FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

MisInfoBot2: fight misinformation on social media

João Filipe Carvalho de Araújo

Mestrado em Engenharia Informática e Computação

Supervisor: Carlos Manuel Milheiro de Oliveira Pinto Soares

Second Supervisor: Vitor Rolla

Third Supervisor: João Costa

August 1, 2022

MisInfoBot2: fight misinformation on social media

João Filipe Carvalho de Araújo

Mestrado em Engenharia Informática e Computação

August 1, 2022

Resumo

O crescimento das redes sociais nos últimos anos tem sido muito grande. Esse crescimento acentuou os seus aspetos positivos, tal como comunicação fácil com muitas pessoas ao mesmo tempo, mas também os negativos, como a dispersão mais fácil de informação errada, às vezes involuntariamente. Como as pessoas usam informação para tomar decisões, a desinformação pode levar as pessoas a decisões erradas, colocando em perigo as próprias, pessoas próximas ou mesmo a população em geral.

Recentemente, foram propostas duas abordagens para este problema, no âmbito do projeto *Misinfobot* [49]. A primeira abordagem decompõe o problema em duas etapas, prevendo se uma afirmação é falsa e, caso seja, retornando informações de contradição, obtida de um corpus de referência. A segunda abordagem adaptou uma arquitetura de deep learning originalmente proposta para deteção de contradições, para determinar se o documento em análise (por exemplo, um post no Twitter) e o documento de referência (por exemplo, informações da OMS) são corrobatórios, contraditórios ou neutros. No entanto, ambas as estratégias têm algumas limitações, tal como não conseguir encontrar informação útil para contradição e detetar contradições onde não ocorreram. Para ultrapassar essas limitações, neste projeto combinamos os dois métodos numa abordagem inovadora. Mais especificamente, usamos o método de detetar desinformação da primeira abordagem e, ao aplicarmos aos textos que foram identificados como falsos um detetor de contradição associado a uma lista de desinformações, melhoramos o método de detetar contradições da segunda abordagem.

Este método foi avaliado empiricamente em dados públicos relacionados com a vacinação contra a covid e comparado com as abordagens propostas no projeto *Misinfobot* [49]. Os resultados mostram que este método tem potencial para contribuir na luta contra a desinformação nas redes sociais.

Abstract

In recent years a tremendous growth in social media popularity has been observed. Although this situation has led to the increase of positive aspects, such as easy communication with several people at the same time, it has also promoted some negative aspects, such as the enhanced spread of wrong information, sometimes involuntarily. Since people use information to make decisions, misinformation may lead to wrong decisions, which, ultimately, can endanger themselves, their close ones, and the general population itself.

Recently, two approaches to circumvent this problem have been proposed in Misinfobot [49]. The first approach decomposes the problem in two steps, finding a false statement and retrieving contradiction information from a reference corpus. The second approach adapted a deep learning architecture, originally proposed for contradiction detection, to determine if the statement document (i.e., a Twitter post) and the reference document (i.e., information from the WHO) are entailed, contradictory or neutral. However, both strategies present some limitations, such as of not being able to retrieve useful and accurate information and detecting contradictions where they didn't occur. To surpass the above mentioned limitations we combined the two methods in an innovative approach; specifically, we used the misinformation detection of the first method and, by applying a contradiction detection classifier associated to a misinformation list to the texts marked as false, improved the contradiction retrieval of the second method.

The efficacy of this new method was empirically evaluated in public data related to covid vaccination, and duly compared with Misinfobot [49]. The outputs generated show that the developed tool has a good potential to contribute to fight against misinformation in social media.

Acknowledgements

First, I would like to give my sympathies to everyone who has lost a loved one due to COVID. May they never be forgotten.

I want to thank my supervisors, Carlos Soares, João Costa and Vitor Rolla, who have guided me in what would have otherwise been an arduous task. The only reason this dissertation came out as good as it did was thanks to their advice.

I also want to thank my parents, who always ensured I had everything I needed. I wouldn't be who I am now without them and all my family. Plus, a big thanks to all my friends that always answered my stupid questions and kept me company these five years.

João Araújo

"If there is no struggle, there is no progress."

Frederick Douglass

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers
BiLM	Bidirectional Language Model
BoW	Bag of Words
CBoW	Continuous Bag of Words
CT-BERT	COVID-Twitter-BERT
CNN	Convolutional Neural Network
DL	Deep Learning
ELMo	Embeddings from Language Models
FN	False Negatives
FP	False Positives
GloVe	Global Vectors for Word Representation
IDF	Inverted Document Frequency
LM	Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Inference
NLP	Natural Language Processing
POS	Part Of Speech
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TL	Transfer Learning
TN	True Negatives
TP	True Positives
WHO	World Health Organization

Chapter 1

Introduction

Social Media are rapidly rising technologies that have significantly impacted our society. They have changed everything on a global level, from politics to communication. They have also changed most aspects of our social life, from relationships, information collection (from news or other sources), ease of communicating with distant friends, finding like-minded people, and socializing. Twitter is considered one of the most popular platforms available and had 217 million monetizable daily active users in 2021 [74].

Unfortunately, together with the above mentioned benefits, other negative issues also appeared. One of them, which will be this project's focus, is **misinformation**. Misinformation, according to one of the definitions of the Cambridge English Dictionary, is "wrong information" [24]. Social Media are one of the main tools for misinformation spread, due to how fast and easy information can be disseminated to a large audience without verification, thus making it hard to contain.

In Figure 1.1, we have the (at the time) president of America retweeting a video. It claimed that masks were not necessary and claimed that hydroxychloroquine (used to treat malaria) was an effective treatment for COVID-19 [80], both of which had already been contradicted by health officials [26][13]. Due to how easy it is to spread incorrect information and due to the lack of fact-checking (both from the author of the post and from the platform used to distribute it), this false information reached a broad audience. This caused many people who followed this advice to later end up in the hospital, flooding an already overloaded medical system even more. Although it is hard to get exact numbers, it is safe to assume that lives were lost due to that.

Thus, we need to find ways to identify and rebut misinformation on Social Media to prevent its spread and to contribute to people's education.

> "(...) Public and private sector, in particular #socialmedia platforms, media, individuals - we all have a role to play to end this pandemic and infodemic." - Tedros Adhanom Ghebreyesus, Current Director-General of the World Health Organization (2022) [29]

Figure 1.1: Trump tweet with misinformation [56]

1.1 Motivation

Although some platforms have strategies to deal with misinformation, the solution is far from perfect. Most of the times the distinction is made manually, which is extremely difficult to upscale due to the amount of information that needs to be checked.

Misinformation detection has become more accessible due to techniques such as **Natural Language Processing (NLP)**, a branch of AI which helps computers understand text and spoken words. It is often used in conjunction with **Deep Learning (DL)** to have better results. One recent approach to misinformation detection using those techniques has been **Misinfobot** [49], a Twitter bot that used two approaches for detecting misinformation related to COVID. The first method used a Supervised Learning Classifier model to mark the text as true or false. If marked as false, it is sent to an Information Retrieval system that searches for the related reference document/information from a reference corpus. The second method used a Contradiction Retrieval approach to calculate the relationship between a tweet and a document from the reference corpus by seeing if they are entailed, contradictory or neutral, returning the documents with contradictions. Although Misinfobot [49] managed to get good results when identifying misinformation in tweets in the first method, neither of the methods was able to systematically retrieve documents with information that is adequate for the misinformation posts, with the second method detecting contradictions where they did not occur. This can be attributed to several factors, such as the

reference corpus not containing the topics mentioned in the tweets of the dataset, the lexical and semantic difference between both of them, the small size of the dataset, and the significant class imbalance between contradiction, entailment, and neutrality.

1.2 Main Objectives

The main objective of this dissertation is the creation of **Misinfobot2**, a variation of Misinfobot [49] that addresses its limitations. Additionally, we will extend the empirical evaluation to misinformation about COVID-19 vaccination.

In order to achieve this goal, we will be expanding Misinfobot further by combining the two methods developed. We will do this by using the misinformation detection of the first method as a first step and the contradiction retrieval of the second method for the ones marked as false as a second step.

This dissertation has three main contributions. One is the misinformation detection method developed, which can later be adapted to other types of misinformation. Another is an empirical validation on COVID-19 vaccination data. The last one is a Twitter bot that uses the developed method to identify misinformation and retrieve the related reference document to rebut it.

