Fake News Detection

Sheena Gaur

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

6th September 2019

Abstract

The spread of false information is much worse now than ever before due to the ease with which people can acquire and upload information into social media sites. This false information constitutes what we now call 'fake news'. While various websites like FullFact, Snopes and PolitiFact provide an environment for people to confirm whether any news is true or not, these sites suffer from the problem of having to manually check the credibility of any news source. Through this research project, we aim to analyze an existing state of the art fake news detection model that provides an automated approach to this problem. This model uses a supervised Deep Learning approach to classify any news article as being fake or not. Furthermore, we also create a novel model that includes the best features provided by the state of the art model but also overcomes its weaknesses. It exploits a multi-dimensional self-attention scheme. This new model called the Hybrid model has shown a performance increase more than 20% than its parent on the same dataset. In the future, we hope to improve the model further while reducing the shortcomings found in this during research.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: Sheena Gaur Signature:

Acknowledgements

I would like to thank Dr. Joemon Jose for all his support and guidance throughout the dissertation. I would also like to thank my family for their encouragement during this time.

Contents

Chapter 1 Introduction

1.1 Motivation

With social media, it has now become much easier to spread misinformation or fake news. Social media, which had initially been developed as a way to communicate and meet with people around the world, is now a hub of disinformation, hoaxes and propaganda. This is because there are no restrictions in place for publishing or distributing content on these sites. With the unguarded spread of such information, it sometimes may trickle down into mainstream news and be reported to the masses. Spreading of misinformation is not something that has started happening recently but one whose impact is more far-reaching now than before. It has been suggested that exposure to false stories prior to the 2016 US election may have affected its results [Alloctt et al., 2017]. The impact of such incidents has seen a loss of peoples trust in mainstream media (as can be seen in Figure 1) and social platforms as a source of information [Saad, 2017] and causing Facebook, among other websites, to change their policies in regards to publishing news content on their site [Alloctt et al., 2019].

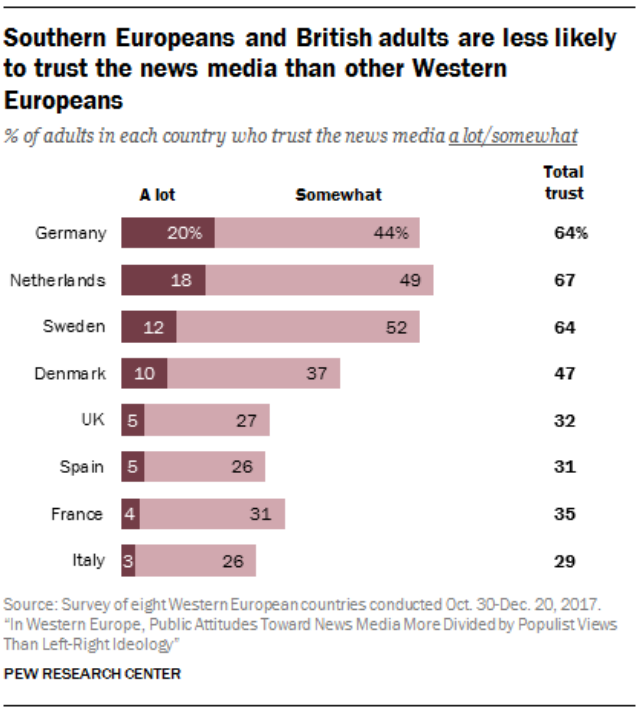

Figure 1: A research done by PEW Research Centre in 2017 showing trust in news outlets by Western European countries [Silver, 2018].

To combat this problem, many independent fact-checking websites such as PolitiFact, Snopes and FullFact are now that verify the accuracy of news content but the fact-checking provided by these websites is done through a manual process by volunteer journalists. While the presence of these websites is a great addition to combat the spread of disinformation, it cannot be denied that it is a long and tedious task. A solution to this problem is the use of automated fact-

checking algorithms that use a supervised learning approach to make this process less cumbersome.

1.2 Purpose

Various deep learning algorithms are currently available that provide a content-based analysis of the source information to classify the news as fake or not. Of these algorithms, the DeClarE model [Popat et al., 2018] stands out due to its unique approach of also considering external sources as additional features for the classification task. It takes into consideration the claim, the articles that contain the claim, the sources of both the articles and the claim which allows the algorithm to aggregate the content from multiple sources to reach a verdict. While this model is at the forefront of the research into fake news detection, a major drawback to this algorithm lies in its structure and complexity. Hence, we investigate the creation of a new model, explaining Self-Attentive Sentence Embedding [Lin et al., 2017] as a representation of text, and addressing the structural issues of the DeClarE model. This newly generated hybrid model is compared against DeClarE using the same datasets and evaluation metrics.

1.3 Summary

In this chapter, we gave a brief introduction to the topic of fake news and the need for automated fact-checking algorithms. The next chapter will give an in-depth analysis of the algorithms selected for this project as well as the objectives of the research. Chapter 3 introduces the methodology of construction and implementation of the models including the challenges faced during the process. Chapter 4 is the evaluation chapter and it documents the results and provides a thorough comparison of the two models. In the last chapter, Chapter 5, we talk about the overall results of this research and prospects for future improvements in this research.

Chapter 2 Analysis

In this chapter, we shall set up the objectives for the research project as well as go into details about the algorithms selected for the project.

2.1 Background

As stated in the previous chapter, fake news has become a prominent issue with the widespread use of social media platforms. The ease of content sharing has allowed any sort of information, factual or otherwise, to propagate through the populous. Fact-checking websites do provide relief by giving credibility tags to news but this is a manual process requiring involved parties to comb through many articles and sources in order to ascertain the credibility of the information [FullFact]. A manual approach is a time consuming method as the people involved have to be meticulous in their research into the topic before providing any credibility label to an article. This process can be made much easier through automation that uses Deep Neural Networks.

2.1.1 Deep Learning Models

In the domain of Deep neural networks and its use in fake news detection, various models are available that handle the classification differently. These text-based models mainly focus on the semantic relationships among the text of the article and its structure. It has been stated that prior work done in this field assume that the claim follows the same structure of subject-object-predicate [Popat et al., 2016] while the writing type of each person is different and may not match the same structure. Some other models [Rashkin et al., 2017] use labeled data from fact-checking websites like PolitiFact to train their model but they do not perform any feature modeling. Other works use social media networks to study the news claims [Zhang et al., 2018] with a focus on Twitter as the main source of information.

A concern with these varied models is that they do not consider the sources of the claim as a part of the modeling. This leaves out a lot of crucial information about the origin of the news. Some misinformation could have its origin in known hoax sites or are said by individuals known for their satire. When such sources information is provided to a human, they can easily disregard the claim as fake as they know the history of such individuals or sites. But that is not the same for a supervised learning model. A novel approach to overcoming this limitation of these models is to introduce more input to the model other than just the article text. DeClarE (Debunking Claims with Interpretable Evidence) [Popat et al., 2018] is a neural network model using natural language format of the claims to assess their credibility and exploits the additional information provided by not only the article but also its sources. It includes the news topic itself as a part of the input which is then compared with the article in question to classify its credibility. Due to this novel approach to fake news classification, the DeClarE model is selected for this project.

2.2 DeClarE Model

The model proposed by Popat et al. (2018) is a multi-input model that bases the credibility of the news on not just the claim (new topic) and the article itself but also on the external sources of the article. An advantage that this model has over existing approaches is that it does not require any manual intervention like feature engineering for processing the input and also that it takes into consideration external sources for the news.

An article context can be very large for a model to make any sense of so the DeClarE model uses bi-directional LSTM (Long Short-Term Memory model). It is a gated RNN that allows the model to accumulate information over time [Goodfellow et al., 2016]. An important aspect of LSTM is self-loops, that allows the gradient to easily flow through and prevent the vanishing gradient problem. A bidirectional LSTM is an LSTM that runs the input both ways. This allows the model to have information both about the past and the future through a forward and backward pass. Hence, the model is able to understand the context information better than normal LSTM.

The second component of the model is an attention mechanism which is employed to focus the attention of the model towards certain common features of the input. This makes the credibility predictions interpretable and user-friendly. This mechanism is used on a combined input of the claim and article to get a better article representation. The structure of the model is shown in Figure 3.1. The visualization of the article attention weights in the research shows a higher concentration of weights on claim related words along with credibility related terms like 'barely true' and 'revealed' etc. The model is versatile enough to be used both for the classification of news and to get a credibility score through regression. This is evidenced by the author of the paper testing the model on four datasets, three of which are used for a classification task and the fourth one for a regression task. The results of the research show that the model performs better than its peers on most of the datasets. The structure for the DeClarE model is discussed in detail in Chapter 3.

2.2.1 Drawbacks of the Model

During the process of model analysis and recreation, some drawbacks were found which are listed below:

- **Input Processing**: The model divides the input context into two parts. One part processed by the Attention mechanism and the second part by just the bidirectional LSTM. The attention mechanism uses the combined article and claim input to focus the attention weights on those words common to both the inputs. In order to handle the large input size, the authors take an average of this combined input but certain nuances of the input may be lost. Using LSTM instead would have been better to generate the initial weights for the input. Due to its ability to retain information, input data would not have to be summarized as an average but rather each iterative hop will assign weights to semantically meaningful words related to both the article and claim. But in the DeClarE it is applied separately on just the article.
- **Reproducibility**: As no source code had been provided by the authors so for this research, the model had to be recreated. The paper explains the

mathematical nuances of the creation of the model but not how the authors implemented it in code. The structure had many complex layers owing to the multiple inputs and operations needed to connect them. The challenges faced in this process are highlighted in the next chapter.

