Department of Computer Science

Submitted in part fulfilment for the degree of BSc in Computer Science and Mathematics.

Deep Learning for Detection of Rumors in Social Media

Soe Aung

2023

Supervisor: Tarique Anwar

To Tarique Anwar

Acknowledgements

I would like to thank Dr Tarique Anwar for all his insightful advice and comments throughout this project.

Contents

List of Figures

List of Tables

1 Executive Summary

Social media has become integrated into our lives and has made information and news consumption more convenient. It is also a double-edged sword in which there is now a massive spike in the spreading of fake news, a problem that is getting worse by the day. Almost 80% of American consumers reported seeing fake news related to the COVID-19 outbreak according to Statista [1].

Fake news travels faster on Twitter than official sources do by a substantial margin [2], and the danger of fake news is in its potential to manipulate public opinion for an agenda that is harmful to society [3]. A good example is the recent COVID-19 pandemic and false information surrounding vaccinations [4]. This is but one of many examples which show the need to authenticate news that is shared on platforms such as Twitter. There are now over 4.9 billion social media users in the world as of January 4th, 2023 [5]. Individuals and other entities not only rely on social media platforms to connect with others but also to consume information. As the online world grows, unreliable sources of information grow along with it. Fake news in recent years has become the most well-known blanket term for a wide variety of false information both online and in the media. Such examples include rumours, conspiracy theories, fake reviews, etc. In this paper, we are focusing on the early detection of rumours and we will use the terms 'fake news' and 'rumour' interchangeably due to their ubiquity in published systems and context.

At present, detection methods are based on news/social media content. In this dissertation, tweets are going to be used in the fake news detection task. To summarise, pre-trained BERT-based Systems were combined with a CNN, logistic regression or support vector machine trained on a real-world Twitter dataset called PHEME [6] were used to help classify the legitimacy of a social media post. The best results came from the Baseline-BERT model with an F1-Score of 0.828, precision of 0.829 and recall of 0.829. These results offer a solution to a problem that is detrimental to our society and make a strong case for BERT systems being further developed for the task of detecting rumours. All models are intended for use with the help of human moderators before any decision is made. False positives/negatives could have serious consequences. The author has no financial or personal motivation behind the project that would compromise its integrity.

2 Introduction

It is quite easy to find an example of the dangers that stem from spreading fake news. The spread of fake news was a significant factor in those who were reluctant to get vaccinated [7] during the COVID-19 pandemic, making vaccine intake harder to increase.

2.0.1 Motivation

There are already solutions available to fact-check information however, the sheer scale of fake news combined with the democratization of information is already so large that not only can an individual fall victim to it, but it is also impossible for the moderation of such content via human fact-checking organisations or moderators [8]. Thus arises the need to automate a solution to help debunk fake news. Building an effective model is difficult as detecting rumours early is a task that will be done with the constraint of a small amount of data to work with. And when it comes to machine learning tasks, the availability of data is key.

2.0.2 Background

This project aims to explore and create various systems for rumour detection. The systems should be able to detect rumours online with high accuracy and generalize as well as possible. Predictions should be made using data where news content tends to spread. The objectives are as follows:

- Review existing literature on rumour detection as well as the scope of fake news/rumours detection as a whole

- Propose methods to implement systems that can detect rumours being spread via a social media post

- Evaluate the systems against each other.

The solutions will build on previous research in rumour detection, with a particular focus on machine learning techniques.

2.0.3 Overview of Dissertation

Having established the background and motivation for this project, the following will give an outline of the structure for the rest of the dissertation.

Chapter 3 is a literature review that will discuss the current state of rumour detection research and focus on machine learning techniques.

Chapter 4 will go over the methodology and implementation of the solutions given.

Chapter 5 will outline the evaluation metrics and compares the results acquired from all systems.

Chapter 6 concludes the report by analysing the results given and considering possibilities for future research.

3 Literature Review

3.1 Rumours

Peterson et al. [9] defined rumours as "a tall tale of explanations of events circulating from person to person and pertaining to an object, event, or issue in public concern." News and social media content is a common focus for rumour detection methods, and anything deriving from it. There are various methods to tackle fake news such as automatic detectors or binary and multi-class classifiers. Rumours can take on various forms. They can use both factual and/or false information in their narratives. This makes the classification of rumours more complex as some rumours may be true, and some may be false. In other instances, a post online may not be a rumour at all but instead a satire piece. This distinction has been discussed by Golbek et al. [10]. Most research methods use Twitter data for training their models. This raises the issue of how well it can generalize as a model trained on Twitter content may be unable to perform on another social media site such as Facebook due to the structuring of posts. This was shown through Kochkina et al. [11] where a massive performance drop was seen in rumours when comparing unseen data of varying posted dates.