1.3 Document Structure

This document is organized into six chapters. In Chapter 2, several concepts about this dissertation are presented. They include supervised learning models, transfer learning, performance metrics, and NLP. We also discuss the related work in the area. The method implemented and the approach used to implement it are described in Chapter 3. In Chapter 4 we describe the datasets and experimental setup used, as well as briefly introduce the experiments that will be performed. We analyze the results of those experiments in Section 5. In Section 6 we summarize this project and its contributions and outline possible future improvements.

Chapter 2

Background and Related Work

In this chapter, we analyze the background needed to better understand the project's key concepts and the related work done in those areas. In Section 2.1, we will describe the classification algorithms used in this project, briefly mention some projects using other algorithms and explain some performance metrics used to evaluate the approaches. We will introduce Natural Language Processing, including text representations and misinformation detection, in Section 2.2. In Section 2.3, we give a brief overview of the current work on fighting COVID-19 misinformation on Social Media. We give a short summary of this chapter in Section 2.4.

Machine Learning (ML) is a sub-field of **Artificial Intelligence (AI)** that aims to have computers simulate the way humans learn. It does this by training algorithms to leverage data in order to improve its performance in a task based on previous experiences. It can be divided in three main areas:

- **Supervised Learning**, where the training data the model receives is already labelled with the desired output;
- **Unsupervised Learning**, where the training data is unlabelled and the model tries to structure the data by grouping or clustering it;
- **Semi-supervised learning**, which receives training data that is mostly unlabelled, but some of it is labelled. It combines the approaches of the previous two methods.

2.1 Supervised Learning

A very popular ML type of approach for NLP classification, which will be the approach used in this project, is **Supervised Learning**. In Supervised Learning, the input data received is labeled with the desired output. The model is trained to predict the correct output from the given training data input. Then, it outputs an inferred function, which can later be used to predict new inputs that do not belong to the training dataset.

Trying to make the model predict the training data too accurately (making it fit the training data too much) can lead to overfitting, making the model memorize the training dataset instead of learning how to predict the outputs. It should be avoided to increase generalization.

2.1.1 Classification Algorithms

Classification is a type of Supervised Learning where the labels belong to a finite set of categorical values (the category numbers have no relationship between them). Formally, after receiving the extracted features x from the input, we send it to the classifier f to predict the label y, $y = f(x)$. The labels are internally represented by a finite set of categorical values L, $y \in L$, with N possible values. If there are only two categories, $N = 2$, we consider the problem as binary classification. If there are more than two categories, $N > 2$, but each input can only be assigned one label, $|y| = 1$, the problem is considered as multi-class. If each input can be assigned multiple labels, $|y| > 1$, we consider the problem as multi-label classification.

This project will focus on **Deep Learning (DL)** models. However, other commonly used algorithms for classification are Support Vector Machine (SVM), Decision Tree, Random Forest and Naive Bayes. DL algorithms typically use **Neural Networks (NN)**, which are inspired by the human brain's neural networks since they consist of multiple layers of interconnected nodes. To make it easier to explain, we use NN based on multilayer perceptrons.

A weight exists between every node of a layer and all nodes in the layer before and the layer after, and there is a bias between each layer. The network changes the values of the weights and biases to solve the classification task. A NN has multiple types of layers, which are also shown in Figure 2.1. Formally, in a node with 2 inputs, x_1, x_2, and 1 output, y, it becomes $y = m_1 \times x_1 + m_2 \times x_2 + b$, with m_1, m_2 being the weights of the respective inputs and b being the bias of the layer.

- **Input Layer** — Receives the inputs; there can only be one of these;
- **Hidden Layers** — There can be zero or more of these, and they are located between the input and output layers. They calculate the weighted sum of the inputs and weights, add the bias and execute the activation function. Therefore, the more we add, the slower the program becomes and the more complex the model becomes;
- **Output Layer** — Outputs the final result.

Neural Networks also have hyperparameters, which are parameters that determine how the network is trained and the network structure. They are set before training and include the number of hidden layers and units of each layer and the learning rate, which defines how much the network changes the weight values. Hyperparameters also include the number of epochs, which define how many times the algorithm processes the whole dataset, and the batch size, which defines how many samples we need to give to the network until it updates its internal parameters (affects the frequency of the updates). Formally, for a learning rate, λ, and a loss function, L, we can calculate the update rule for the new weight with the formula $w^{(i+1)} = w^{(i)} - \lambda \frac{dL}{dw}$ at the end of each batch.

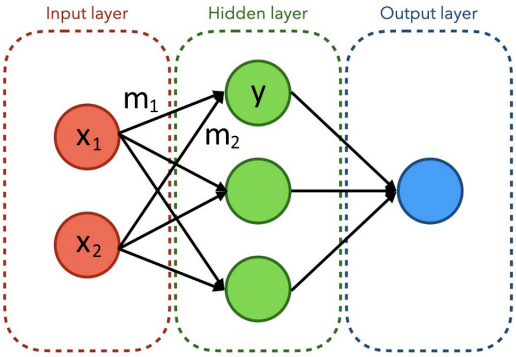


Figure 2.1: Example of a Neural Network (adapted from [50])

Several architectures can be chosen for the NLP context, and right now, some of the more popular are **Recurrent Neural Networks (RNN)**, **Convolution Neural Networks (CNN)** and **Transformer-based NN** [52].

- **Recurrent Neural Networks** — Data is processed sequentially, making it especially useful for NLP since the meaning of a text' strongly depends on the order of its words. They use the output from hidden layers as inputs alongside the standard inputs of hidden layers. The architecture can be seen in Figure 2.2, where we can see the hidden layer (A) of a previous iteration sending its output as an input to the hidden layer in the next iteration;

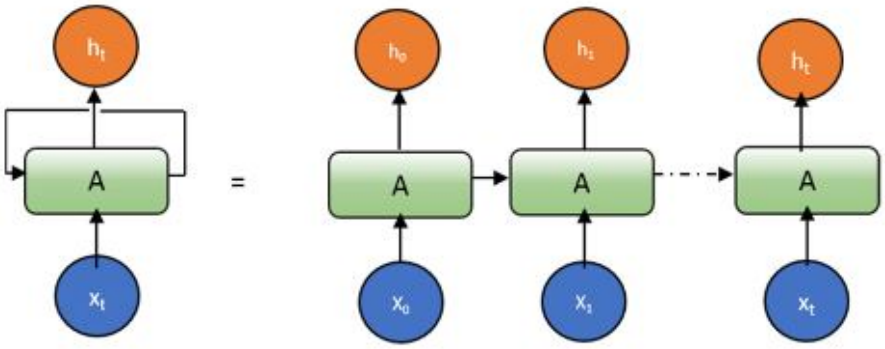


Figure 2.2: Example of a Recurrent Neural Network (reproduced from [4])

- **Convolutional Neural Networks** — They are most commonly used for images [76], and they have three main types of layers: Convolutional layers, which convolve (compute the dot product) the filters across the input (tensor); Pooling layers, which reduce the dimensions of the data by combining the outputs of clusters; and one Fully-Connected Layer, which performs the classification task. The Fully-Connected layer is always the last of the hidden layers, and the rest of the layers consists of alternating Convolutional and Pooling Layers (no two of the same type of layer in a row). An example of this architecture can be seen in Figure 2.3.