- **Data Pre-processing**: During the experimentation process of the DeClarE model evaluation, the input data was prepossessed to remove those articles that have the maximum relevance score, i.e., those articles that share a similarity with the claim. But by removing this data, the ability of the model to handle real-life data is reduced [See Appendix A]. Hence, it can be said that the model is less robust.
- **Penalization**: The article input is used twice in this model which is then later combined. This may cause a redundancy problem in the weights but the authors have not used any penalization term to accommodate this.

2.2.2 Experimental Bias

With further analysis of the datasets, we suspect an experimental bias could have happened. The DeClarE model does its evaluation using the PolitiFact and Snopes datasets among others. These datasets are structured such that a claim could be associated with multiple articles hence having the same claim occur multiple times. When we do a dataset division, for example, an 80-20 training and test set split, it has the same claim being seen both in the training and test set causing the results to be biased. This issue will be discussed further in this research.

2.3 Self-Attentive Sentence Embedding Approach

Research was done to find a compatible way to handle the language processing of the input in a way that can combine the divided flow of the DeClarE model but still maintain the benefits of the multi-input structure. The model proposed in the paper 'A Structured Self-Attentive Sentence Embedding' [Lin et al., 2017] was selected for this reason. This model provided the benefits of being easier to reproduce and provided support for penalization hence overcoming two of the drawbacks of DeClarE.

The Self-Attentive Sentence Embedding is a model proposed for language modeling, specifically for sentiment analysis. This model uses a self-attention mechanism as an extra source of information for models that take only a single input. Self-attention is also called intra-attention, which enhances focus on the input to produce a representation of it with aggregated focus on the central features [Vaswani et al., 2017]. This, when combined with a bidirectional LSTM, allows multiple representations to be extracted from the input sequence at each hop to create a matrix representation of the input sequence vector. The paper states that this arrangement relieves the stress of long-term memorization from LSTM, as the attention mechanism has access to hidden states of past time-steps which are combined in each hop into a single representation. Another novel contribution of this paper is the penalization term. When the bidirectional LSTM is combined with the self-attention mechanism, it generates the sentence embedding for the sequence. This embedding may be prone to redundancy problem due to repeated attention on the same sequence. This penalization term takes the squared Frobenius norm of the self-attention weights which is then minimized along with the loss. The Frobenius norm, also called the Hilbert-

Schmidt norm, can be defined as the square root of the sum of squares of its matrix entries [Ford, 2015].

$$||A||_{\mathrm{F}} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2}$$

Figure 2.1: General formula for Frobenius norm [ScienceDirect]

The attention weight and its transpose supplied to the norm have values summing up to 1, hence they are assumed to be probability masses. The penalization term encourages the diagonal elements of the term to approach 1. This forces the model to focus attention only on certain terms at a time. Due to these benefits provided by the model, it is selected to replace the attention mechanism in DeClarE and generate a new model.

2.4 Proposed Model

While the Self-Attention model does provide various benefits, it was created for the purpose of sentiment analysis and could not be used for the task at hand. The proposed new model, called the Hybrid Model, uses the benefits provided by the DeClarE model towards fake news detection and overcomes its weaknesses by using Self-Attentive Sentence Embedding as its core. It overcomes the problem of input processing by combining the bidirectional LSTM and self-attention into a single flow of the network which can be seen in Figure 2.2. As given in the paper by Lin et al. (2017), this combination will focus the attention on those aspects of the article which are related to the claim while not losing any data due to averaging.

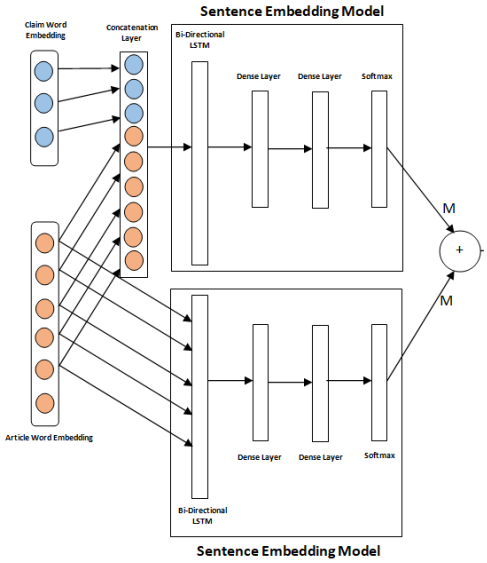

Figure 2.2: Partial structure of the Hybrid model. Here the '+' is used to show a element-wise multiplication operation

With the use of the penalization term utilizing the Frobenius norm, any redundancy caused due to using the article twice is handled. Also, as the model has a single core component, represented as the Sentence Embedding Model in Figure 2.2, it can be reused which further reduces the code complexity. This was not the case for the DeClarE model which had two separate streams of network

movement involving different components. It must be noted that the upper and lower portion of the model is combined using element-wise multiplication (as seen in Figure 2.2) rather than with inner product as done in DeClarE model. Multiplying the weights returned by the two networks elevates emphasis on the terms focused on by the Sentence Embedding section rather reducing them down through a dot product (which gives a scalar result).

To allow the model to be robust and generalizable, the data processing proposed for the DeClarE model is not attempted [See Appendix A]. The articles having very little similarity with the claim term may still share the topic information on the news and should, therefore, be classifiable. The DeClarE model also combined those claims that did not have many articles associated with them into a dummy claim. They did not specify how the dummy claim text is worded or what the credibility value was set for this claim, hence this was also not included in the data processing part.

Due to the advantages that the model has over DeClarE, it is hypothesized that the Hybrid model will perform better than the DeClarE model on the datasets provided by the authors of the DeClarE model. One of the main objectives of the research is to create the Hybrid model and prove that the hypothesis is not false.

2.5 Research Objectives

The objectives of this research project and its scope can be summarized as below:

- Objective 1: DeClarE Model Reproducibility
    - Recreation of the DeClarE model
    - Handling Dataset Imbalances
    - Evaluation of the model against results stated in the paper
    - Study the behaviour of the hyper-parameters
- Objective 2: Creation of the Hybrid model
    - Generate a new model which will have better performance than the existing state of the art
    - Evaluate the new model performance metrics against the DeClarE Model
- Objective 3: Manage experimental bias introduced into the evaluation due to the repetitive structure of the datasets and report the results. This objective is crucial as the DeClarE paper does not report their evaluation results on unique claims and is a fairer estimate of a models performance.

2.6 Summary

In this chapter, the prior research done on the topic of fake news detection is discussed. The state of the art model in this field, i.e., the DeClarE model is analyzed along with its drawbacks. A solution to the drawbacks is proposed in the form of a new model named the Hybrid model that utilizes the Self-Attentive sentence embedding in combination with the outline of the DeClarE model. Based on this analysis, the objectives of the project are laid out. In the next chapter, the model setup methodology is discussed which involves the design and implementation of the involved models.

Chapter 3 Methodologies

This chapter details the model designs and their implementation. The chapter also further illustrates the challenges faced during the process of implementation of the models.

3.1 DeClarE Model

3.1.1 DeClarE Model Design

The DeClarE model is an evidence aware deep learning model that utilizes an attention mechanism to allow the model to focus on certain parts of the articles. The design of the model is unique in the aspect that it considers external sources of the claim and articles which provides evidence for the credibility assessment of a claim. In that regard, this is a multi-input model.

Model Structure

The model consists of two major sections. The first, which is the upper part of Figure 3.1, is the attention mechanism which is used to represent the interconnections between the article and claim. The attention mechanism allows the model to focus on the salient features of the article that support the claim and hence this section requires both article and claim embedding in order to work. As can be seen from the figure, the attention mechanism comprises of the concatenated article and claim embedding which is passed through a dense layer having an activation function of *tanh* or *ReLU*. This is then passed through a *softmax* layer to get the final attention weights as given in (1). Here a_k represents the attention score for each article word and its relevance to the claim.

$$a_k = \frac{\exp(a'_k)}{\sum_k \exp(a'_k)} \qquad (1)$$

The second section, the lower part of Figure 3.1, is a Bidirectional LSTM which takes just the article embedding as the input. As per the authors, Bidirectional LSTM is used so that the model can capture the article related salient features. As article stories tend to large, LSTM can handle the long dependency and memory problems that would otherwise hamper learning. Bidirectional LSTM is used instead of normal LSTM to capture both future and past features which are concatenated to give the final output. This output is combined with the attention weights through an average of the dot product between them to give a weighted average of the hidden state for the articles as in (2) where h_k represents the article term embedding for the article input matrix.

$$g = \frac{1}{k}\sum_k a_k h_k \qquad (2)$$

This is then passed through several dense layers after combining with the article and claim source embedding. The final layer of the model can vary depending on how the model is going to be used. If it is going to be used for a credibility score calculation through a regression task then a linear activation function is used. For a classification task, a sigmoid activation function is used to give a binary

output about the credibility of the claim. As the aim for of the research is to classify a claim as being true or false therefore for this paper the second method is used.