3.2 Natural Language Processing

Most methods use several natural language processing techniques. Natural Language Processing (NLP for short) is a branch of computer science or in particular, artificial intelligence, in which we are concerned with interactions between humans and computers. It is the core of several day-to-day programs such as chatbots or translators. It is not constrained to text but also makes use of speech. NLP techniques analyse syntax, semantics, pragmatics, and morphology from text and utilise them in classifying a piece of text. These methods rely on the analysis of textual content itself as opposed to verifying the claims outright, which is a contrast to fact-checking.

Computers cannot directly understand the text in articles and headlines so there has to be a way to convert the words into a numerical form that it can understand. *Bag of words* is an example of such a conversion. This method counts occurrences of each word in a given text block. It should be noted that the words counted must be identical, i.e., "hate" and "hated" will be treated completely differently despite the similarity. Toshevska et al. [12] mention how such methods have a deficiency in getting the necessary information. In place of such classical methods, we now see models converting words into vectors, allowing the model to make mathematical calculations on sentences while taking into account more complex information.

3.3 Supervised Learning

Supervised learning uses labelled datasets to train algorithms to predict outcomes for unseen data and is said to be the most important methodology in machine learning [13]. Input data is fed into a model and has weights adjusted until the model has an appropriate fit. It has a lot of use in deep learning algorithms such as neural networks.

3.3.1 Logistic Regression

With origins that date back as far as the 19th century [14], logistic regression is a statistical model often used in classification problems. Based on a dataset of some independent variables, it will estimate the probability of some event occurring. The outcome will be between 0 and 1 since it is a probability. There are multiple kinds of logistic regression. A binary logistic regression will only have two possible outcomes of 0 or 1. A multinomial logistic regression can have three or more outcomes. The logistic function is of the form:

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}} \tag{3.1}$$

where μ is the value of the function's midpoint and s is the scale parameter which determines the spread of the probability distribution.

Logistic regression has uses in tasks such as fraud detection or disease prediction. It can also be used in the context of rumour detection. For example, Reddy et al. [15] use logistic regression to perform a sentiment analysis on Amazon product reviews, a topic that is not too dissimilar to rumour detection. Bangyal et al. [16] apply logistic regression amongst 7 other machine learning algorithms on a binary labelled COVID-19 dataset to detect fake news, one of these methods included a support vector machine.

3.3.2 Support Vector Machines

Introduced by Boser et al. [17], a support vector machine (SVM) is a set of machine learning algorithms that analyze data for classification and regression analysis. They have an advantage in that overfitting is generally not a problem and can handle high-dimensional data. Joachims's [18] analysis concludes that SVMs get good results from text classification. SVMs will separate data with hyperplanes and find the most appropriate hyperplane, the one farthest from data points or the one with a maximised margin. The equations used in SVM are such that:

$$w * x_i - b = 1 \text{ if } y_i \geq 1 \tag{3.2}$$

$$w * x_i - b = 1 \text{ if } y_i \leq 1 \tag{3.3}$$

$$\forall i; y_i(w + b) \geq 1$$

Where w is the normal vector to the hyperplane, x is the vectorised input data and b determines the offset from the origin. When b is positive, the offset is in the direction of w and when b is negative, the offset is in the direction of $-w$.

3.4 Deep Learning

Deep learning is a class of machine learning algorithms that use multiple layers to extract more features from an input. This method can be traced back to 1943 when Mcculloch et al. [19] created a computer model based on the neural networks of the human brain. Most deep learning models will be based on neural networks. Neural networks are a class of machine learning models made up of neurons that are connected by weights. These weights are updated during a training process, in which the network will try to learn patterns on a given data set. There are many kinds of neural networks each with their pros and cons. Convolutional neural networks are a form of neural networks which typically are used in the classification of images. They are also quite capable of NLP workloads. There are already proposed solutions that for example, utilise convolutional neural networks for rumour detection. Ashgar et al. [20] used a BiLSTM-CNN which was trained on the PHEME dataset [6]. This type of model, instead of training a single model, uses two models. The first model learns the sequence of the input given. The reverse of that sequence is learned by the second model. These are known as forward and backward long-term-short-term memory.

3.5 Transfer Learning For Deep Learning

Transfer learning is a machine learning technique in which a pre-trained machine learning model is applied to a different but related problem [21]. The general idea is to use the knowledge of another model learned from a task using a lot of training data in another task where data is not as plentiful. Transfer learning is used a lot in natural language processing tasks due to the amount of computational power that is needed. A lot of research on rumour detection makes use of transfer learning, take Choudhry et al. [22] for example. A popular pre-trained model in rumour detection is BERT, which comes with several variants.