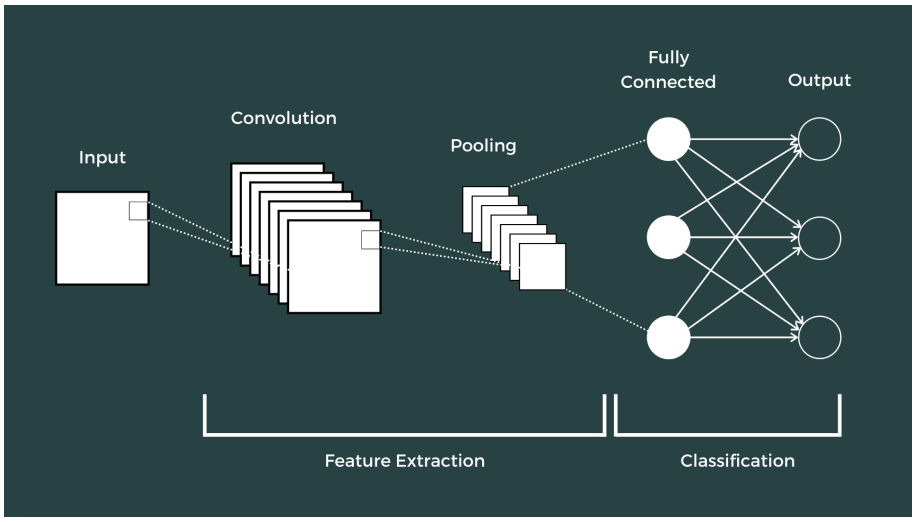


Figure 2.3: Example of a Convolutional Neural Network (reproduced from [61])

2.1.1.1 Transformer

Recently, the Transformer, which uses an encoder-decoder architecture, was created, and it was shown to outperform both RNN and CNN [78], hence it is the approach we will be using in this project. It was released by Google in 2017 and has since grown in popularity [87]. They can reduce computation time because the Transformer architecture allows for more parallelization than RNNs, which need to process the words in a sequential order [43]. This reduction is due to relying on a multi-headed self-attention mechanism since it allows the model to compute every word in a sentence at once (in parallel) by providing context for any position in the input. We can see it represented in Figure 2.4, where the left half contains the encoder and the right half contains the decoder. Each encoder layer generates encodings containing information about which parts of the inputs are relevant to each other and then passes it to the next encoder layer as input. The decoding layers do the exact opposite by receiving the encodings and generating an output sentence [77].

2.1.2 Transfer Learning

Transfer learning refers to taking the knowledge learned by a model while solving one problem (source domain) and then applying it to a different but similar problem (target domain), with domain referring to the area of interest. This is possible since the low-level features the model learns are not specific to a dataset or task. This way, we can reduce the size of the dataset required, thus avoiding a high effort in data-labeling, boost the performance of a learner if it is from a related domain [48] and prevent the new network from starting from scratch. For a more in-depth look at Transfer Learning, [83] and [53] can also be seen. We can separate transfer-learning techniques into four types by the source domain's label availability and the target domain.

- **Transductive Transfer Learning** — Occurs when we have labeled data in the source domain and no labeled data in the target domain;

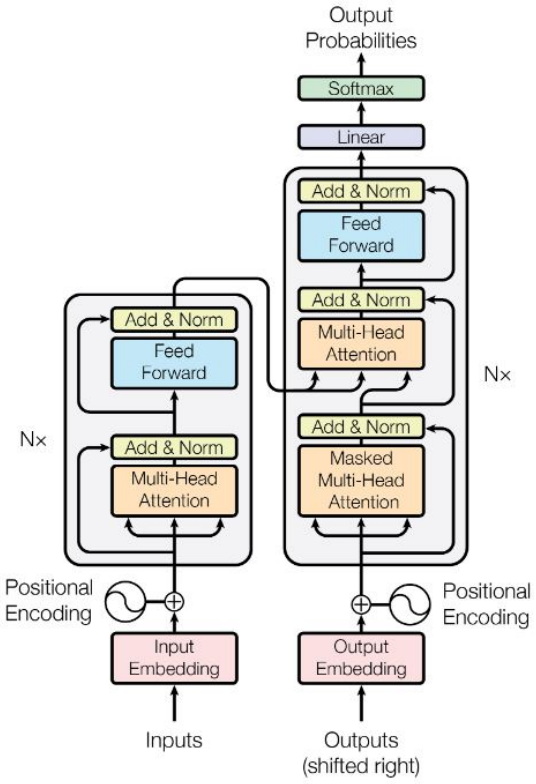


Figure 2.4: Example of a Transformer (reproduced from [77])

- **Inductive Transfer Learning** — Occurs when the tasks in both the source domain and the target domain are different, whether or not the domains are the same;
- **Cross-Modality Transfer Learning** — Occurs when we are transferring between different mediums, such as from image to text. The label spaces between the source and the target domain can also be different;
- **Unsupervised Transfer Learning** — Occurs when there is no labeled data in both the source and target domains;

If the performance reduces after Transfer Learning occurs, we call that Negative Transfer Learning. It happens when too much unrelated knowledge is transferred from the source domains. We can also separate according to the types of knowledge transferred, such as:

- **Instance-transfer** — Tries to reuse knowledge from the source domain by re-weighting the source domain instances;
- **Feature-representation-transfer** — Tries to minimize domain divergence by finding good feature representations. The BERT model, which is used in this project (Section 2.2.1.3), uses this type of representation;

- **Parameter-transfer** — Assumes that they share some parameters or hyperparameters between both tasks. Thus, it is not suitable if there is a significant domain shift;
- **Relational-knowledge-transfer** — Attempts to handle data that is not independent (that has a relationship with other data points). Assumes that source domain data has a relationship similar to the relationship that data of the target domain has.

2.1.3 Performance Metrics

Performance estimation consists of trying to predict how the model will perform (the loss it will incur) when receiving a set of unseen data. Performance metrics are what we use to evaluate the model's performance on the tasks. The most commonly used performance metrics are based on the confusion matrix, which we can see in Figure 2.1 exemplifying a problem of misinformation detection. The Predicted class represents what the classifier predicted, and the Actual class represents the actual values of the data (the ground-truth).

		Actual value		
		Misinformation	Real	Total
Predicted Value	Misinformation	TP	FN	All Predicted as Misinformation
	Real	FP	TN	All Predicted as Real
	Total	All Misinformation	All Real	All

Table 2.1: Example of a Confusion Matrix

We will use the problem addressed in this project to explain the metrics. Since our goal is to find misinformations, we will consider the positive class the news marked as misinformation and the negative class the news marked as real. **TP (True Positives)** represents the number of texts that were predicted correctly as being misinformation, while the **FP (False Positives)** represents the texts that were incorrectly predicted as misinformation. **TN (True Negatives)** are the texts that were correctly predicted as being real, and **FN (False Negatives)** represent the texts that were incorrectly predicted as real. The sum of the four values gives us the total number of predictions made.

With these four values, we can now calculate several metrics that we can use to evaluate performance. We have the **Accuracy** (Eq. 2.1), which is the percentage of correct results in relation to the total number of cases. This makes it not ideal for unbalanced datasets since it can lead to a high accuracy with a model only predicting the class with the higher occurrence. There is also the **Precision** (Eq. 2.2), which is the percentage of the true positives against all predicted as positive, and can be used to see which percentage of the predicted positives are actual positives. We also have **Recall** (Eq. 2.3), which is the percentage of positives that were classified correctly, and the **Specificity** (Eq. 2.4), which is the percentage of the negatives that were classified correctly. Then we have the **F1 Score** (Eq. 2.5), which is calculated by the harmonic mean between the Precision and Recall. Lastly, we have the **Fallout** (Eq. 2.6), which is the percentage of negatives that were predicted as positive in relation to all actual negatives.

$$Accuracy: \quad \frac{TN+TP}{TP+FP+FN+TN} \tag{2.1}$$

$$Precision: \quad \frac{TP}{TP+FP} \tag{2.2}$$

$$Recall/Sensitivity: \quad \frac{TP}{TP+FN} \tag{2.3}$$

$$Specificity: \quad \frac{TN}{TN+FP} \tag{2.4}$$

$$F1Score: \quad \frac{2*Precision*Recal}{Precision+Recall} \tag{2.5}$$

$$Fallout: \quad \frac{FP}{TN+FP} = 1-Specificity \tag{2.6}$$

Another important parameter to consider is the **ROC (Receiver Operating Characteristic) Curve**, which shows the diagnostic ability of a classifier (how well the model can predict the class) as its discrimination threshold is changed. The discrimination threshold is the probability that a class score (probability of the input belonging to that class) has to exceed so that an object is considered to belong to that class. It is obtained by plotting the Recall (True Positive Rate) against the Fallout (False Positive Rate) at various threshold settings. The **Area Under the Curve (AUC)** metric shows us how much the classifier can distinguish classes since it is the probability that the classifier ranks a random positive example higher than a random negative example. This is usually a commonly used measure to compare models since it is scale-invariant (only considers the order of the predictions, not their actual value) and does not depend on the classification threshold chosen. However, since it is scale-invariant, it will not tell us how well calibrated the outputs are, and the classification threshold invariance is not desirable if we want to reduce one type of classification error due to different costs.