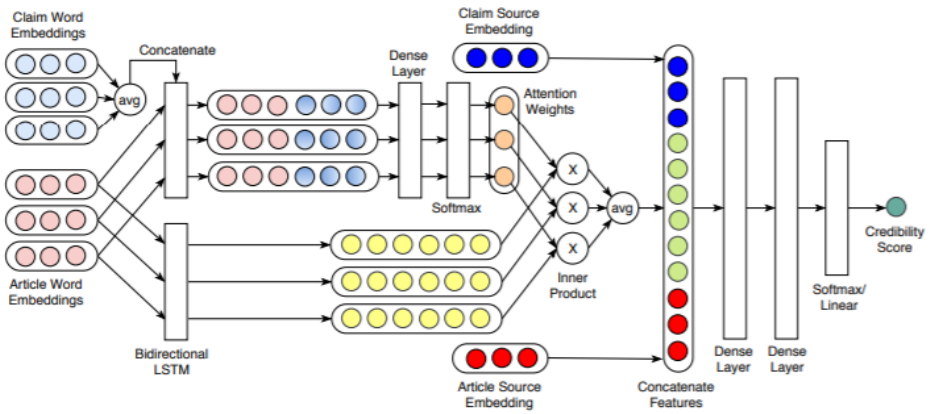

Figure 3.1: The figure details the structure of the DeClarE model. [Popat et al., 2018]

Inputs

For the model, an input tuple is created that includes the claim C_n, the corresponding article $A_{m,n}$, claim source CS_n and article source $AS_{m,n}$. C_n is of length *l* and each C_i is a d-dimensional word embedding of the l^{th} claim word in C_n. CS_n is the word embedded representation of the claim source. Each claim can be present in multiple articles, hence $A_{m,n}$ is a *m*n* matrix of articles associated with the claim. As in the case of claim, each article word, $A_{m,n,k}$ is represented by a word embedding. The article source is also linked to the article hence is also having dimensions *m*n*. Even though multiple articles and their sources are associated with a single claim, they are not combined to form a single input. Each article and its source are taken as a separate input with the claim and claim source. Therefore the input is <C,A,CS,AS> which are shown in Figure 3.1 input as the claim word embedding, article word embedding, claim source embedding and article source embedding.

3.1.2 DeClarE Model Implementation

The DeClarE model is implemented in Python using Keras with Tensorflow as the backend. Keras library is selected because it would have smaller learning curve as compared to other deep learning libraries and it is taught as a part of the Deep Learning course. Google Colab is used as the development environment due to the availability of a performance GPU. The structure of the implementation is given in Figure 3.2.

The inputs to the model are the vector representation of the input data. The claim and article are padded to the same length. The article and claim text are of different length and in order to create an embedding matrix for them, Keras requires the input to be of the same length; hence padding is added to the input. The inputs are passed through the embedding layer to create the required word embedding. As per the DeClarE design, an average of the input claims is taken per claim word embedding which is then concatenated with the article word embedding after passing both of them through the Flatten layer. The Flatten

layer converts the input into a one-dimensional vector. This averaging is done with the help of a Lambda layer. As Keras does not have an inbuilt averaging function to get the mean along one dimension, Lambdas are used. This allowed user-defined functions to be wrapped into a Keras layer to ensure the flow in the network is not broken. The concatenated output passes through a dense layer with the activation of *tanh* and an Activation layer with *softmax* function. This forms the DeClarE Attention model.

Article embedding by itself also passes through a Bidirectional LSTM. The output of the bidirectional LSTM model and the Attention model are then combined through a dot product. To accomplish this, the dot Functional API of Keras is used. The average of this output is concatenated with the flattened embeddings of article source and claim source. This is then passed through two dense layers of with activation function of *ReLU*. Finally, the model is passed through a sigmoid activated dense layer to get the output between 0 and 1. For model compilation, binary cross-entropy is used rather categorical cross-entropy which is what the authors used. The reason for the change is that the model is being used for binary classification and using categorical cross-entropy threw compilation errors. Another change in the model is that the LSTM size for each pass is 50 rather than 64 as it is not compatible with a padding of 100 given to the input in terms of visualization of weights. The LSTM size of 50 is needed in order to seamlessly combine through dot product (as dot_13 in Figure 3.2) with attention weights returned by the attention mechanism.

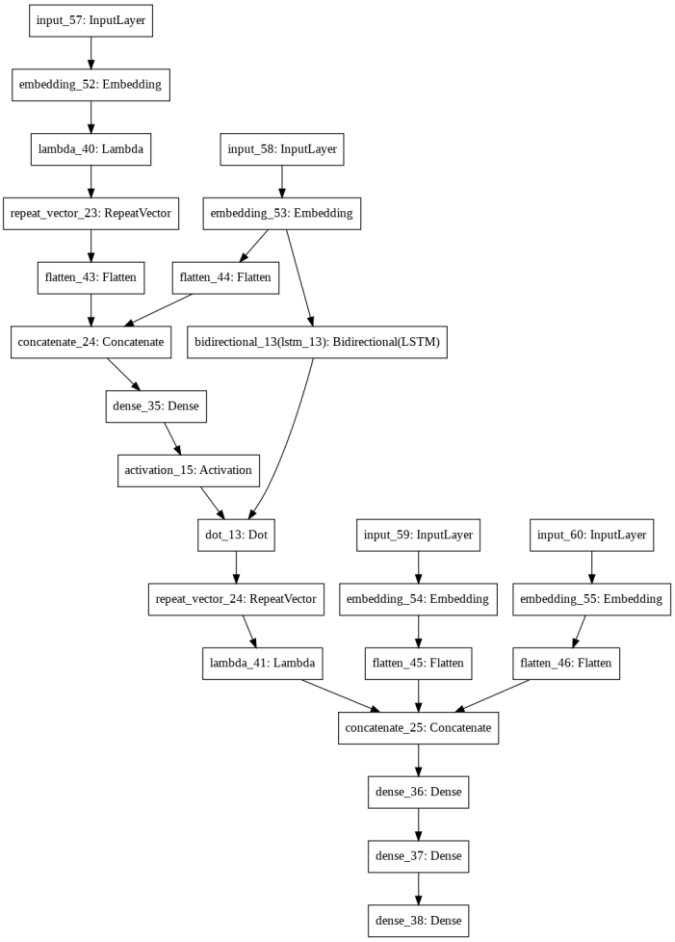

Figure 3.2: DeClarE model layer structure in Keras.

Challenges

Many challenges were faced during the recreation of this model and it took up the bulk of the project time to make it as precise as possible. One of the major challenges faced was that the authors had not provided the source code for their paper leading to anyone trying to verify their results having to generate the model from scratch. Initial search for the implementation led to various partially created models. One of these, which is used as the base for the implementation, was found to have many compilation errors. Trying to fix these errors wasted a lot of time and eventually, it was decided to create the model again by using sections of this one as the foundation. A major problem was faced trying to average the claim embeddings. The averaged claim embedding and the article source it was to be combined with were initially throwing errors due to incompatibility of the dimensions. It was then thought to use the Flatten layer to overcome this problem but the Flatten layer also expected the input to be of a certain dimension. After various trials and with the use of a Repeater layer, this problem was rectified. The Repeat layer is used to repeat the input matrix a number of times along an axis. The learning rate was also changed from 0.002 to 0.0001. This was done because the model reached the highest possible accuracy of ~55% and loss of ~0.65 in a single epoch and did not change in the subsequent epochs with 0.002 learning rate. This points to the fact that the model did not learn anything.

3.2 Hybrid Model

3.2.1 Hybrid Model Self-Attention Design

The proposed Hybrid model exploits the DeClarE model structure. It consists of four inputs and the central part is the Self-Attentive Sentence Embedding. The DeClarE model also uses an attention mechanism but as we discussed in Section 2.1, a major problem with that is that it is only applied to one half of the model. In order to focus more on the salient features, the output of the Bi-LSTM is also passed through a self-attention mechanism as in Lin et al. (2017). This generates the sentence embedding as shown in Figure 3.3.

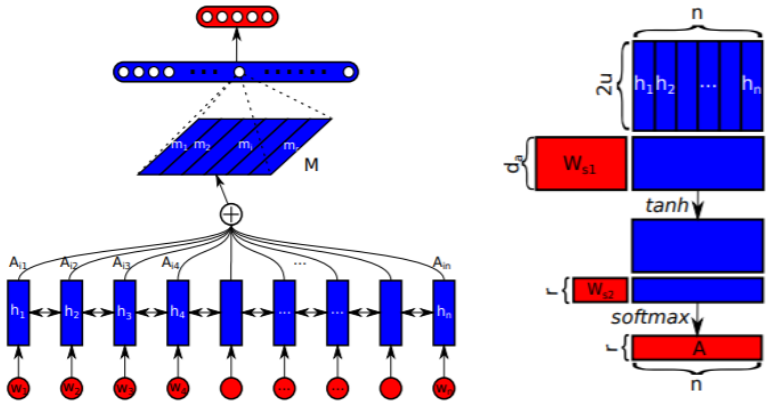

Figure 3.3: The left-hand figure shows the Bidirectional LSTM and its output which combines to form the sentence embeddings. The right-hand figure shows the self-attention mechanism applied on the hidden states. These are combined to create the matrix M. [Lin et al., 2017]

In the Hybrid model, the idea is to find common words in the combined input (Article+Claim) with the aim of emphasizing the article words present in the claim with the help of attention weights. To achieve this, the concatenated article and claim tensors are passed through the bidirectional LSTM, the output of which is passed to the self-attentive mechanism. The combination of passing through LSTM and attention mechanism creates an embedding matrix M_1 as in (3) [Lin et al., 2017]. The concatenation ensures that specific common features between the two are focused on in the article.

$$M_1 = AH \qquad (3)$$

Here, A represents the outcome of the self-attention through a *softmax* activation layer over the second dimension. H represents the hidden states of the LSTM. When the combination of article and claim is passed through this combination of Bidirectional LSTM and self-attention, the resultant matrix M_1 represents the weight matrix where higher weights are given to those terms that are common to both claim and article.