3.5.1 BERT

BERT stands for 'bidirectional encoder representations from transformers'. It is a language representation model that was developed by Google researchers [23] and introduced in 2018. BERT allows for the use of sentence embeddings for text-based data, an approach that will take the semantic information of sentences and turn them into vectors of a numerical type to pass onto a model that will learn the context of the data. The base model for BERT is already big, with the total parameters numbering around 110 million. The larger version has around 340 million parameters. Several published models get promising results from BERT for language tasks, such as the fakeBERT model [24] or Kar et al.[25] who used BERT as a base and combined it with other models such as an MLP and, Heidari et al.[26] who used BERT on social bots to detect fake news using a COVID-19 dataset.

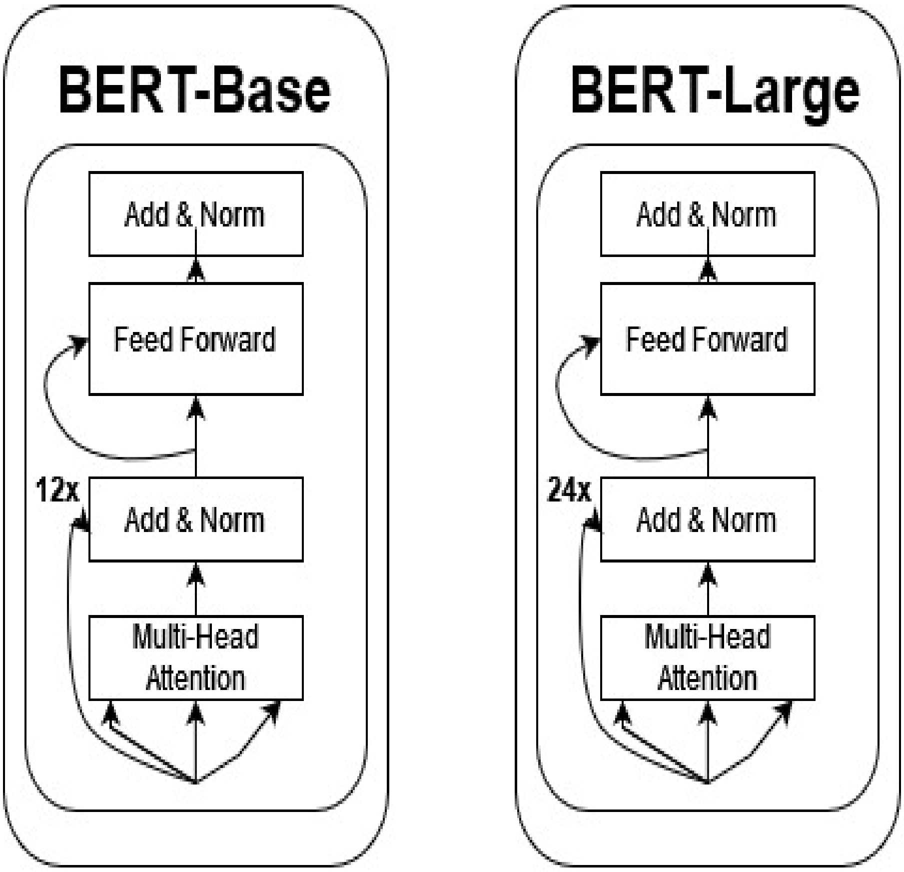

Figure 3.1: Bert Architecture for BERT-Base and BERT-Large.
BERT large has 24 layers of encoders stacked on top of each other instead of 12.

4 Methodology And Implementation

4.1 Methodology

4.1.1 Dataset

There does not seem to be a golden standard dataset for the task of rumour detection however, there are some commonly used datasets such as the PHEME dataset [6] or the Twitter 15/16 dataset [27]. For this reason, the proposed models were tested with the PHEME dataset [6], a Twitter dataset from 2016 based on various topics/threads which were a hot spot for the spreading of rumours. It should be noted that other papers may implement a binary model, where 0 may indicate a non-rumour and 1 will indicate a rumour. However, this experiment extends the problem to multi-class classification. The labels are numbered between 0 to 3. 0 represents a false-rumour, 1 represents a true-rumour, 2 represents a non-rumour (this could be from an official source or could just be a statement such as "end police brutality") and 3 represents an unverified-rumour. All working models will be using data with textual features. The dataset extracted does include some numerical features however in practice only the plaintext tweets and labels were actually used in the working models. We will discuss more on using other features in a later section.

Table 4.1: PHEME dataset labels.

False-Rumour	True-Rumour	Non-Rumour	Unverified	None
636	1065	4018	696	1

Table 4.2: A sample of a cleaned tweet with only lowercase text and its label.