When calculating metrics for multi-class and multi-label problems, we cannot use the binary-average to calculate the precision, recall and F1-score. Instead, we use the micro-average, which consists of calculating the total values for the confusion matrix (for example, FP is the total number of classes incorrectly predicted as belonging to the positive class). Other possible values for the aggregation of the results for multiple classes involve calculating the metric for each individual label (as if each class was the positive class, one at a time). Then, in the case of the macro-average, we calculate their unweighted mean (so it does not take into account label imbalance). In the case of weighted, we calculate the weighted average of the metric for each label.

Lastly, there is also the Hamming Loss, which calculates the Hamming distance between two sets of samples (the minimum number of changes that are necessary to change one sample into another). Since it is a Loss function, the goal is to minimize it.

2.2 Natural Language Processing

Natural Language Processing (NLP) is a sub-field of AI that helps computers analyze and interpret natural human language with the goal of understanding it and being able to manipulate it. The evolution of Social Media has drastically changed the amount and types of NLP available. The considerable growth in the amount of data available and fast development in the ML field has led to several new possibilities for text-related applications. Since most algorithms are more suitable for numerical data, NLP approaches need numerical representations of text. As these representations directly affect the model's accuracy, several techniques to convert words to their numerical representation are described in Section 2.2.1.

2.2.1 Text Representations

One of the first methods of text representation was **Bag-of-Words (BoW)**, which stores the frequency of the words in a text document, but not their order or grammar. There are also **N-grams**, which work like the BoW, but store the frequency for a sequence of N items from the original text (thus, the BoW can be considered a 1-gram).

Another method of text representation was **Term Frequency-Inverse Document Frequency (TF-IDF)**, a statistic that measures the importance of a word in a corpus. It does this by multiplying the frequency of the words in the document (Term Frequency) with the weight of the word, which is assigned in an inverse relationship to its frequency in the whole corpus (less frequent terms have a larger weight) (Inverse Document Frequency).

In the meantime, NLP performance has been boosted with the use of distributed representations of words. Since the previous two representations ignore the semantic meaning of the word, they are not as predominantly used. We now have Word Embeddings, which can encode the syntactic and semantic relationship of words. Word Embeddings consist of mapping the words to a vector space based on their meaning, making the words with a similar meaning expected to be closer together in the vector space. They can be **Non-Contextualized** (if the word representation is independent of the context; homonyms will have the same representation), or **Contextualized** (if they take the context of the word into account when representing it). The representation that we will use on this project is **Bidirectional Encoder Representations from Transformers (BERT)**, which uses Contextualized Word Embeddings.

2.2.1.1 Non-Contextualized Word Embeddings

There are two main models for non-contextualized word embeddings.

One of them is **Word2Vec**, a NN developed by Google that can use one of two architectures:

- **Continuous Bag of Words (CBoW)** — It chooses a word from a phrase and then uses the surrounding context words to predict the current word. The order of the context words does not influence the prediction of the model;

- **Skip-Gram** — Uses the current word to predict target context words. It weighs nearby context words more heavily (gives them more importance) than the more distant context words;

They have their advantages and disadvantages since (according to the author) CBoW is faster, but Skip-Gram is better for infrequent words [30].

There is also **Global Vectors for Word Representation (GloVe)**, which was developed at Stanford and derived semantic similarity from a co-occurrence matrix. It does this by measuring the number of times a word has co-occurred with other words. It is better than Word2Vec because it also relies on word co-occurrence besides the local context information.

2.2.1.2 Contextualized Word Embeddings

For this type of embedding, a word's representation relies on its context. Therefore, homonyms now have a different representation since their context is different.

We have **Long Short-Term Memory (LSTM)**, which uses RNN and maintains long-term dependencies. They alleviate the vanishing gradient problem (long-term gradients which are back-propagated tending to zero) by allowing the gradients to also flow unchanged. However, since they cannot be parallelized, they are not fast.

We also have **Embeddings from Language Models (ELMo)**, which is the result of a Bidirectional Language Model (BiLM) pre-trained on a vast corpus (it combines a forward and a backwards LSTM). It uses a deep contextualized word representation that considers both syntax and semantics. The word vectors generated are computed from the hidden states of the LSTMs for each token.

2.2.1.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT was designed by Google and published in 2018 [23]. It is transformer-based and designed to pre-train bidirectional representations from unlabelled text, taking the left and right context into account in all layers. This makes it a potent model, able to be fine-tuned with just an additional output layer and able to achieve state-of-the-art results in NLP.

It mainly consists of two stages, pre-training and fine-tuning. BERT's **pre-training** was made on unlabelled data from BooksCorpus (800M words) and English Wikipedia (2,500M words)[23] using two different tasks: **Masked Language Modelling (MLM)**, where some of the tokens are masked and BERT tries to predict them from the context, and **Next Sentence Prediction (NSP)**, where given two sentences, BERT predicts the probability of the second sentence following the first sentence. The **fine-tuning** consists of plugging in the specific inputs and outputs into BERT and then fine-tuning the parameters [68].

BERT uses a sequence of tokens to represent a sentence or a pair of sentences. It uses WordPiece Embeddings with a vocabulary of 30,000 tokens[23]. The first token of the sequence is always the special classification token [CLS]. Classification tasks use the [CLS] final hidden state as an aggregate sequence representation. If we receive a pair of sentences instead of just one, the

token [SEP] is used to separate them and an embedding is added to each token to indicate whether they belong to the first or second sentence. For each token, we sum the corresponding token, position, and segment embeddings to calculate its input representation.

There is also **RoBERTa** [39] (Robustly Optimized BERT Pretraining Approach), a model developed by Facebook that was introduced in 2019 and is based on BERT, with the following modifications: it removes Next Sentence Prediction from the pre-training tasks; it trains with much larger learning rates and mini-batches on a bigger set of data; it dynamically changes the masking pattern of the Masked Language Modelling task whenever they send a sequence to the model (therefore, in different epochs the same masked token will have a different mask).

2.2.2 Natural Language Inference

Another topic of note within NLP is **Natural Language Inference (NLI)**, also known as Recognizing Textual Entailment. It is the task of identifying whether the relationship between a premise and a hypothesis received is a contradiction, entailment, or neutral.

Two very popular benchmark datasets for NLI are SNLI [10] and MultiNLI [86]. They were both developed by Stanford University. SNLI has 570k sentence pairs labelled as entailment, contradiction, or neutrality. In [89] they use it to test SemBERT, where they integrate explicit contextual semantics in BERT models to improve performance in natural language understanding. Multi-NLI has 433k pairs and, besides containing the labels of textual entailment, it also contains a label for its genre (for example fiction). In [37] they use it to test ALBERT (A Lite BERT), where they use two parameter-reduction techniques to lower memory consumption and increase the training speed and use self-supervised loss to improve tasks with multi-sentence inputs.

2.3 COVID-19 Misinformation Detection

In order to detect COVID misinformation in social media, several approaches have been developed, with most of them focusing on Twitter. Some of them also use news articles from fact-checking websites for training ([59], [25] and [1]).

Some approaches, like [45], [73] and [3], decided to focus instead on sentiment analysis for COVID-related tweets (identifying whether the tweet had a positive, negative or neutral attitude regarding the theme), and [36] focused on both. This can also help in misinformation detection, as shown by [5].

They used several of the ML techniques, such as Transformer-based, RNN ([25] and [1]), SVM, and Linear Regression, with most of the non-DL methods being used for comparison purposes. [25] and [21] used an Ensemble of methods to try to combine their strengths.

In terms of the word representation chosen, the most commonly used were BERT ([59] and [21]), RoBERTa ([90] and [7]), their derivations ([79] and [20]) and GloVe ([25] and [1]).

A few of the approaches also focus on the information besides the tweet text. Some try to focus on network-based methods by analyzing the propagation, but they only work after the misinformation has fully spread. To deal with this problem, [60] uses a source-based method that

analyses the community of the poster (including followers and retweets) while also analyzing propagation and profile on the side. [22], on top of analyzing the tweet's content, uses a heuristic post-processing technique that analyses the username of who posted and the URLs contained in the tweet. They calculate the probability of the username posting misinformation or the URL containing misinformation by seeing how often they spread misinformation (in the dataset).

These approaches only attempt to identify misinformation. However, it is also important to provide the correct information and support for it. Other than Misinfobot [49], that we go into detail in Section 1.1, the only other work that we found that separated the misinformations found in the COVID context by topic was [82], where they used a Knowledge Graph in order to represent the misinformation topics. However, they only identified the topic without trying to rebut it.