Our objective is to highlight the article words that provide information about the sentiment of the article. Due to this reason, the article embedding is separately passed through the self-attention mechanism again. When used separately on just the article embedding input, it works like a sentiment analysis system that is useful for picking out positive and negative words in the article and resulting in embedding matrix M_2. The two sections are combined and averaged. They are combined using element-wise multiplication, see equation (4), which is in contrast to the dot product used in the DeClarE model. The reason multiplication is selected instead of a dot product was to aggregate the results of the weights from the two matrices, M_1 and M_2. The areas with higher weight will have more focus through multiplication and hence overlapping elements will be highlighted. This way the terms common to article and claim are combined with words related to sentimentality to provide credibility to the claim.

$$M = \sum_i m_{1i} m_{2i} \qquad (4)$$

This result (M) is concatenated with the averaged out article source and claim source embedding. A visualization of how these calculations affect the attention weight of the article embedding is given in Figure 3.4. Average of the outcomes is taken which presents an overall representation of the embeddings and enhances the attention focused features of each individual section. These results are then passed through a dense layer and a sigmoid activated fully connected layer to get the result in the range of 0 to 1.

As the article is used twice in this model, once in combination with the claim and then separately, there may be some redundancies. In order to counter them, a penalization term is added during training. The penalization term allows the outcome to be more focused. The penalty term added is in (5).

$$P = \|AA^T - I\|_F^2 \qquad (5)$$

Here, the F stands for the Frobenius norm. This penalization term is an integral part of the self-attentive research paper and the results of its impact on attention weight matrix prevents high weight values to be spread everywhere in the matrix and hence not being able to show with terms are actually of importance. This is the reason this term was also used in the Hybrid model. This penalization

forces the model to focus on only a single aspect of the input and prevents the heavy accumulation of weights on identical inputs. This is achieved by the AA^T-I term which forces the diagonal elements of AA^T to approach 1 whereas the rest are zero.

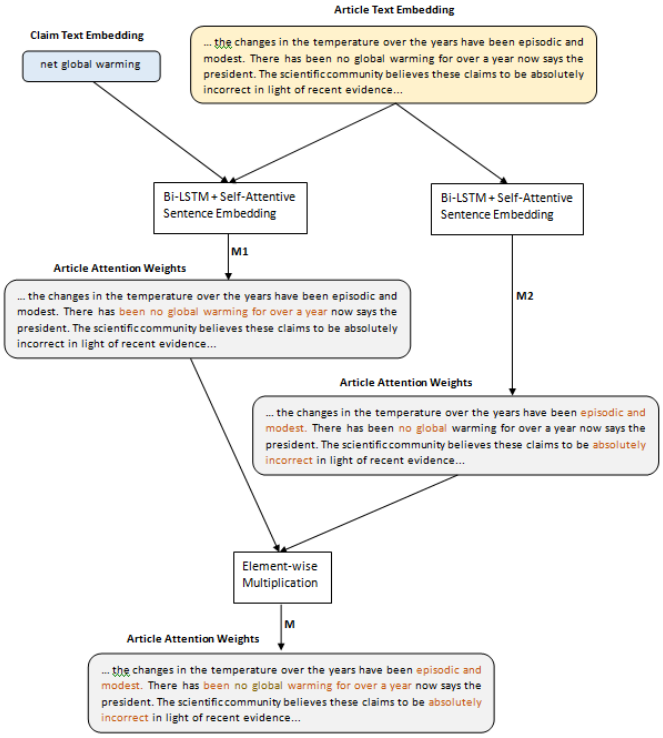

Figure 3.4: Visualization of the attention weights when passed through the Self-Attention mechanism. Here, the red coloured text shows the weight emphasis. In the output Attention weight, the combination of the two weights can be seen with 'global warming' in a darker shade as it is highlighted in both M1 and M2.

The inputs to this model are similar to the DeClarE model and share the same dimensions.

3.2.2 Hybrid Model Implementation Approach

The Hybrid model is implemented in Python using the PyTorch library. The PyTorch library is an easy to use library that has a lot of online support available. It allows us to easily create complex equations inside the network layer which is not possible with Keras. To do a simple average operation in Keras, a functional API is to be used or a lambda function has to be created which is difficult to do if one does not have experience in creating one. This is much simpler in PyTorch as mathematical operations can be easily included in the model class without disrupting the flow of the network. The model generated in the Self-Attentive Sentence Embedding paper is very good for getting semantically relevant information from the input. This is because the sentence is represented as a matrix rather than a vector. Various versions of its code are available online, one of which was used to form the basis for our Hybrid model.

In the model, the input article and claim are converted into a tensor of vector sequences and are padded to ensure consistent length for all rows. One thing to be noted as a difference between the input to DeClarE and Hybrid model is the length of the claim embedding. In the DeClarE model, both the article and the claim were padded to the size of 100 making the total input size to be 200. In the

Hybrid model, the claim vector padding is kept as 50 while the article padding is still 100 making the shape along one dimension of the tensor to be 150. The size 100 is kept as taking into account that the maximum size of the article is around 100. The claim size is kept a 50 for the same reason. Also, rather than 200, 150 reduces a lot of the zero-padded input to the model. This zero-padding adds no insight into the article features and hence is better to keep its size to a minimum. A class was created for the model which contained a *forward()* method which defines the flow of the network and can be seen as the representation of the model. The *forward()* method is a central method for a neural network created in PyTorch as it specifies how the data will flow in the model.

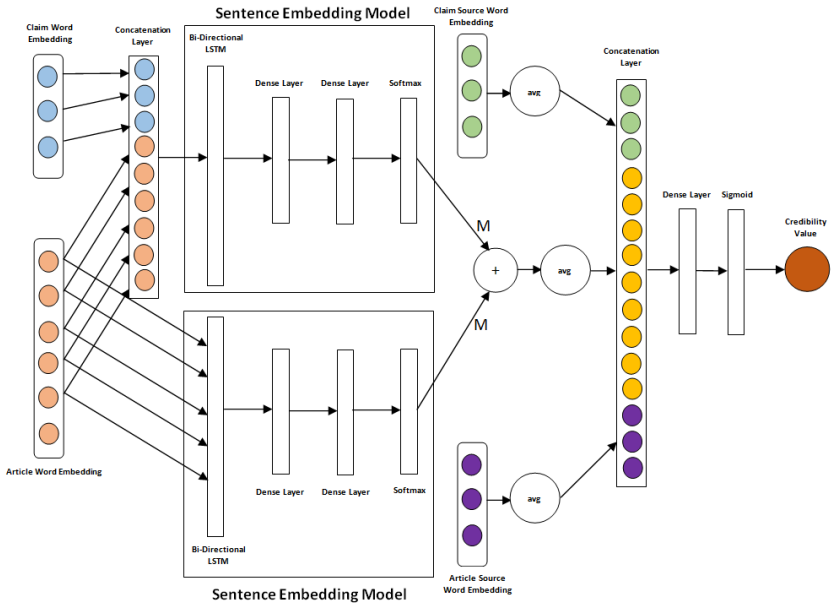

Figure 3.5: Detailed structure of the Hybrid model. In the model the '+' symbol indicates an operation on the input which in this case is matrix multiplication. The Sentence Embedding Model structure which is central to the Hybrid model is the structure presented in Lin et al., 2017.

As can be seen in Figure 3.5, the article and claim input embedding is concatenated together to form a single input. Like in Keras, PyTorch also offers an Embedding layer which was used to create the embedded input for the model. For the embedding layer, a pre-trained embedding is used. The concatenated output is then passed to the Sentence Embedding Model (SEM). Through this combination, at each attention hop, the model will find common words in the combined input allowing those articles which have more in common with the claim to have higher attention weights. The concatenated input is passed through the SEM which consists of a bidirectional LSTM and the self-attention mechanism. This produces the sentence embedding matrix M. The internal flow of the Sentence Embedding model is the same as in Figure 3.3. The article embedding is also separately passed through the SEM. This works as a sentiment analysis for the article wherein the attention hops focus on the negative and positive aspects of the article. A negative tone may have words related to its falsehood and opposite for the truth. Both these outcomes of the SEM are combined through element-wise multiplication to enhance common weights and averaged out to get a proper state representation for each article. The multiplication and the averaging are done as normal mathematical operations on PyTorch variables and no special layer or Lambdas are required.

The output is concatenated with the averaged article source and claim source embedding which is passed through a dense layer and sigmoid function to get the final credibility classification. A dense layer is an object of class Linear built into PyTorch and for sigmoid; the inbuilt sigmoid function is used. For the training of the model, the regularization term discussed in Section 3.2.1 is used along with Binary Cross-Entropy Loss.

Challenges

One of the major challenges we faced for this model was the learning curve for understanding PyTorch. Even though it is an easy library to use for beginners, none of its tutorials offers information on handling multi-input structures. The creation of the multi-input structure was done through trial and error which consumed more time than the creation of the rest of the model. Also, the baseline code had bugs concerning the calculation of the accuracy and the visualization of the weights which took some time to fix. Also as the performance of this model was too high, many different variations of the model were created to find if this issue is occurring due to a bug [See Appendix B]. The changes in the accuracy values were checked at the addition of different input combinations. This change gradually increased at the addition of each new input which showed that the model was positively responding to the inputs and learning from them.