Text	Label
"breaking least dead injured gunman open fire offices charlie hebdosatirical mag published mohammed cartoons"	1

4.2 Proposed Models

4.2.1 Baseline BERT

As a base point of comparison, we took a pre-trained uncased BERT model to classify the tweets. The implementation was based on a BERT classifier for news article classification [28].

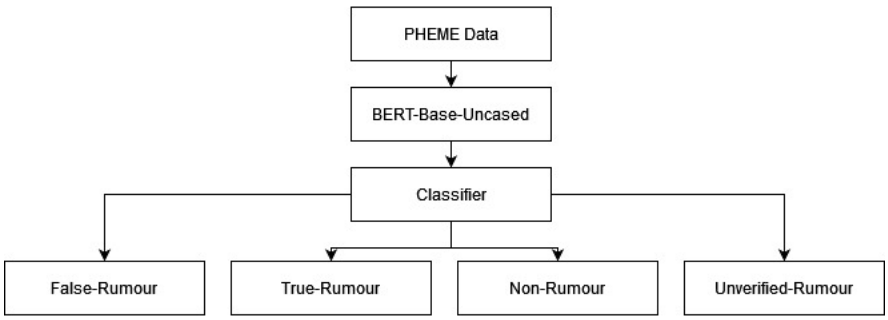

Figure 4.1: Baseline-BERT-Model

4.2.2 BERT CNN

This model extracts the BERT embeddings from the pre-trained BERT model and then passes it onto a convolutional neural network. Originally Lim's binary model [29] was used as a base and then extended to a multi-class implementation. The CNN architecture used here is also entirely different.

4.2.3 BERT Logistic Regression

This model extracts the BERT embeddings from the pre-trained BERT model and then passes it onto a logistic regression.

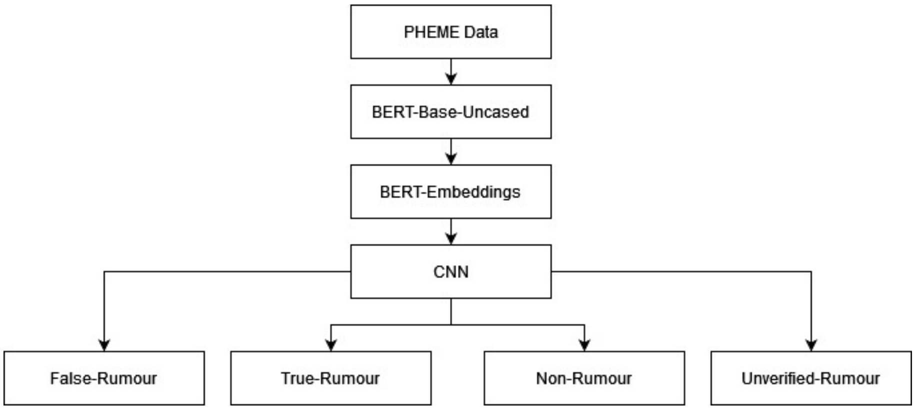

Figure 4.2: BERT-CNN-Model

Layer Type	Input	Output
Conv1d	64	128
Batchnorm	128	128
Dropout	128	128
Conv1d	128	256
Batchnorm	256	256
Conv1d	256	512
Batchnorm	512	512
Linear	6144	728
Linear	728	256
Linear	256	4

Table 4.3: CNN Architecture

4.2.4 BERT SVM

This model extracts the BERT embeddings from the pre-trained BERT model and then passes it onto a support vector machine.

4.3 Implementation

4.3.1 Tools

The models were written in Python inside a Google Colab environment. Colab is a jupyter notebook environment that runs inside the cloud and is used for deep learning workloads among other tasks. The bulk of the code utilizes methods from the Pytorch and scikit-learn libraries. Pytorch

