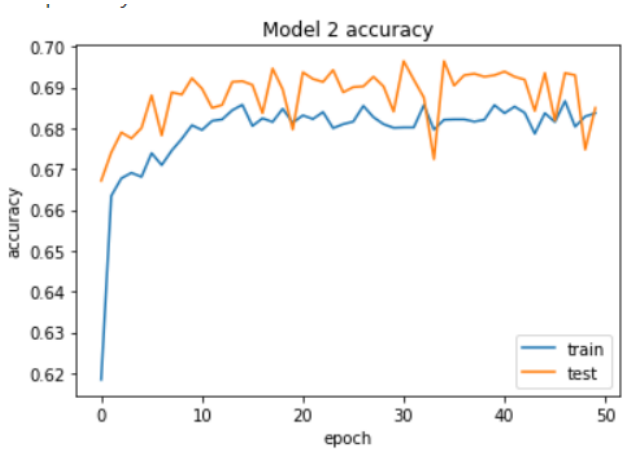
The average accuracies obtained for the respective models were:

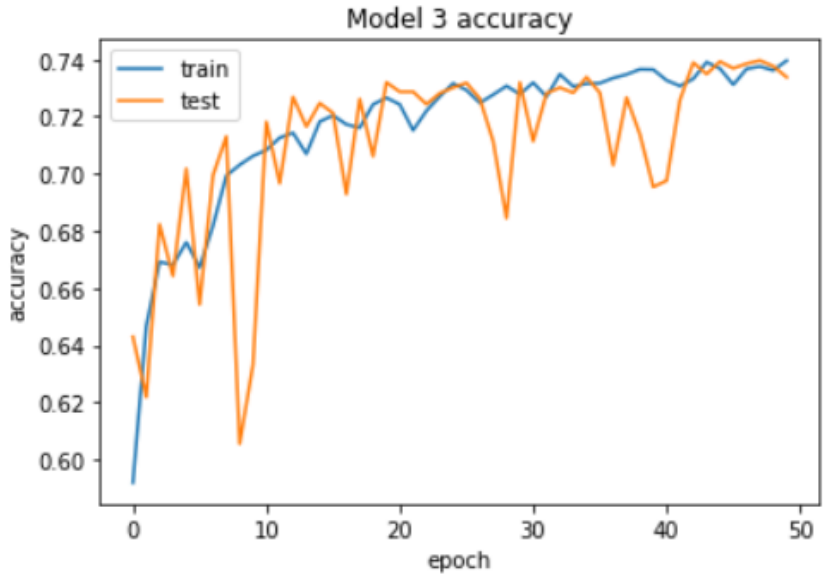
- Model 1 - 74.2%
- Model 2 - 70.1%
- Model 3 - 78.9

5.2 New Dataset

This refers to the Kaggle dataset mentioned in Section 4. Each model was again run on Mini-Batch Gradient Descent with a batch size of 10 and the number of epochs as 50. The results obtained with this in terms of performance were nearly consistent with the original dataset test.







The average accuracies obtained for the respective models were:

- Model 1 - 50.2%
- Model 2 - 69.8%
- Model 3 - 74.9 %

6 Comparative Case Studies

6.1 Motivation

While this paper aimed to compare only the 3 models mentioned before, we felt it necessary to compare with some different architectures altogether, so as to establish the relative performance of this one.

Additionally, the paper makes claims of the relative inferiority of models like the LSTM model when compared on the same task. However, no implementations and results are shown to back this claim. Thus, the following work covers our extension of the ideas put forward by the authors of this paper and our contributions to the concepts of the paper, so as to either back or refute these claims. Lastly, we put forth possible explanations for the relative performance of the discussed models.