3.3 Summary

In this chapter, we studied the DeClarE model in details and listed the process flowed to implement the model in Keras. We further elaborated the challenges faced during its construction. Next, the chapter focuses on the methodology of the new model, the Hybrid Model, specifying the design decisions made in its creation process. The section also includes details about the construction of this model in PyTorch and the challenges faced during the process. As a novel approach, the Hybrid model combines the bidirectional LSTM with the self-attention mechanism, to take advantage of the benefits provided by both and rather than use them separately as done in DeClarE.

Chapter 4 Evaluation

In this chapter, the performance of the recreated DeClarE model and the Hybrid model is evaluated against various metrics. The Snopes and PolitiFact datasets are used for this purpose.

4.1 Datasets

For the purpose of evaluation of the two models, the Snopes and the PolitiFact datasets are used. The reason for using these datasets was to ensure that the research done on these models would match the one done in the Fake New Detection paper [Popat et. al, 2018]. In the paper, the author assessed the model against the Snopes, PolitiFact, SemEval-2017 Task 8 and the NewsTrust datasets. The links to three of these datasets (Snopes, PolitiFact and NewsTrust) were provided by the author. An attempt was made to do the same but both the SemEval and the NewsTrust datasets were found to be incomplete in some way and hence were not used.

It must also be noted that the structure of the datasets is such that a claim can have many articles related to it. Hence, a claim may appear multiple times as an input, each time with a different article. This may cause the claim to be repeated in both the training and test data.

4.1.1 PolitiFact Dataset

The PolitiFact dataset consists of news evaluations from the PolitiFact website, a fact-checking website. The dataset contains the credibility label, claim ID, claim text, claim source, article and article source. Here the credibility label had five possible values based on the ratings given to news articles by PolitiFact which are- 'Pants on Fire!', 'False', 'Mostly False', 'Half-True', 'Mostly True', 'True'. To use this dataset for binary classification, 'Pants on Fire!', 'False', 'Mostly False', were changed to 0 (false) and the rest to 1 (true), as done for the DeClarE model. This dataset had a near equal division of true and false claims. The statistics for this can be seen in Table 5.1.

Datasets	PolitiFact	Snopes Before Down-sampling	Snopes After Down-sampling
Total Claims	29556	29242	15507
True Claims	15019	7507	7507
False Claims	14537	21735	8000

Table 5.1: Details about the PolitiFact and Snopes dataset.

4.1.2 Snopes Dataset

As with the PolitiFact dataset, the Snopes dataset also contained news evaluations from the Snopes website. The dataset contained the credibility label, claim ID, claim text, article and article source but no claim source. The values for the credibility label here were - 'false', 'mostly false', 'mostly true', 'true', which were changed to 0 for 'false', 'mostly false' and 1 for the rest, as is done in DeClarE. After this change, the class disparity was seen between false and true claims. There were three times as many false claims (21735) than true claims

(7507). To rectify this class imbalance problem, in the data preprocessing part, down-sampling of the false claims was done to ensure that the model was not unduly biased towards a single class. After pre-processing, the details about the dataset are given in Table 5.1.

Both the models are experimented on using the same dataset settings given in Table 5.1 in order to study the differences.

4.2 DeClarE Model Training Analysis

The DeClarE model was compiled using the Adam optimizer with a learning rate of 0.0001. The model was run with a batch size of 100 in 30 epochs. Because the number of epochs was set to 30, the learning rate was kept low to allow the model to learn features at each epoch. If the learning rate was kept high, the model reached its lowest loss and accuracy in a single epoch without actually learning anything. An 80-20 split is done for the training and test data. The accuracy and loss for the training and validation of the model are given in Figure 5.1 (a) and (b) for the PolitiFact dataset and Figure 5.1 (c) and (d) for the Snopes dataset.

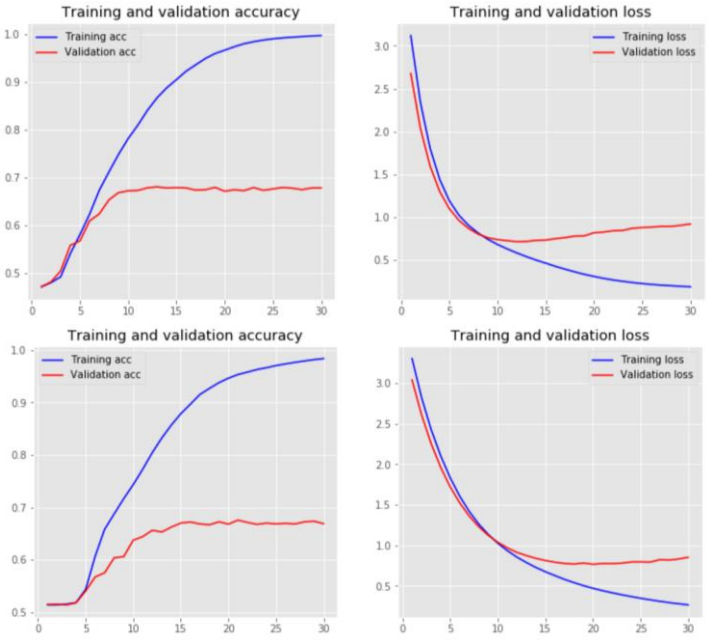

Figure 5.1 (a) and (b) (Top two): For the PolitiFact dataset. **(c) and (d) (Lower two):** For the Snopes dataset. Figures on the left (a) and (c), show the training and validation accuracy. Figures on the right (b) and (d), show the loss for training and validation.

As can be seen from Figure 5.1(a) and (c), the training accuracy curve for both PolitiFact and Snopes are similar, showing that the model learns at a similar rate irrespective of the dataset. In both cases, the training accuracy starts to level of around the 20th epoch. This shows that it was not necessary to train the model for 30 epochs, and the model would have reached its highest accuracy even if we stopped the training at 20. This is also evidenced by the training loss of both the datasets as in Figure 5.1 (b) and (d). There is an exponential decrease in the loss up to 25th epoch after which it slowly reduces to a value of ~0.45 for both datasets. But as the validation loss is higher than the training loss, the model is

not over-fitting. When the validation accuracies are compared, it shows a sharp increase up to the 15th epoch for Snopes and 10th epoch for PolitiFact. A similar trend is noticed in the loss curves. So this shows that if we had trained the model for 15 epochs rather than 30 we would have gotten a similar result. This would have saved on the training time on the model, reducing its learning time by half while keeping the same level of effectiveness. A mistake was made during the training process wherein emphasis should have been given to the validation data metrics rather than training data. This was not rectified as it did not influence the results of the model but did affect the duration of the training time.

4.2.1 DeClarE Model Evaluation Metrics

As done in the paper, the improvement of the model was checked for different configurations and its performance noted. The details for each of the configurations are given below:

- DeClarE (Plain) - This model only consists of the bi-LSTM i.e. no attention or source embedding. This configuration takes the article embedding as an input.
- DeClarE (Plain+Attn) - This configuration consists of the bi-LSTM and the Attention but no source embedding. This means that the model takes the article and claim embedding input.
- DeClarE (Plain+SrEmb) - The configuration consists of the bi-LSTM and the source embedding. This implies that the input to the model is the article embedding the claim and the article source embedding.
- DeClarE (Full) – This is the complete model with bi-LSTM, attention and the source embedding.

To gauge the performance of the models, Accuracy, F1 measure (macro) and Area under the Curve is used. The performance of each the configurations is documented for both the Snopes and the PolitiFact datasets. A simple Bi-LSTM model is used as the baseline in order to evaluate the reproducibility of the model at predicting the credibility of the news. As both the DeClarE and Hybrid model use Bi-LSTM in their structure, having a simple model using just Bi-LSTM seemed fitting as a baseline. The details of the metrics are given in Table 5.2. In the table, the 'Reported' values highlighted as grey rows are the performance metrics reported by the authors in the DeClarE paper.

Main Results

As can be seen from the table, we can say that our implementation of the model has near accurately reproduced the reported results for the PolitiFact dataset. For each of the configurations, PolitiFact shows performance similar to the reported data with a variation of ~2-4% for accuracy. However, our results on the Snopes dataset diverge from the reported results. For the authors, the Snopes dataset had the highest performance and was second only to a Distant Supervision model that they had compared with. When compared to the results received in our implementation, Snopes dataset, though fares better than PolitiFact in F1 measure and AUC for DeClarE (Plain) and DeClarE (Plain+SrEmb) but not for the Full configuration. The true claims accuracy in DeClarE (Full) is deficient by ~20% as compared to the reported, which is a large margin to fail by. A peculiar trend noticed is that the performance is only lacking in the (Plain+Attn) and (Full) configurations. When we analyze the structure of

these configurations, it is seen that they are the only configurations that include claim as an input. This lack of performance provides evidence towards the quality of the claim text as an input in the Snopes dataset.

Result Discussion

The DeClarE model recreated for this project matches the performance reported by the authors only for the PolitiFact dataset. While Snopes dataset does diverge from the results, it is still not predicting the results randomly. Even with the changes made in the hyper-parameters, like learning rate, the metrics follow a similar trend. In Table 5.2, The Full configurations have an F1 score of 0.65 and 0.67 for Snopes and PolitiFact respectively. These are slightly above average score for a binary classification model and show that while the predictions are not random they are still not as good as one would expect from a state of the art model. Of these results, the F1 score for Snopes is a cause for concern as the expected value from the model for it was 0.79, a very good performance for a model. For the AUC measure, both the datasets are deficient by a margin of 0.2. This implies that the model created by the authors show a greater distinction between True Positives and True Negatives when compared to the model recreated here. Even with changes to the hyper-parameters, the AUC values could not be recreated for our model.