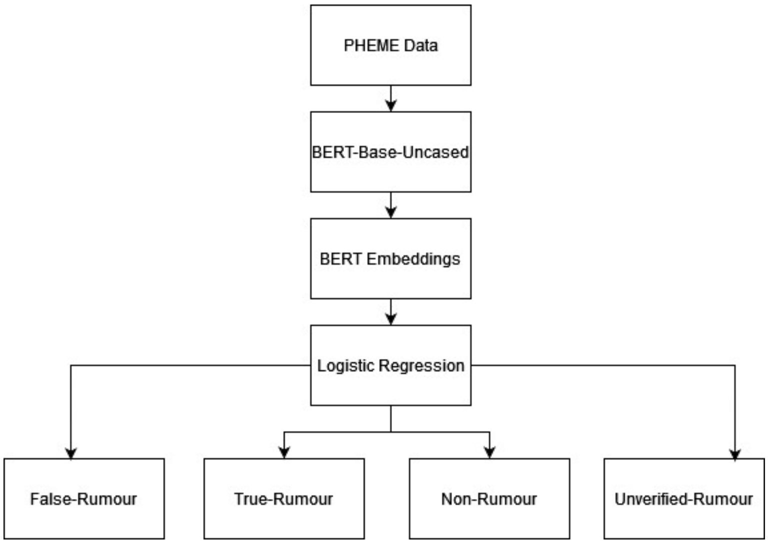

Figure 4.3: BERT-Logistic-Regression-Model

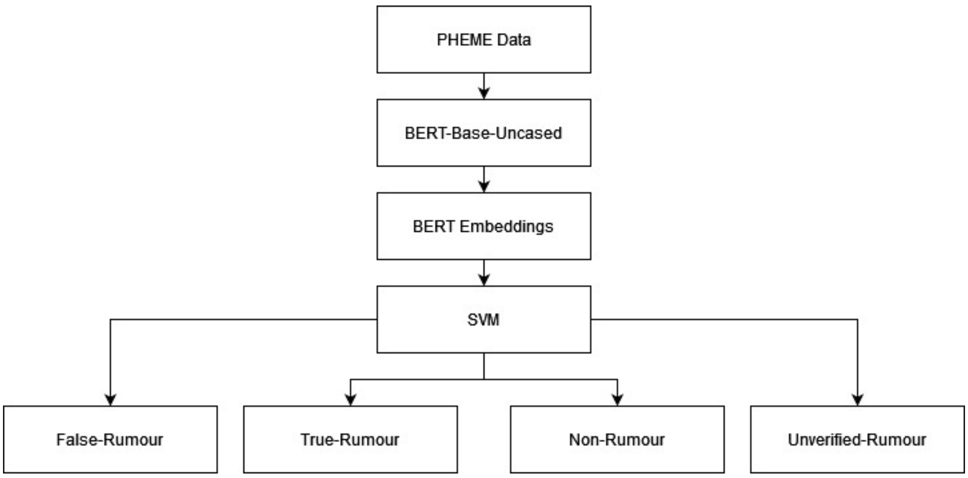

Figure 4.4: BERT-SVM-Model

was picked over alternatives such as Keras due to the wider available community support and easier debugging process. Numpy was also used for mathematical operations. scikit-learn provided useful evaluation metrics such as precision, recall, and f1-scores. Finally, the hugging face library was used to load a pre-trained BERT model. A Tesla-T4 GPU provided by Colab was used due to a lack of a physical GPU.

4.3.2 Data Pre-Processing

All tools for this experiment were coded in Python. The first thing required is to get the data into a readable format. The PHEME [6] dataset had a JSON structure with various annotations including the tweet id, date and time of a tweet, the tweet content, and what class of rumour it was. To extract the tweets, a Python script was used to access all relevant directories of the dataset folders, extract all the annotations and content, and finally convert it into a CSV file. As there were multiple threads, another script was used to concatenate each thread together into one big dataset. All working models used plaintext tweets and labels. The tweets were further cleaned afterwards using the preprocessor library from Pypi. A single tweet will be made up of a lowercase, emoji, and punctuation-free plaintext.

4.3.3 BERT Embeddings

In order for text to be read we need to convert it into vectors that can be understood by our models and then passed on for training. After loading the data, a process similar to Lewis-Lim's [29] pre-processing was used to obtain BERT embeddings and batch data for the plaintext tweets, being passed through a pre-trained uncased BERT model and outputting a 3d tensor of Nx64x728. N denotes the number of rows, 64 is the max length of sentences taken during the process and 728 is the number of hidden layers in BERT.

4.3.4 Training and Testing

Training and testing for the baseline BERT model and BERT-CNN were done with the torch and torch.nn packages. The BERT-Logistic-Regression model used the scikit-learn Logistic Regression method from the scikit-learn linear model module. The BERT-SVM used the support vector class from the scikit-learn SVM module. All models have an 80:20 training and test ratio. All random seeds are set to 42 for each model in order for the results to be reproducible. The testing process counts the number of correct predictions made by the model over the total number of predictions.

4.3.5 Hyperparameter Settings

The following subsection shows all configurations for each proposed model. Each model was tested with every possible combination of the hyperparameters shown in the tables below. Note that the bold values are the most optimal of the values given.