2.4 Summary

In this chapter, we briefly reviewed the background and state-of-the-art for this project's topics.

We talked about the different algorithms used in supervised learning, focusing on the Deep Learning ones since they have better performance and will be this project's focus. We also talked about the types of transfer learning, and about performance measures.

For textual representations, we will use BERT or BERT variants.

We also talked about the work that has already been done in the area of preventing COVID misinformation.

Chapter 3

Methodology

In this chapter, we outline the methodology that we will use to solve the problem. In Section 3.1, we specify the notation we use to describe the problem. The method implemented and the approach used to implement it are described in Section 3.2. We give a short summary of this chapter in Section 3.3.

3.1 Problem Formulation

As previously mentioned, we implement a Twitter bot that identifies misinformation in tweets and rebuts it with the related reference information. To do this, we, as proposed in MisinfoBot [49], decompose the problem of dealing with misinformation into two sub-problems: identifying if the information received is true or fake (Section 3.2.1) and, in case it is fake, retrieving rebuttal information (Section 3.2.2). For classifying if the information is true or false, we use the classifier of Misinfobot's first approach (Classify & Retrieve). For retrieving rebuttal information, we retrieve information that contradicts the misinformation from a reference corpus using the model from Misinfobot's second approach (Contradiction Retrieval). A picture summarizing the scheme can be seen in Fig 3.1.

Mathematically formalizing it, we have a statement document, sd, and a collection of Y Reference Documents $RD = \{RD_1, RD_2, ..., RD_Y\}$. We want to retrieve a reference document rd so that $rd \in RD$ and rd contradicts sd.

We used the COVID-19 vaccination context as a case scenario to test this approach, using tweets related to vaccination against COVID and misinformations associated with that topic. However, the solution should be easily adaptable to other misinformation topics, such as politics or vaccination for other diseases, by changing the dataset and learning models.

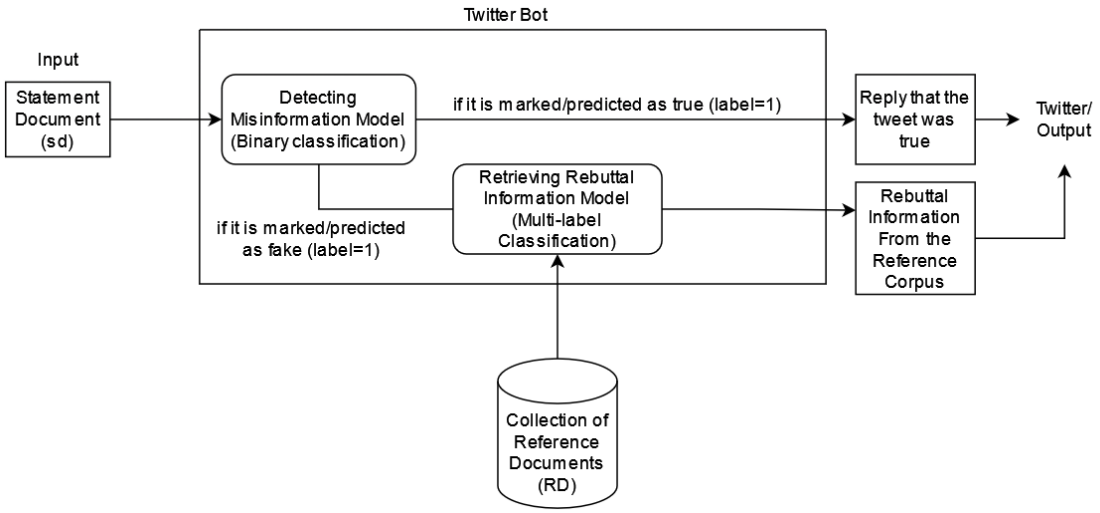


Figure 3.1: Representation of the scheme

3.2 Method Description

3.2.1 Detecting Misinformation

We can summarize the Detecting Misinformation sub-task as a Supervised Learning Classification problem. So, we use a classifier model that we fine-tune to classify the document received as true or fake. Since there are only two possible outcomes, it can be considered a binary classification problem. The model will calculate the probability of the text being true or fake by attributing a score to it (1 if true, 0 if fake). We use a threshold of 0.5 if we want to obtain the class with the highest probability .

Mathematically formalizing it, the model, M_f, receives a statement document, sd as input, such that the model outputs a score, $M_f(sd) \in [0,1]$, and then a threshold is used to define the class (e.g. 0.5).

If the text received is marked as false, it is then sent to the Retrieving Rebuttal Information component (Section 3.2.2) to retrieve the correct rebuttal information.

3.2.2 Retrieving Rebuttal Information

In order to retrieve the rebuttal information for the Statement Document received, we developed two approaches. The first, called Contradictions to Reference Documents, tries to find contradictions between the Reference Documents and the Statement Document and is described in Section 3.2.2.1. The second, called Entailments to Misinformations, tries to find entailments between the Statement Document and the misinformation topics and is described in Section 3.2.2.2.

3.2.2.1 Contradictions to Reference Documents

We treat the Contradictions to Reference Documents approach as a Supervised Learning problem that identifies contradictions between the text received and the reference documents. Each pair of the text received and one of the reference documents will be classified between contradiction, entailment or neutral. Hence, it can be considered a multi-class classification problem. The pre-trained model will be fine-tuned to calculate the probability of the text contradicting each reference document and then return the reference document with the highest contradiction score value. This reference document will later be returned as an answer for the received tweet.

Mathematically formalizing it, the model, M_c, receives a collection of Y pairs, P, made between a statement document, sd, and a collection of Y Reference Documents $RD = \{RD_1, RD_2, ..., RD_Y\}$ as input, such that $P = \{(sd, RD_1), (sd, RD_2), ..., (sd, RD_Y)\}$. The model outputs a score, $M_c(p) \in [0,1]$ for each pair $p \in P$ for whether that pair contains a contradiction or not, and then the Reference Document from the pair with the highest score is chosen.

3.2.2.2 Entailments to Misinformations

In order to try to avoid some problems Misinfobot [49] had, such as the tweets mentioning topics that did not appear in the reference corpus, we created the Entailments to Misinformations approach. We treat this approach as a Supervised Learning problem that identifies entailments between the text received and the misinformation topics. Each pair of the text received and one of the misinformation topics will be classified between contradiction, entailment or neutral. Hence, it can be considered a multi-class classification problem. The pre-trained model will be fine-tuned to calculate the probability of the text entailing each misinformation topic and then return the misinformation topic with the highest entailment score value. This misinformation topic will then be converted to one of the Reference Documents that contradict it. That Reference Document will be returned as an answer for the received tweet.

Mathematically formalizing it, the model, M_e, receives a collection of Y pairs, P, made between a statement document, sd, and a collection of Y misinformation topics $MT = \{MT_1, MT_2, ..., MT_Y\}$ as input, such that $P = \{(sd, MT_1), (sd, MT_2), ..., (sd, MT_Y)\}$. The model outputs a score, $M_e(p) \in [0,1]$ for each pair $p \in P$ for whether that pair contains an entailment or not, and then the misinformation topic from the pair with the highest score is chosen.

3.2.3 Complete Method

In the end we combine the predictions of both sub-tasks (Detecting Misinformation and Retrieving Rebuttal Information), forming a multi-label problem (with 0 being the label for a document being marked as true). Then we calculate the performance metrics.

Mathematically formalizing it, the complete method, M_{comp}, upon receiving a statement document, sd, and a collection of Y misinformation topics $MT = \{MT_1, MT_2, ..., MT_Y\}$ as input, will output a label prediction y, $y = M_{comp}(sd)$, such that $y \in \{0, Y\}$

3.3 Summary

This method aims to identify misinformation on a given text and rebut it with the correct information. To do this, we separate those tasks in two supervised learning approaches.

To detect if the text provided contains misinformation, we use a binary classifier that marks the text as true or fake. If the text is marked as fake, we describe two approaches: one tries to find contradictions between Reference Documents and the Statement Document; the other tries to find entailments between misinformation topics and the Statement Document.

We use true tweets and reference documents and misinformation topics related to COVID 19 vaccination in our implementation to try to tackle the current infodemic about COVID vaccination on Twitter.