Dataset	**Snopes**					**PolitiFact**			
Configuration	Baseline model	DeClarE (Plain)	DeClarE (Plain + Attn)	DeClarE (Plain + SrEmb)	**DeClarE (Full)**	DeClarE (Plain)	DeClarE (Plain + Attn)	DeClarE (Plain + SrEmb)	**DeClarE (Full)**
True Claims Accuracy	39.4 %	71.50 %	56.5%	74.80 %	**57.7%**	67.70 %	68.4%	67.7%	**67.0%**
Reported True Claims Accuracy	-	74.37 %	78.34 %	77.43 %	**78.96 %**	62.67 %	65.53 %	66.71 %	**67.32%**
False Claims Accuracy	70.1 %	71.80 %	76.20 %	67.70 %	**73.8%**	68.90 %	64.6%	71.6%	**68.4%**
Reported False Claim Accuracy	-	78.57 %	78.91 %	79.80 %	**78.32 %**	69.05 %	68.49 %	69.28 %	**69.62%**
Macro F1 Score	0.54	0.716	0.6616	0.711	**0.656**	0.683	0.664	0.696	**0.676**
Reported Macro F1 Score	-	0.78	0.79	0.79	**0.79**	0.66	0.66	0.67	**0.68**
AUC	0.547	0.716	0.663	0.712	**0.657**	0.683	0.664	0.696	**0.676**
Reported AUC	-	0.83	0.85	0.85	**0.86**	0.70	0.72	0.74	**0.80**

Table 5.2: Comparison of the various configurations of the DeClarE model.

The DeClarE model recreated here is compared against a baseline model which is a simple bi-LSTM with a dense layer. While it is important to compare the results of the recreated model with the reported results, it is also important to compare the results of the model on its own. For this reason, the baseline model is used. The baseline model has a low accuracy score for true claims (~40%) and high for false claims (70%). This is an expected behaviour for any dummy model when the dataset has more false claims than true claims. The F1 measure is also 0.54 showing that this model predicts near-random credibility values for the input. The baseline model results are compared to the Snopes dataset and not PolitiFact because the performance metrics on Snopes are lagging behind when compared to the reported values. The comparison was done to show that while the results of Snopes are less than the expected, individually, it performs better than the baseline model by a margin of ~20%.

The reason for the major deviation in performance in Snopes when compared to PolitiFact can be attributed to the data processing done before training the model. The authors had done pre-processing of the data to include only those <claim, article> sets that had a maximum relevance score for both datasets. This reduced the size of the datasets to 1/6th its original size. The same was attempted for this project, but the data size reduced by 1/8th of the original and the results were also close to 50% [See Appendix A] so this approach we balanced the dataset by down-sampling instead. Another major difference between the two is the Learning Rate. The rate specified in the paper is 0.02 whereas the one used for this project is 0.0001. As stated before this was done to give the model more time to learn considering the number of epochs it was run for. One other cause could be that the PolitiFact dataset contains more training data and is more balanced than Snopes. A third cause for the difference in performance is related to the quality of the datasets. Given in Table 5.3 is a snippet of data from the PolitiFact and Snopes dataset.

Columns	PolitiFact Data	Snopes Data
Claim	federal tax code loopholes giving incentives companies shipping jobs overseas	best buy chain eschewing use word christmas 2006 holiday print advertising
Claim Source	barack obama	-
Article	for firms moving overseas in order to create a disincentive to offshore what they say though makes it sound like the tax code is currently luring companies out of the us but i also want to close those loopholes that are giving incentives for companies that are shipping jobs overseas i want to provide tax breaks for companies that are investing here in the united states obama said wednesday he went on to say right now you can actually take a deduction for moving a plant overseas i think most americans would say that doesnt make sense and all that	there are several holidays throughout that time period and we certainly need to be respectful of all of them pic of best buy ad wishing muslims a happy eid link to quoting 2 best buy christmas ban is the best buy chain eschewing use of the word christmas in its 2006 holiday advertising home politics christmas best buy best buy claim the best buy chain is eschewing use of the word christmas in its 2006 holiday print best buy has announced they will be using happy holidays this coming christmas shopping season and they will not be using merry Christmas

Table 5.3: Row snippet of the PolitiFact and Snopes datasets

In the Snopes dataset, the Article text which is a crucial input to the model has a lot of repetitions and grammatical errors as highlighted in the table whereas the

PolitiFact dataset has a proper structure for both the Article and the Claim. Another difference is that Snopes dataset has no claim source input and is used as such by the authors. This makes the dataset incomplete. These reasons may cause the model to perform better with PolitiFact than with Snopes.

Through this evaluation, we can say that the DeClarE model was implemented somewhat successfully with some changes in the hyper-parameters and input data. The dataset imbalance was also handled by the use of down-sampling. An area for concern that still remains is experimental bias introduced due to the appearance of the same claim in both the training and test data. Because of this, the results may not reflect the true performance of the model.

4.3 Hybrid Model

As with the DeClarE model, evaluation of the hybrid model was done using the same metrics and datasets. The optimizer used for this model is RMSProp as it yielded better results when compared to other optimizers [See Appendix B]. While running, it was seen that the model reached the highest accuracy in just 3 epochs and so the same for maintained for each of its configurations defined below. The datasets are divided in an 80-20 split of training and test set. To analyze the performance of the model, its evaluation was documented for the different configurations as was in the case of DeClarE. These results are listed in Table 5.4.

The various configurations and their details are given below:

- Hybrid (Article) - This configuration takes only a single input and works similar to the original base model for the Self-Attentive Sentence Embedding. This configuration uses only a single self-attentive layer for the Article embedding.
- Hybrid (Article+Claim) - This takes two inputs, the article and the source embedding. It uses two self-attentive structures, one for the article and the other for the article and claim.
- Hybrid (Article+SrcEmb) - The model takes the article embedding which is inputted into a self-attentive structure which is then combined with the article and the source embedding. This configuration takes three inputs- article, article source and claim source.
- Hybrid (Full) - This represents the full model with four inputs of the article, claim, claim source and article source.

4.3.1 Model Evaluation Metrics

As the hybrid model is a vastly different variation to the original model, it cannot be compared to the results produced by the authors of the paper [Lin et al., 2017]. From Table 5.4, it can be seen that the model gives very high performance on both the datasets.

A score of such a high value does raise questions about model over-fitting on the data but as the test data is supposed to be unseen, such an argument cannot be definitively made. Also, the progressive increase in the performance with each successive configuration does point towards the impact the inputs are making and adding to the performance. Furthermore, other optimizers like Adam were

used to check for variation in performance, while there was a drop of about 5-10% in precision, there was also a drop in performance for unseen data [See Appendix B]. This shows that the current configuration is the best one for the model. The Hybrid (Article+Claim) and Hybrid (Full) has a better performance (> 95%) than Hybrid (Article) and Hybrid (Article+SrEmb). When (Full) is individually compared with (Article+Claim), (Article+Claim) shows better results, with an increase of ~1% in F1 measure and AUC. This shows that the Hybrid model can provide correct classification even when two of its inputs are not present making it ideal for an incomplete dataset like Snopes.

Dataset	Configuration	True Claims Accuracy (%)	False Claims Accuracy (%)	Macro F1 Score	AUC
Snopes	Hybrid (Article)	94.9	79.4	0.867	0.871
	Hybrid (Article + Claim)	**97.2**	**97.5**	**0.974**	**0.973**
	Hybrid (Article + SrEmb)	90.5	82.6	0.863	0.865
	Hybrid (Full)	**96.8**	**95.5**	**0.961**	**0.961**
PolitiFact	Hybrid (Article)	88.1	79.2	0.837	0.836
	Hybrid (Article + Claim)	**94.2**	**97.5**	**0.958**	**0.958**
	Hybrid (Article + SrEmb)	82.2	90.0	0.861	0.86
	Hybrid (Full)	**96.5**	**97.3**	**0.969**	**0.968**

Table 5.4: Comparison of the various configurations of the Hybrid model.

Result Discussion

From Table 5.4, we can see that the model has a very good performance for in all of its metrics for both the datasets. The fact that it performs well even for an incomplete dataset like Snopes shows how flexible the model is. It can be said that for the Hybrid model, in order to make accurate predictions, four inputs are not required. While the fact is not obvious for the Snopes dataset, it is clearly visible in the PolitiFact dataset. For the Snopes dataset, the difference is not large, there is only 0.03 difference in its F1 measure and 0.06 in the AUC value between its (Article+Claim) model and the Full model. In PolitiFact, there is ~2-4% difference in the performance.

Through this evaluation, we have achieved one of our objectives of creating a new model which performs better than the state of the art. But a major issue that still exists, related to the authenticity of these results, is experimental bias. As with DeClarE, during testing, the model may have already seen some of the claims in the training data, therefore, these results could be constituted as being unfair. A solution to this problem is discussed in the next section.