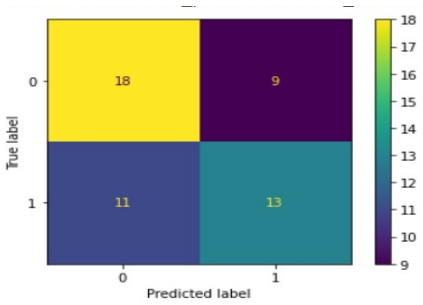
6.2 Support Vector Machines

Despite the general superiority of Deep Learning models over traditional Machine Learning models, we felt it was worth comparing with a strong ML model so as to establish a lower bound on the performance of the 3 models, instead of only focussing on their relative performances. The candidate chosen for this study is an SVM with a Gaussian RBF kernel as the similarity function, as:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

The combination of a relatively small number of features along with the property of SVMs being large-margin classifiers justifies a deeper look into this comparison. The model was run for The results obtained were as follows:

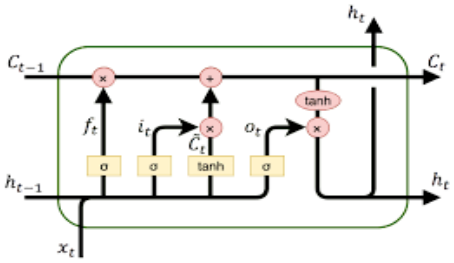
```
Accuracy score: 0.6078431372549019
Report              precision    recall  f1-score   support

           0       0.62      0.67      0.64        27
           1       0.59      0.54      0.57        24

    accuracy                           0.61        51
   macro avg       0.61      0.60      0.60        51
weighted avg       0.61      0.61      0.61        51
```



As observed, the SVM massively underperforms at this task with the same dataset, despite the training conditions being the same. Also, the low accuracy cannot simply be chalked up to skewed classes, as indicated by both the dataset description, as well as the F1-scores obtained. This establishes the superiority of all 3 Neural Network models discussed in the paper over traditional ML models in this regard.

6.3 Long Short-Term Memory

While the last study compared the NN models to theoretically inferior models, this study aims to do the exact opposite. Long Short-Term Memory(LSTMs) are superior versions of the basic Recurrent Neural Networks(RNNs) with 3 gates to help deal with the Vanishing Gradient Problem, thus allowing the model to perform better with long-range dependencies.



They are the most powerful candidate for sequence input-based tasks, outside of the Transformer architecture. The paper claims that their model outperforms an LSTM model at this task with their mentioned dataset, without providing any results to back this. Thus, this study aims to compare the performance of the 2 and provide possible explanations for the results.
The pre-processing of data for this model involves:

- Tokenization
- Padding the sequences
- Encoding

The architecture used for this model was:

- **Embedding Layer:** This layer trains an embedding matrix to learn word embeddings of the input language. These embeddings can be pre-trained after learning from an algorithm like Word2Vec or GloVe. For this study, we chose not to upload pre-trained embeddings.
- **LSTM Layer:** A 128-cell LSTM layer that accepts the input sequence and produces the output activations.
- **Dense Layer:** Accepts the LSTM activations and runs it through a Neural Network layer, ending with a ReLu activation.
- **Output Layer:** Accepts the Dense Layer's activations and performs Sigmoid activation for the final classification.

The model was trained with a batch size of 64 for 30 epochs, owing to its inherently large training time. The results obtained with this were:

*The model gave an accuracy of **70.58%.***

We now look into some possible explanations for these results, the most likely being:

- Due to the relatively small amount of data, the LSTM couldn't sufficiently train its massive number of coefficients. This produces a significant difference in performance, as LSTM have far more coefficients than simple NNs.
- As the word embeddings used are trained alongside the weights, the model likely cannot learn good embeddings, curbing its performance.

As clearly indicated by the results, the claims in the paper have been validated. Thus, in the setting of scarce data, the linguistic feature-based model described by the paper is able to outperform an LSTM-based model.

7 Possible Improvements

The training set performance of all 3 models leaves room for improvement. Thus, high bias could be a possible issue here. Three ways to tackle this are:

- Find more linguistic features for the model.
- Create an integrated model with features other than linguistics
- Increase the complexity of the model(number of layers, number of activations, etc.) so as to increase the complexity of the resultant hypothesis used.

8 References

- Original Paper - Linguistic feature based learning model for fake news detection and classification
- All Discussed Implementation - ML Theory Project 181CO209-181CO220