Chapter 4

Experimental Setup

In this chapter, we will describe the experimental setup used. In Section 4.1, we describe the experiments made. In Section 4.2, we describe the datasets used and how they were gathered. In Section 4.3 we describe the experimental setup, such as the models, hyperparameters, and frameworks used, as well as the experiments made to evaluate the methods described in Chapter 3.

4.1 Experiments

In this project, we have two main research questions:

- **RQ1** — Can the Contradictions to Reference Documents approach present a better result than the original approaches developed by Misinfobot [49]?
- **RQ2** — Can we produce better results by using the Entailments to Misinformations approach?

In order to solve **RQ1**, first we fine-tuned all the models using the hyperparameters described in Section 4.3.2. We use the Statement Documents Dataset for the Detecting Misinformation model and the Annotations Dataset for the Rebuttal Retrieval model, and find the best performing model one (for the metrics of the test set) for both steps. Then, we fine-tune a copy of the best performing models in Misinfobot [49] datasets and compare the results obtained from the complete method with Misinfobot previous results.

In order to solve **RQ2**, first we also fine-tuned all the models using the hyperparameters described in Section 4.3.2. We use the Statement Documents Dataset for the Detecting Misinformation model and the alternative Annotations Dataset (the one using Entailment to misinformation topics) for the Rebuttal Retrieval model, and find the best performing model one (for the metrics of the test set) for both steps. Then, we compare it to the results obtained in RQ1.

4.2 Datasets

Like any **ML** problem, in particular, a Supervised Classification problem, a very important factor for the results obtained is the dataset used. Here we introduce the datasets that we used for the different parts of the project.

4.2.1 Statement Documents (tweets)

Datasets from several papers were analyzed for this step. We focused on labelled datasets concerning misinformation relating to COVID vaccination. In the end, we choose CoVaxLiesV2 [81] since it not only followed the above requisites, but also was quite extensive (over 6000 tweets). It had also already divided the tweets through 48 misinformation topics (a tweet can be related to more than one topic), thus providing categories that can be used also in the second sub-task (the contradiction detection). The most common Bi-Grams of the dataset, obtained using the nltk [8] library, can be seen in Figure 4.1, which was drawn using matplotlib's example [41]. In it, we can see that "covid 19" is the most popular Bi-Gram, and that some misinformation topics also form Bi-Grams ("coronavirus bioweapon", "man made", "bill gates").

This dataset only contained the tweets IDs, not their text, due to Twitter privacy policies. After their retrieval, the number of tweets was 6661, not 9322. This is due to some tweets being deleted in the meantime or the account that made them being set as private or being suspended. We also had to do some cleaning pre-processing, such as replacing all links with http, since they don't have relevance for our work, and extracting the words from hashtags so the individual words could be obtained. All usernames were replaced with @user due to Twitter's Privacy Policy.

4.2.2 Reference Corpus

In order to be able to search for contradictions, a reference corpus to compare them also had to be created. In order to create it, FAQ and Q&A answers relating to covid vaccination from several trusted medical sites (such as the CDC) were collected in a list to make their search easier. From those, the ones that best covered each of the 48 misinformation topics introduced in the CoVaxLiesV2 [81] dataset were taken. Other websites, such as Reuters, were used for the misinformation topics that could not be found on that list. The complete list of websites used and their distribution in the dataset can be seen in Table 4.1. Extra text was then removed, such as links to more information. This dataset is publicly available at [6].

4.2.3 Annotations Datasets

These datasets combine the first two, creating pairs between the Statement Documents (the tweets) and the Reference Documents. This is done as preparation of the training datasets for use in the two variants of the second component of the method. It was manually annotated based on the existing relations between tweets and misinformation topics from CoVaxLiesV2 [81], which only created a pair if the Statement Document and the Reference Document are related. This is done

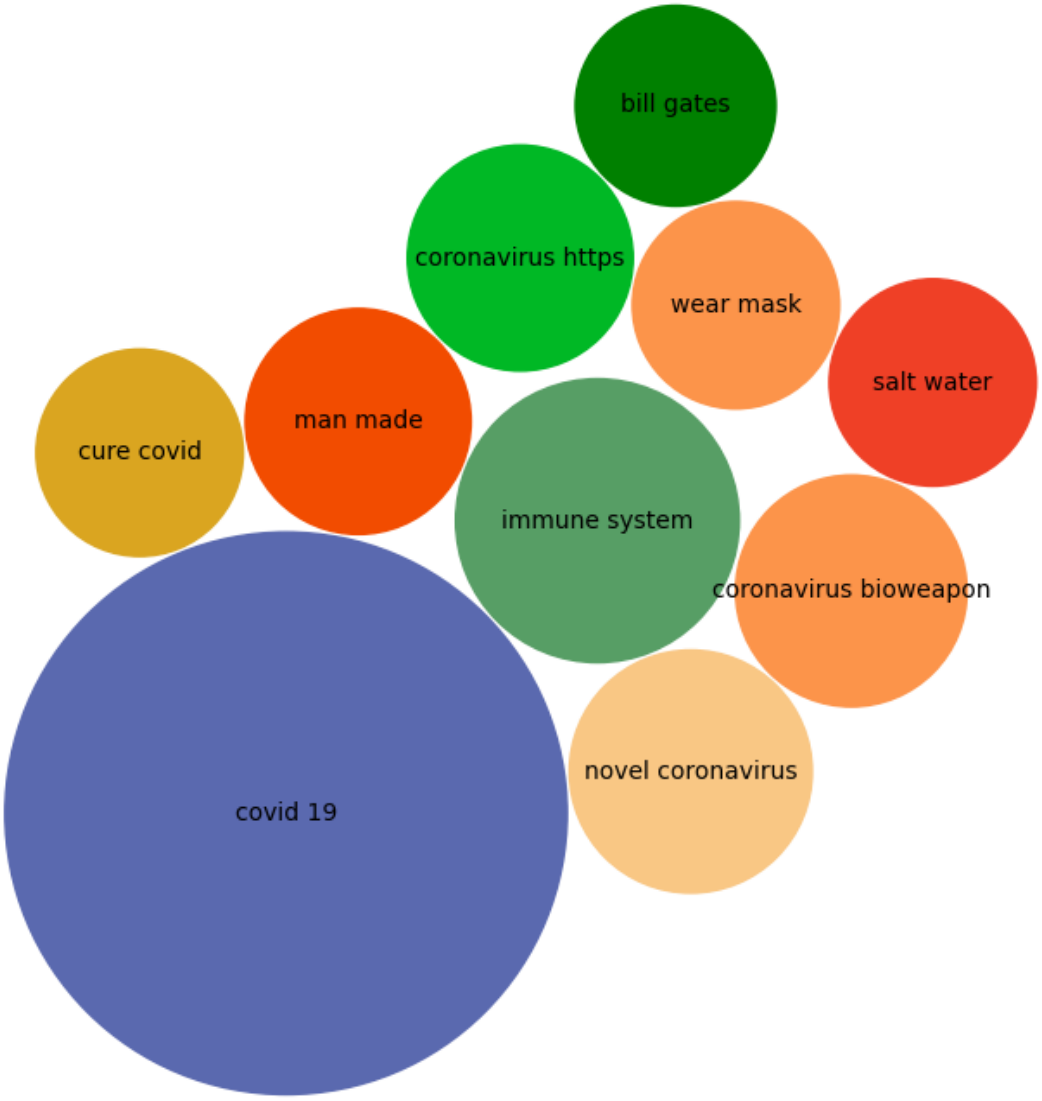


Figure 4.1: Packed Bubble Chart of the most common Bi-Grams.

in order to avoid Misinfobot's [49] problem of extreme class imbalance, which had most pairs marked as neutral.

For the first variant, an Annotations Dataset was created containing pairs between Statement Documents (the tweets) and Reference Documents from the Reference Corpus, named Annotations of Contradictions. If a tweet in CoVaxLiesV2 was marked as accepting a misinformation topic, the pairs of that tweet and the Reference Documents that contradict that misinformation topic were marked as a contradiction. Likewise, if a tweet in CoVaxLiesV2 was marked as rejecting a misinformation topic, the pairs of that tweet and the Reference Documents that contradict that misinformation topic were marked as an entailment. For all other pairs of CoVaxLiesV2, their tweets were marked as neutral with all Reference Documents that contradict the misinformation topics of those pairs.