4.4 Critical Analysis between Models

4.4.1 Model Comparison

A comparison of the results of the models is given in Table 5.5 using the results of the DeClarE and the Hybrid models from Tables 5.2 and 5.4. For the purpose of this evaluation two versions of the Hybrid model are used: the (Article+claim)

and (Full) configurations. This is done because of two reasons. One, both of these configurations yielded very high performance metrics and, two, we also want to take into consideration the hypothesis that article and claim embedding are not required to get the best results. As can be seen from the table, in both aspects of F1 measure and AUC, the Hybrid model for both of its configurations has performed better than the DeClarE model. There is a difference of nearly 30% in their results. This points towards the superiority of the Hybrid model over the DeClarE model. With the use of self-attention and Bi-directional LSTM for the critical inputs of article and claim, allows the model to perform well even for an incomplete dataset like Snopes. This also holds correct for the true and false claim precision values. For DeClarE the precision values remain in the domain of 55-75% whereas it is >95% for the Hybrid model for both configurations. Also, as the AUC values approach 1, the Hybrid model is able to correctly predict both True Negatives which in this scenario are false claims and True Positives (true claims) to a high level of accuracy. That is not the case for DeClarE which has an AUC score of 0.65-0.67. When the Hybrid model is compared internally, the (Article+Claim) configuration performs better only for the Snopes dataset and worse for PolitiFact. This may be due to the dataset differences highlighted in Section 4.2.1. With only a variation of 0.1, it cannot be conclusively said whether (Article+Claim) is better than (Full).

Statistics	DeClarE(Full)		Hybrid (Full)		Hybrid (A+C)	
	Snopes	PolitiFact	Snopes	PolitiFact	Snopes	PolitiFact
True Claims Accuracy	57.7%	67%	96.8%	96.5%	97.2%	94.2%
False Claims Accuracy	73.8%	68.4%	95.5%	97.3%	97.5%	97.5%
F1 Macro Measure	0.656	0.676	0.961	0.969	0.974	0.958
AUC	0.657	0.676	0.961	0.968	0.973	0.958

Table 5.5: Comparison table between the DeClarE and the Hybrid model. Here, A+C means Article+Claim.

In term of efficiency, it can be said that the DeClarE model is slower as it was only able to reach this level of accuracy after 15 epochs where Hybrid was able to achieve a much better score in 3. This comparison shows that the Hybrid model is better at predicting fake news than DeClarE model. Through these comparisons, a sub-task of our second objective which is to evaluate our new model against DeClarE model is achieved.

4.4.2 Unique Claim Evaluation

As discussed previously, a problem with both the PolitiFact and the Snopes datasets is that a claim has multiple articles associated with it. Therefore, there are multiple rows with the same credibility, claim and claim source data but different article data. This implies that there could be a repetition of the claims in the training and the test data when the data is split in an 80-20 ratio. This issue is tackled in this section, which also caters to the third and final objective of our research.

To capture the true performance of the model, for both the datasets, the results were evaluated for unique claims (claims that are only present in the test set) as well. The detailed claim-related information for both the datasets is given in Table 5.6. As can be seen in the table, the true and false claim division has a

difference of ~25 claims which is not much and would not cause any serious class imbalances.

Statistics	Snopes	PolitiFact
True Claims	52	65
False Claims	35	94
Total Unique claims	69	138
Total Unique Data (Article)	87	159
Total Test Data	3102	5912

Table 5.6: Unique claim statistics for PolitiFact and Snopes dataset

The results of the evaluation on the unique test set are given in Table 5.7. As done in the previous section, both the (Full) and (Article+Claim) configurations are considered for the Hybrid model. Even when only two inputs are supplied to the Hybrid model, which is in the (Article+Claim) configuration, the model handles unseen data better than the DeClarE model with an F1 measure in the range of 0.58-0.62. In the DeClarE model, the F1 measure for PolitiFact dataset is 0.52 meaning that the model is randomly predicting true or false values for the input data making it no better than a baseline model. Its performance is even worse for Snopes where it predicts everything as True and F1 is 0.28. This could be related to the fact that there are more true claims than false showing that the model does not generalize well to new data. When the two models are compared as whole, it can be seen that the Hybrid model performs better than DeClarE for both of its configurations. Out of the two configurations, (Article+Claim) has the best results with an F1 measure in the range of 0.58-0.62 where a Hybrid (Full) has a range of 0.57-0.6. This shows the advantage Hybrid (Article+Claim) has over DeClarE (Full) and points towards the robustness of the Hybrid model. But for both of the datasets, the metrics are not as good in predictions than the results given for the complete test set showing some discrepancy in the reported results. This shows that the model is not as effective on unseen data.

Metrics	DeClarE		Hybrid			
	Snopes	PolitiFact	Snopes (Full)	Snopes (A+C)	PolitiFact (Full)	PolitiFact (A+C)
True Claims Accuracy	100%	46.2%	69.2%	71.2%	56.9%	55.4%
False Claims Accuracy	0%	59.6%	51.4%	54.3%	57.4%	62.8%
F1 Measure	0.28	0.52	0.60	0.62	0.56	0.58
AUC	0.5	0.52	0.60	0.62	0.57	0.59

Table 5.7: Unique claims evaluation metrics. Here A+C is Article+Claim

The performance metrics on this fair evaluation show that the Hybrid model is much more robust and superior to the DeClarE model. However, even with a better accuracy, F1 and AUC, the performance of the Hybrid model is not as good as when the entire test data is used, having a difference of ~35%. Same is the case for the DeClarE model as it also shows a performance difference of ~15%. As the authors of the DeClarE paper did not provide details of the model's performance for unique inputs, the results cannot be compared.

4.5 Summary

In this chapter, we checked the reproducibility of the DeClarE model against the evaluations in the DeClarE paper. It was found that the model matched the results of the paper for PolitiFact datasets and lagged in the Snopes data. This model was also compared in terms of performance against the Hybrid model and through the evaluation, it was found that the Hybrid model far exceeded the DeClarE model both over the entirely of the test data and the on unique claims on both the datasets. This shows that the Hybrid model is not only more efficient but also handles unseen data better.

Chapter 5 Conclusion

In this last chapter, an overall review of the research is discussed. The objectives achieved, challenges faced and the impact of the evaluation are studied in Section 5.1. In the next section, possible improvements to the work and future prospects are considered for the model.

5.1 Discussion

This research project had two major aims: to recreate a state of the art model for fake news detection, DeClarE model, and secondly to develop a new model that overcomes the weaknesses of DeClarE while maintaining its strengths.

Reproducibility

For the first goal, the DeClarE model was developed based on the description provided in its corresponding research paper. Various challenges were faced during this process most of which related to the creation of the model using Keras. While the majority of its structure is paper accurate, many hyper-parameters are not the same. The learning rate is changed from 0.002 to 0.0001 as that yielded better performance. A major change from the paper is the pre-processing done to the datasets. The authors used a relevance score to select only those articles are similar to their claim which reduced the size of the dataset. We argue that this process reduced the model's capability to handle unknown data and hence its generalisability. Hence, the same was not attempted for this project, rather only down-sampling of the Snopes dataset is done to prevent the class imbalance problem. The results of the model showed that its performance nearly matched the paper for the PolitiFact dataset but not for the Snopes dataset. Various reasons for the reduced performance are discussed but a big contributing factor could be the dataset itself. By not providing one of the required inputs to the model, i.e., claim source, its performance was sure to be lagging behind a more balanced PolitiFact.

Novel Approach

The second goal of the project was to develop a more efficient and effective model than DeClarE. The Hybrid model created for this purpose exploits the structure of the DeClarE model by overcoming its weaknesses by including the model created for Self-Attentive Sentence Embedding (Lin et al. 2017). This model, implemented in PyTorch, also uses bidirectional LSTM but it is coupled with a self-attentive mechanism which improves the models focus on crucial features of an article to classify its credibility value. Another unique feature added to this model is the penalty term using Frobenius norm, introduced in the Self-Attentive Sentence Embedding paper, this norm handles the redundancy problems that would have occurred when using the same input twice. After initial difficulty related to the learning curve of understanding a new library, this model was easier to create. In its evaluation, this model showed very high accuracy (> 85%) scores for the two datasets leading to doubts about over-fitting the model. Even after changing the hyper-parameters, input structure and optimizers the model still yielded a high performance on the test set which I supposed to be unseen

data. A reason for this high performance could be related to the structure of the datasets themselves. A claim has multiple articles related to it. These articles could spill over to the test set giving the model a boost in performance. But in this scenario, the DeClarE model, that shares the same datasets, should also have a higher performance (> 80%) on the test set. As that is not the case and both models are evaluated on the same datasets, this enhanced score is seen as an advantage of the Hybrid model. When the two models are compared, the Hybrid model has much better performance (> 30%) than DeClarE on all three measurements: accuracy, F1 measure and AUC for both the datasets. It must be noted that accuracy is used instead of precision and recall, as is the norm, so that it can be easily compared to the results of the DeClarE paper. The authors of the paper did not provide the results of the precision and recall metrics. The fact that the Hybrid model has a better performance on the Snopes dataset a well as the PolitiFact dataset points towards its robustness.

Experimental Bias

Due to the experimental bias that may be introduced, the performances of the models are also compared on the unique data. Even in these results, the Hybrid model has shown better generalizability than DeClarE. But these performances are subpar when compared to the model's results on the entire test set, showing a far larger margin for improvement than first anticipated. However, it provides a fairer basis for comparison of the models.

5.2 Future Work

Looking at the average performance of the Hybrid model on unique claims and articles, it is quite clear that the model needs more refinement. One such change that can be added is the use of a Gated Encoder [Memisevic, 2013] referenced in the 'A Self-Attentive Sentence Embedding' paper to combine the output of the two Sentence Embeddings. The authors have used this structure on the SNLI dataset which provides a pair of inputs to the model and is used for a textual entailment task. The model created thereof is used to find any semantic contradiction between the pair of inputs. Currently, the Hybrid model is only able to classify an article as being true or false whereas news articles can also contain information that is partly true or false and cannot be placed in just one of two classes. The model can be improved to become a multi-class classifier. Once improved the Hybrid model can be integrated into different social media sites to filter out those articles classified as 'fake'. It can also be used by exiting fact-checking websites to reduce manual labour and time required to classify an article.