Hyperparameter	Value
Epoch	5 10 **15**
Dropout	**0.5**
Batch Size	**2**
Learning Rate	1e-4 **1e-5** 1e-6
Optimiser	**Adam**
Loss Function	**Cross Entropy**
Activation Function	**ReLU**

Table 4.4: Hyperparameter settings for Base-BERT model

Hyperparameter	Value
Epoch	5 10 **15**
Dropout	**0.5**
Batch Size	**2**
Kernel Size	**4**
Stride	**4**
Learning Rate	1e-4 1e-5 **1e-6**
Optimiser	**Adam**
Loss Function	**Cross Entropy**
Activation Function	**ReLU**

Table 4.5: Hyperparameter settings for BERT-CNN model

Hyperparameter	Value
Iterations	**100**,200,300
Penalty	**L2**
Solver	Newton-CG, **Limited-Memory-BFGS**

Table 4.6: Hyperparameter settings for BERT-Logistic-Regression model

Hyperparameter	Value
Iterations	100,200,**300**
Penalty	**L2**
Kernel	Radial-Basis-Function, **Polynomial**, Sigmoid

Table 4.7: Hyperparameter settings for BERT-SVM model

5 Evaluation and Results

5.1 Evaluation

5.1.1 Evaluation Metrics

This rumour detection task is a multi-class classification problem (True-Rumours, False-Rumours, Unverified-Rumours, and Non-Rumours). The Evaluation metrics used here will be the F1-Score, Precision, Recall, and Accuracy measures. Weighted F1-Score will be used as this is a multi-class problem. It also provides a better idea of performance due to the imbalance of the dataset.

For the following formulae below let TP denote True Positives, FP denote False Positives and FN denote False Negatives.

Precision

For each of the classes X, precision is defined as the amount of true positive cases divided by the sum of true positives and false positives as shown by equation (5.1).

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

Recall

For each of the classes X, recall is defined as the number of true positives divided by the sum of true positives and false negatives as shown by equation (5.2).

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

F1-Score

The F1-score is a harmonic mean of precision and recall denoted by equation (5.3) and should always have a value that is in between the precision and recall measures.

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.3}$$

5.2 Results

The aim was to create systems that can detect online rumours spread via social media posts early on. The table below shows the performance of all working models with metrics calculated to 3 significant figures.

BERT embeddings take up the majority of the computation in all proposed systems. While this could have been addressed with the use of a smaller variant of BERT such as DistilBERT [30], the trade-off would be the model performance.

Table 5.1: Model Comparisons

Model	F1-Score	Precision	Recall
Baseline-BERT	0.**828**	**0.829**	**0.829**
BERT-CNN	0.449	0.359	0.599
BERT-Logistic-Regression	0.705	0.708	0.722
BERT-SVM	0.615	0.637	0.630

Given that there are 4 classes, the average expected accuracy from a random guess would be 25%. All models exceeded this accuracy floor. The Baseline-BERT model outperforms all other models with a f1-score of 0.828. The BERT-Logistic-Regression suffered in performance mainly due to the massive data imbalance skewed towards non-rumours. The same problem occurred with the BERT-SVM as SVMs perform poorly on imbalanced datasets. This was even more evident in the BERT-CNN while over the baseline accuracy of 25%, got there by predicting non-rumours too frequently. The logistic regression may have been unable to surpass the baseline model as it is hard to obtain complex relationships with logistic regression. It is also rather difficult to make a conclusion from just one simple plaintext sentence. For this reason, attempts were made to combine some numerical features from the dataset such as the follower counts, retweets, or favourited tweet counts. The idea was to combine all numerical features into one tensor and then concatenate it with the tensor containing BERT embeddings. Due to time constraints, this was not implemented, however, the hypothesis was that having these extra features would allow the model to understand the data better. Xiong et al. [31] for example combined both textual and image features in their fake news detection systems.

6 Conclusion

Rumours are a complex matter to both detect and moderate, the methods used here allow for an automated means to combat their spread. In this dissertation, transfer learning with the use of a BERT model combined with other networks or functions was proposed to detect rumours on social media accurately. The textual inputs were based on BERT to obtain embeddings which were then fed into logistic regression. Overall two out of four models performed well. These results were obtained with just textual features. The techniques used here were tested on the PHEME dataset. Performance was measured through a weighted F1-Score, precision and recall. All models were trained in social media posts in English so targeting other languages may not work as intended. However, it is possible to train on a dataset in another language provided the structure is consistent with the aforementioned models.

Future works could utilize several other kinds of features, such as a network of reactions to tweets connected to particular tweet IDs for a more effective learning process than merely inferring from the headline of a tweet. The PHEME dataset [6] contains the tweet IDs and the tweet IDs of reactions to a selection of headlines and would be a viable dataset for such a model.

A Dataset Dimensions

Table A.1: Dataset dimensions for PHEME.

Full Dataset	Train	Test
6146x2	4109x2	2036x2

B Confusion Matrices

The following confusion matrices have the true positives excluded to illustrate the problem with labelling non-rumours correctly.