Website	Times used
Mayo Clinic Health System [70]	7
CDC [14] [15] [16] [17] [18] [19]	11
John Hopkins [34]	5
MU Health Care [12]	5
Reuters [63] [64] [65] [66]	4
Yale New Haven Health [33]	3
Politifact [54] [55]	2
Poynter [57] [58]	2
WHO [84]	1
European Centre for Disease Prevention and Control [28]	1
New Jersey COVID-19 Information Hub [35]	1
Britannica [11]	1
Wikipedia [85]	1
Snopes [69]	1
UNESCO [75]	1
National Center for Complementary and Integrative Health [27]	1
Nebraska Medicine [42]	1

Table 4.1: Reference Documents Sources

For the second variant, a second Annotations dataset that contains pairs between Statement Documents (the tweets) and a list of the misinformation topics covered in the Reference Corpus was created, named Annotations of Entailments. If a tweet in CoVaxLiesV2 was marked as accepting a misinformation topic, the pair of that tweet and that misinformation topic was marked as entailment. Likewise, if a tweet in CoVaxLiesV2 was marked as rejecting a misinformation topic, the pair of that tweet and that misinformation topic was marked as a contradiction. For all other pairs of CoVaxLiesV2, their tweets were marked as neutral with all misinformation topics of those pairs. We can see in Figure 4.2 that although most tweets only have one topic of misinformation associated with them, several of them have more, up to 17 (due to several of them actually being unrelated to the tweet). As the second component of the method models the relation between fake tweets and reference documents, the pairing for training purposes only considered posts labeled as contradiction or neutral. Hence, all tweets that should be marked as true by the first step have been removed from the dataset. In Figure 4.3 we show the new distribution of misinformations per Statement Document after that removal. We can see that the dataset is unbalanced regarding the topics covered, with some misinformation topics having five times as many tweets as others.

4.3 Experimental Setup

4.3.1 Language Models used

One very important component of a classification task in NLP is the language model used. Since they usually already come with pre-training, it is essential to choose some whose pre-training is relevant and related to the target domain. Huggingface [87] is a very good resource since it

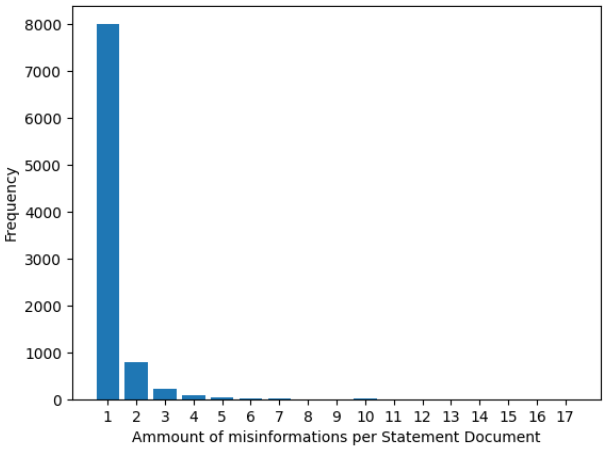


Figure 4.2: Distribution of misinformation topics per Statement Document

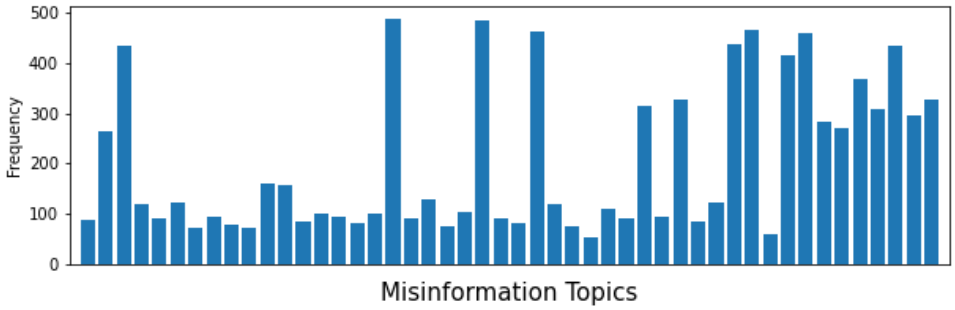


Figure 4.3: Distribution of Statement Documents per misinformation topic

contains several pre-trained models. Of their models, we prioritized those related to COVID, with an emphasis on those trained on Twitter. For the second step, we prioritized models trained in contradiction detection, thus we looked for models using know datasets for that task, such as SNLI [10]. We ended up using several models for the same task because we wanted to be sure that we obtained the best performance, and it is hard to know in advance which model will perform the best since it depends on which task we are tackling. The models chosen can be seen below, where we also named them. The ones named DM refer to the ones used for the first step (Detecting Misinformation, Section 3.2.1) and RR refers to the ones used for the second step (Rebuttal Retrieval, Section 3.2.2). All models were then fine-tuned with the respective dataset. The models chosen can be seen on Table 4.2 and on Table 4.3.

4.3.2 Hyperparameters and Performance Estimation

Another very important decision when training a model for a classification task is the hyperparameters chosen since they can make the difference between the model working or not. So we

Model	Name	Description
ans/vaccinating-covid-tweets [2]	DM1	Based on BERTweet and then pre-trained on COVID-19 vaccinating tweets. Also fine-tuned on a fact-classification task for COVID vaccine related statements
digitalepidemiologylab/covid-twitter-bert-v2 [44]	DM2	Based on BERT-large-uncased and then pre-trained on a large corpus of COVID-19 tweets
vinai/bertweet-covid19-base-cased [46]	DM3	Uses a training procedure based on RoBERTa, it's a model pretrained on a very large corpus of 850M tweets, of which 5M are relating to COVID-19
vinai/bertweet-covid19-base-uncased [46]	DM4	The same as DM3, but using an uncased model. Will help us see if using cased or uncased models will give us better results.

Table 4.2: Language Models chosen for the Misinformation Detection sub-task

Model	Name	Description
cross-encoder/nli-roberta-base [62]	RR1	A Sentence Transformers Cross-Encoder model trained on the SNLI [10] and MultiNLI [86] datasets, two very large datasets for language that have the contradiction, entailment and neutral labels
MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli [38]	RR2	A DeBERTa-v3-base [31] model trained to detect textual entailment, neutrality or contradiction. This model was trained on the MultiNLI [86], Fever-NLI [71] and Adversarial-NLI (ANLI) [47] datasets
ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli [47]	RR3	A RoBERTa-Large model pretrained in the SNLI [10], MNLI [86], FEVER-NLI [71], ANLI (R1, R2, R3) datasets. This model is already trained to return the contradiction, entailment or neutral label

Table 4.3: Language Models chosen for the Rebuttal Retrieval sub-task

used Bayesian Optimization from the KerasTurner [51] library in order to find the optimal learning rate, warmup steps and epochs since it has been shown to get better results with fewer tries in comparison to grid and random search [72]. It consists of trying to gather information about the function optimum so it can reason about experiments before they are run, and we can see the values obtained in Table 4.4. For the maximum sequence length, we used 128 for the first step and 256 for the second step, and for the batch size, we used 16 for both the training and evaluation,

since it was the maximum we could use with our hardware (except for model RR3, were we had to reduce the training batch size to 8). We used AdamW [40] as the learning rate optimizer and evaluated the model at the end of each epoch. In order to implement the WarmUp steps, we used [88].

	Epochs	Learning Rate	WarmUp Steps	Maximum Sequence Length
DM1	4	5e-5	500	128
DM2	4	5e-5	400	128
DM3	3	5e-5	550	128
DM4	3	5e-5	550	128
RR1	4	1e-5	400	256
RR2	3	5e-5	1200	256
RR3	4	1e-6	1700	256

Table 4.4: HyperParameters used

The dataset is distributed between the train (60%), evaluate (20%) and test (20%) set, as seen in Table 4.5 for the Detecting Misinformation sub-task. We can observe that there is a class imbalance since 67% of the tweets are marked as true. This is partly because, when retrieving the tweets with the Twitter API, several fake ones being either deleted, made from an account that was set as private or from a suspended account.

	True	Fake	Total
Train	2727 (68%)	1269 (32%)	3996
Evaluate	882 (66%)	450 (34%)	1332
Test	875 (66%)	458 (34%)	1333
Total	4484 (67%)	2177 (33%)	6661

Table 4.5: Distribution of the dataset for the Detecting Misinformation sub-task

For the Retrieving Rebuttal Information sub-task, the dataset is distributed as seen in Table 4.6. Like previously stated in Section 4.2.3, the pairs are only created if the reference document and the tweet are about similar topics, and all tweets that should be marked as true by the first step have been removed from the dataset. This is the reason for the extremely low number of the Entails class (it is not 0 because some tweets contain both truth and lies, making them contradict some reference documents while also entailing others).