5.3 Summary

Through this project, a new model has been developed to help counter the spread of fake news online. A critical analysis of the performance of this new model has been done against the latest model in this field and it has shown promise, even though there is much room for improvement. In this final chapter, an overview of the research is presented which includes a short discussion on the final results and how they compare as a whole. Future improvements and areas where such a model can be utilized are also touched upon.

Chapter 6 References

Hunt Alloctt and Matthew Gentzkow, *Social Media and Fake news in the 2016 elections,* Journal of Economic Perspectives, Volume 31, Pages 211-236, 2017.

Hunt Allcott, Matthew Gentzkow and Chuan Yu, *Trends in the Diffusion of Misinformation on Social Media,* the National Bureau of Economic Research, Working paper 25500, 2019.

William Ford, *Numerical Linear Algebra with Applications*, Chapter- 7.2, 2015.

FullFact, *How do you fact check?,* About Full Fact, [Online] Available at: https://fullfact.org/about/frequently-asked-questions/ [Accessed 22nd August 2019]

Ian Goodfellow, Yoshua Bengio , Aaron Courville, *Deep Learning*, MIT Press, Page-410-411, 2016.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou and Yoshua Bengio, *A Structured Self-attentive Sentence Embedding*, ICLR, 2017.

Roland Memisevic, *Learning to relate images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume-35, Issue- 8, Pages- 1829 – 1846, 2013.

Kashyap Popat, Subhabrata Mukherjee Jannik Strötgen and Gerhard Weikum, *Credibility Assessment of Textual Claims on the Web,* CIKM '16 Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Pages 2173-2178, 2016

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates and Gerhard Weikum, *DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning*, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Pages 22-32, 2018.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova and Yejin Choi, *Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking,* Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Pages- 2931–2937, 2107.

Lydia Saad, Military, *Small Business, Police Still Stir Most Confidence,* Politics, news.gallop.com, 2018, [Online] Available at: https://news.gallup.com/poll/236243/military-small-business-police-stir-confidence.aspx [Accessed 4th August 2019].

ScienceDirect, *Vector and Matrix Norms*, [Online] Available at: https://www.sciencedirect.com/topics/engineering/frobenius-norm [Accessed 23rd August 2019]

Laura Silver, *In Western Europe, Public Attittudes Toward News Media more Divided by Populist Views Than Left-Right Ideology,* journalism.org, 2018, [Online] Available at: https://www.journalism.org/2018/05/14/in-western-europe-public-attitudes-toward-news-media-more-divided-by-populist-views-than-left-right-ideology/ [Accessed 3rd August 2019]

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin, *Attention Is All You Need*, https://arxiv.org, 2017

Queenie Wong, *Fake news is thriving thanks to social media users, study finds,* cnet, 2019 [Online] Available at: https://www.cnet.com/news/fake-news-more-likely-to-spread-on-social-media-study-finds/ [Accessed 3rd August 2019]

Jiawei Zhang, Bowen Dong and Philip S. Yu, *FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network,* https://arxiv.org/pdf/1805.08751.pdf , 2018 [Accessed 22nd August 2019]

Appendix A DeClarE Model Results with Data Preprocessing

An attempt was made to create the DeClarE model using the data processing suggested by Popat et al. (2018). The results for the same are given below along with the expected results as documented in the DeClarE paper.

Dataset	Snopes (sigma = 0.2)					PolitiFact (sigma = 0.5)			
Configuration	Baseline model	DeClarE (Plain)	DeClarE (Plain + Attn)	DeClarE (Plain + SrEmb)	**DeClarE (Full)**	DeClarE (Plain)	DeClarE (Plain + Attn)	DeClarE (Plain + SrEmb)	**DeClarE (Full)**
True Claims Accuracy	39.4%	47.80 %	0%	38.50 %	**15.90%**	0%	58.30%	0%	**77.00%**
Expected True Claims Accuracy	-	74.37 %	78.34%	77.43 %	**78.96%**	62.67 %	65.53%	66.71 %	**67.32%**
False Claims Accuracy	70.1%	84.70 %	100%	87.00 %	**96.00%**	100%	70.50%	100%	**56.60%**
Expected False Claim Accuracy	-	78.57 %	78.91%	79.80 %	**78.32%**	69.05 %	68.49%	69.28 %	**69.62%**
Macro F1 Score	0.54	0.664	0.431	0.635	**0.554**	0.361	0.644	0.361	**0.65**
Expected Macro F1 Score	-	0.78	0.79	0.79	**0.79**	0.66	0.66	0.67	**0.68**
AUC	0.547	0.662	0.5	0.627	**0.559**	0.5	0.643	0.5	**0.66**
Expected AUC	-	0.83	0.85	0.85	**0.86**	0.70	0.72	0.74	**0.80**

Table A.1: Performance for DeClarE model with data pre-processing.

In this, the learning rate is kept as 0.002 and the threshold (sigma) for similarity is kept as 0.5 for PolitiFact and 0.2 for Snopes. This similarity measure is used to filter out those articles that are not similar to their corresponding claim up to a defined threshold. In the paper, the authors kept this threshold as 0.5 for both datasets but that is not the case here. This is because, at 0.5, all the data from Snopes gets filtered leaving no data for training and testing. As can be seen from Table A.1, for both the Snopes and PolitiFact datasets the problem areas are highlighted in red. In DeClarE (Full) for Snopes, the true claims accuracy is very low leading to the AUC and F1 values to drop to 0.55. That means that the model is making near borderline random predictions which not what the expected values show (in grey). The DeClarE (Full) for PolitiFact has a better performance but has a 56% accuracy for false claims which is much less than the expected 69%. The model also gives 0% and 100% accuracy for some of the configurations for both datasets which is not ideal.

Appendix B Hybrid Model Performance with Adam optimizer

To compare the performance of the Hybrid model with various optimizers and to ensure that over-fitting is not occurring, the Hybrid model was trained using the Adam optimizer to see the performance impact on the unique dataset. Furthermore, non-randomized input was used in order to compare it better with the DeClarE model. The statistics of the unshuffled data is the same as DeClarE in Table 5.6. This change was only done for the (Article+Claim) configuration as that has yielded the best results till now. Table B.1 lists the performance on the complete test set.

Metrics	Snopes (Article+Claim)	PolitiFact (Article+Claim)
True Claim Accuracy	68.2%	93.8%
False Claim Accuracy	86.4%	91.9%
F1 Macro Measure	0.773	0.929
AUC	0.772	0.929

Table B.1: Evaluation metrics on the unshuffled complete test set.

PolitiFact has a smaller drop in performance as compared to Snopes when compared to the results in Table 5.4. The results of these configurations, on the unique claims, are given in Table B.2.

Metrics	Snopes	PolitiFact
True Claim Accuracy	45.7%	58.4%
False Claim Accuracy	73.8%	54.2%
F1 Macro Measure	0.595	0.556
AUC	0.597	0.563

Table B.2: Evaluation metrics on the unshuffled unique claims

As can be seen from both the tables, even with a change in the optimizers, the Hybrid model shows a higher F1 measure and AUC values for both the datasets than the DeClarE model when compared with the results in Table 5.2. The Snopes dataset does have an unbalanced accuracy but that could be attributed to the structure of the dataset being incomplete as stated in Section 4.2.1 under Results Discussion.

But as with RMSProp optimizer, the results for unique test data are not satisfactory leading to the conclusion that the model requires further improvement.

Appendix C Model Weights Visualization

In this appendix, we shall discuss the visualization weights of the Hybrid model. The weights visualized are the attention weights generated from the SEM portion of the model. Two instances are discussed, one, where the prediction was correct and the second where it was incorrect.

Case 1: Correct Prediction

Claim: *no net global warming decade*

Figure C.1 shows an example of a correct prediction made by the model. The claim is **false** and is predicted as **false.** This emphasis given by the weights on a term is shown by a red shading. The shade intensity shows how the model focused on each term. Darker shade means more focus and vice-versa. We can see it highlights words like '*grossly overstated'* as well as the word related to the claim as well.

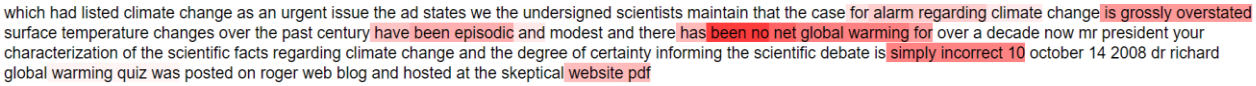

Figure C.1: Visualization of attention weights on an article for Case 1.

Case 2: Incorrect Prediction

Claim: *sixty million americans depend social security seniors america depend social security 90 percent income*

The claim is **true** but is predicted as **false** by the model. It focuses on words common to the claim such as *'60 million americans'* but as the article does not contain any positive or negative terms related to the claim, the model predicts it as the majority class with is false. This is an example of an incorrect prediction made by the model.

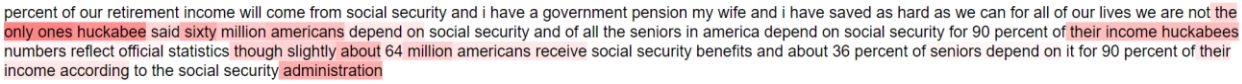

Figure C.2: Visualization of attention weights on an article for Case 2.

These visualizations prove that the two SEM sections of the Hybrid model are working as intended. The first SEM focuses on the terms common to both the article and claim. The second highlights the sentiment related terms.