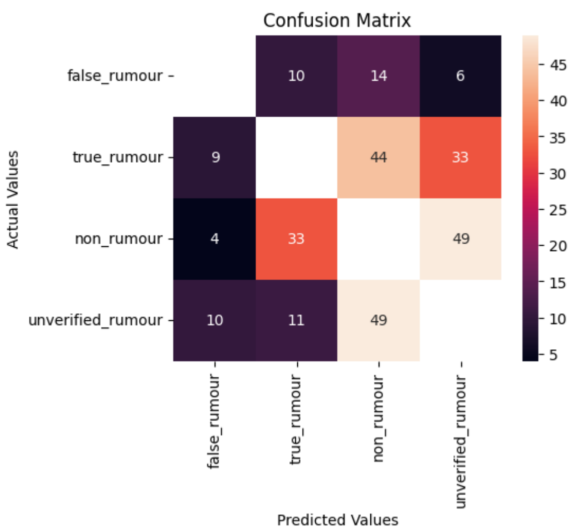

Figure B.1: Baseline-BERT-Confusion-Matrix

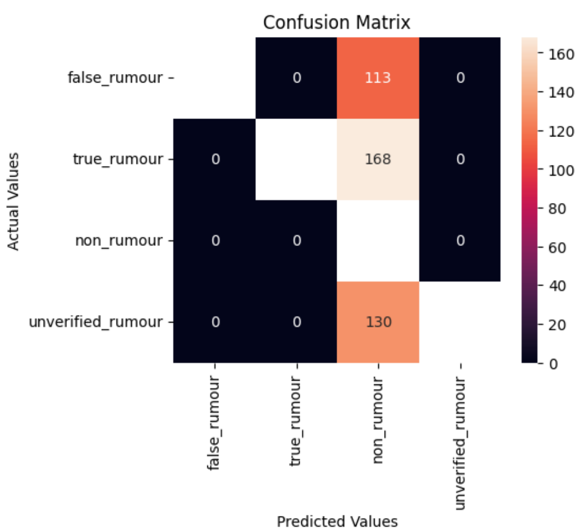

Figure B.2: BERT-CNN-Confusion-Matrix

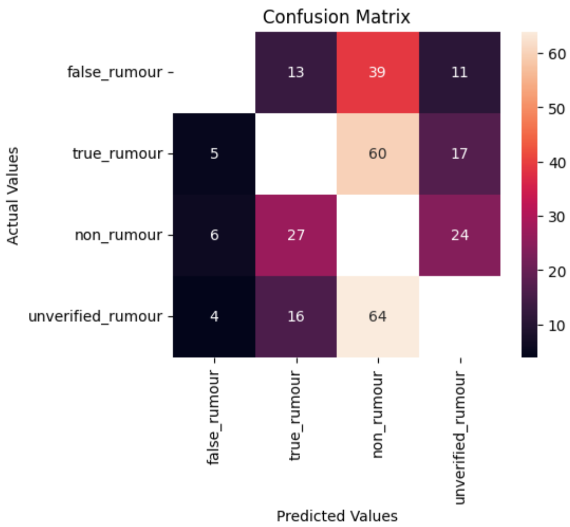

Figure B.3: BERT-Logistic-Regression-Confusion-Matrix

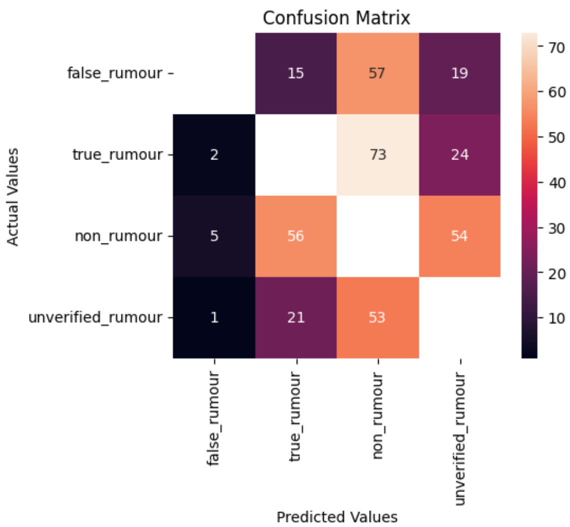

Figure B.4: BERT-SVM-Confusion-Matrix

C Footnote

Due to time constraints, the hypothetical BERT-CNN model that combines both numerical and textual features was worked on alongside the project parallel Tuba Yamak.

Bibliography

[1] A. Watson, ‘Fake news in the u.s. - statistics & facts,’ 2021.

[2] P. Dizikes, ‘Study: On twitter, false news travels faster than true stories,’ 2018.

[3] S. Oxenham, ‘Fake news shapes our opinions even when we know it’s not true,’ 2016.

[4] J. Goodman and F. Carmichael, ‘Coronavirus: False and misleading claims about vaccines debunked,’ 2020.

[5] D. Ruby, ‘Social media users in the world — (2023 demographics),’ 2023.

[6] A. Zubiaga, G. Wong Sak Hoi, M. Liakata and R. Procter, ‘Pheme dataset of rumours and non-rumours,’ 2016.

[7] S. Coughlan, ‘Whatsapp rumours fear over bame covid vaccine take up,’ 1947.

[8] N. Walter, J. Cohen, R. L. Holbert and Y. Morag, ‘Fact-checking: A meta-analysis of what works and for whom,’ *Political Communication*, 2020.

[9] W. A. Peterson and N. P. Gist, ‘Rumor and public opinion,’ *American Journal of Sociology*, vol. 57, no. 2, pp. 159–167, 1951.

[10] J. Golbeck, M. Mauriello, B. Auxier *et al.*, ‘Fake news vs satire: A dataset and analysis,’ in *Proceedings of the 10th ACM Conference on Web Science*, 2018, pp. 17–21.

[11] E. Kochkina, T. Hossain, R. L. Logan IV *et al.*, ‘Evaluating the generalisability of neural rumour verification models,’ *Information Processing & Management*, 2023.

[12] M. Toshevska, F. Stojanovska and J. Kalajdjieski, ‘Comparative analysis of word embeddings for capturing word similarities,’ *arXiv preprint arXiv:2005.03812*, 2020.

[13] P. Cunningham, M. Cord and S. J. Delany, ‘Supervised learning,’ *Machine learning techniques for multimedia: case studies on organization and retrieval*, pp. 21–49, 2008.

[14] J. S. Cramer, ‘The origins of logistic regression,’ 2002.

[15] P. S. Reddy, D. R. Sri, C. S. Reddy and S. Shaik, 'Sentimental analysis using logistic regression,'

[16] W. H. Bangyal, R. Qasim, N. U. Rehman *et al.*, 'Detection of fake news text classification on covid-19 using deep learning approaches,' *Computational and mathematical methods in medicine*, 2021.

[17] B. E. Boser, I. M. Guyon and V. N. Vapnik, 'A training algorithm for optimal margin classifiers,' in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[18] T. Joachims, 'Text categorization with support vector machines: Learning with many relevant features,' in *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings*, Springer, 2005, pp. 137–142.

[19] W. S. McCulloch and W. Pitts, 'A logical calculus of the ideas immanent in nervous activity,' *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[20] M. Z. Asghar, A. Habib, A. Habib, A. Khan, R. Ali and A. Khattak, 'Exploring deep neural networks for rumor detection,' *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[21] S. Bozinovski, 'Reminder of the first paper on transfer learning in neural networks, 1976,' *Informatica*, vol. 44, no. 3, 2020.

[22] A. Choudhry, I. Khatri, M. Jain and D. K. Vishwakarma, 'An emotion-aware multitask approach to fake news and rumor detection using transfer learning,' *IEEE Transactions on Computational Social Systems*, 2022.

[23] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, 'Bert: Pre-training of deep bidirectional transformers for language understanding,' *arXiv preprint arXiv:1810.04805*, 2018.

[24] R. K. Kaliyar, A. Goswami and P. Narang, 'Fakebert: Fake news detection in social media with a bert-based deep learning approach,' *Multimedia tools and applications*, 2021.

[25] D. Kar, M. Bhardwaj, S. Samanta and A. P. Azad, 'No rumours please! a multi-indic-lingual approach for covid fake-tweet detection,' IEEE, 2021.

[26] M. Heidari, S. Zad, P. Hajibabaee *et al.*, 'Bert model for fake news detection based on social bot activities in the covid-19 pandemic,' in *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE, 2021.

[27] J. Ma, W. Gao and K.-F. Wong, 'Detect rumors in microblog posts using propagation structure via kernel learning,' in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[28] R. Winastwan, 'Text classification with bert in pytorch,' Towards Data Science, 2021.

[29] S. L. Lim, 'The early bert catches the worm: Utilising bert and convolutional neural networks for early fake news detection,' 2021.

[30] V. Sanh, L. Debut, J. Chaumond and T. Wolf, 'Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter,' *arXiv preprint arXiv:1910.01108*, 2019.

[31] S. Xiong, G. Zhang, V. Batra, L. Xi, L. Shi and L. Liu, 'Trimoon: Two-round inconsistency-based multi-modal fusion network for fake news detection,' *Information Fusion*, vol. 93, pp. 150–158, 2023.