	Contradicts	Neutral	Entails	Total
Train	1403 (35%)	2525 (64%)	37 (2%)	3961
Evaluate	472 (36%)	836 (63%)	12 (1%)	1320
Test	494 (86%)	812 (13%)	15 (1%)	1321
Total	2369 (36%)	4173 (63%)	60 (1%)	6602

Table 4.6: Distribution of the dataset for the Retrieving Rebuttal sub-task

4.3.3 Frameworks used

In order to implement the Twitter Bot, we took advantage of the Tweepy [67] python library, which made interacting with the Twitter API much easier.

Due to the heavy requirements of the BERT and RoBERTa models, we ended up using Google Colaboratory [9] to run the experiments since it has free access to an almost 16 Gb GPU for runtimes of up to 12 hours.

We also used the HuggingFace Transformers [87] library in conjunction with TensorFlow2 to develop the model. Transformers is an extremely popular NLP library right now that has support for both PyTorch and TensorFlow. It also allows access to an extensive catalog of pre-trained models, which allowed us to save time for the pre-training phase.

4.4 Summary

In this chapter, we explored the experimental setup used.

We describe our two main research questions. **RQ1** addresses the new method proposed by combining the previous two methods, and **RQ2** addresses to use an alternative approach to **RQ1**.

We mentioned the datasets that are going to be used, how they were obtained and how they were pre-processed. We also presented their distribution and how they will be used.

We also discussed the experimental setup we will use, such as the models, hyperparameters and frameworks used.

Chapter 5

Results and Analysis

In this chapter, we analyze the results obtained from the experiments. To evaluate these results, we use the metrics described in Section 2.1.3, since this problem is a Classification Problem. As this problem has two steps, we first present the individual results of each step in Section 5.1, and then we present the results of the Research Questions in Section 5.2.

5.1 Results

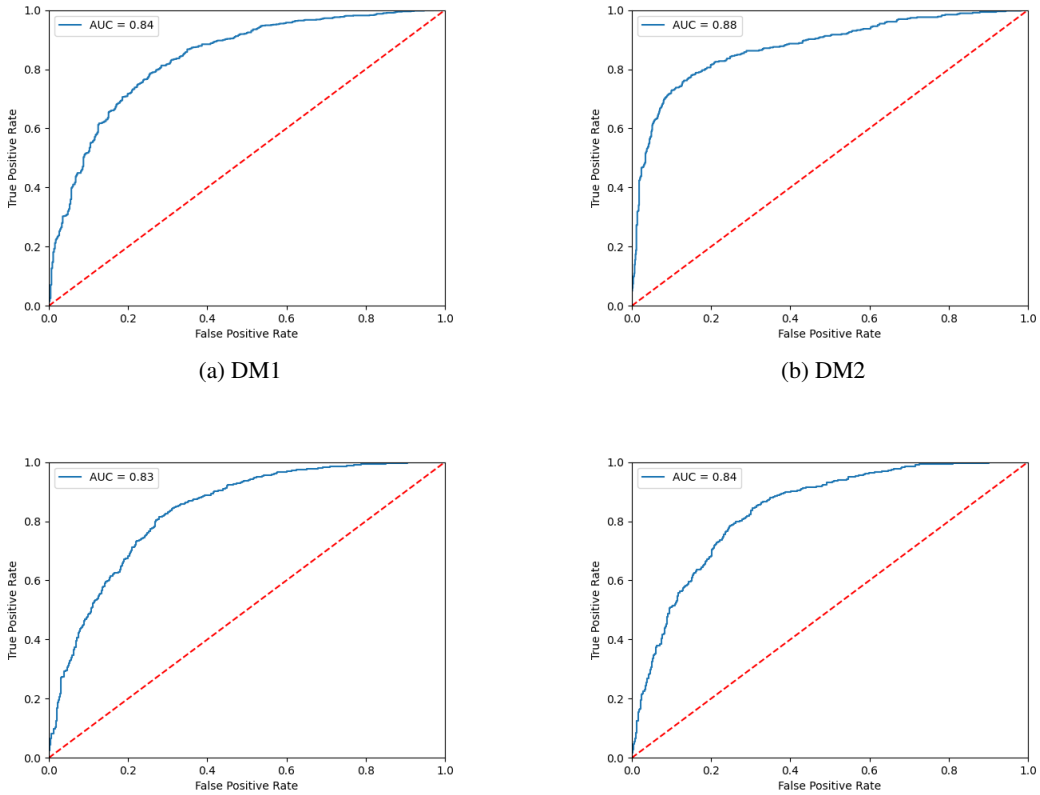
In this Section we present the individual results of each sub-task in order to be able to analyze each result better and it's meaning. The confusion matrices obtained as a result of the experiments are in Annex A.

5.1.1 Detecting Misinformation sub-task

We fine-tuned the pre-trained models on the Statement Documents dataset. Table 5.1 presents their performance metrics on the test set. We also calculated the ROC curve and AUC, which can be seen in Figure 5.1. Overall, all models had good results (over 80% AUC, with the ROC curve being well above the line representing random predictions). This is partly due to all of them being trained on COVID-19 datasets. However, we can see that the model DM2 (covid- twitter-bert-v2) has the best F1 score and AUC, which can be attributed to being pre-trained on the largest corpus of the same target domain (tweets related to COVID-19 vaccination). This allows the model to already be very familiar with the vocabulary used, thus boosting its performance.

Model	Accuracy	Precision	Recall	F1	AUC
DM1	78.54%	73.62%	58.51%	65.20%	83.73%
DM2	83.87%	79.13%	72.05%	75.42%	87.59%
DM3	78.99%	75.28%	57.86%	65.43%	83.40%
DM4	79.21%	76.38%	57.20%	65.41%	83.64%

Table 5.1: Detecting Misinformation performance metrics



(a) DM1 (b) DM2

(c) DM3 (d) DM4

Figure 5.1: ROC Curves of Detecting Misinformation sub-task

5.1.2 Retrieving Rebuttal Information sub-task

5.1.2.1 Contradictions to Reference Documents approach

We fine-tuned the pre-trained models on the Annotations of Contradictions dataset. We considered this a multi-class problem when calculating the performance metrics, and their values can be seen in Table 5.2. We can also see their ROC curves in Figure 5.2. Micro-average was used for the precision, recall and F1 score. We can see that all three models have very good and similar results since their AUC is over 87%. This can also be seen on their ROC curves, where, once again, the curve is well above the line representing random predictions. This is due to having done much pre-training in several NLI datasets. However, RR2 (DeBERTa-v3-base-mnli-fever-anli) has the best results due to its model, DeBERTa, which has been proven to outperform Roberta-base on MNLI tasks [32].

5.1.2.2 Entailment to Misinformation Topics approach

As an alternative approach to Section 5.1.2.1, we fine-tuned the pre-trained models on the Annotations of Entailments dataset. We can see their performance metrics and the difference between

Model	Accuracy	Precision	Recall	F1	AUC
RR1	78.95%	79.13%	62.95%	70.12%	87.70%
RR2	80.69%	80.88%	66.80%	73.17%	89.83%
RR3	80.77%	78.89%	69.63%	73.97%	88.70%

Table 5.2: Contradictions to Reference Documents sub-task performance metrics

their performance metrics and the ones from Section 5.1.2.1 in Table 5.3. Their ROC curves can be seen in Figure 5.3. Overall, their metric results and ROC Curves are better than the ones from Section 5.1.2.1. This shows that the proposed approach of using entailment to misinformation topics instead of contradiction to Reference Documents can give better results due to the tweets' text being more similar in form and content to the misinformation topics.

Model	Accuracy	Precision	Recall	F1	AUC
RR1	82.28%	80.91%	71.25%	75.78%	90.42%
RR2	82.05%	88.43%	61.94%	74.85%	92.13%
RR3	83.42%	82.27%	73.28%	77.51%	89.66%
RR1_diff	3.33%	1.78%	8.30%	5.66%	2.72%
RR2_diff	1.36%	7.55%	-4.86%	1.68%	2.30%
RR3_diff	2.65%	3.38%	3.65%	3.54%	0.96%

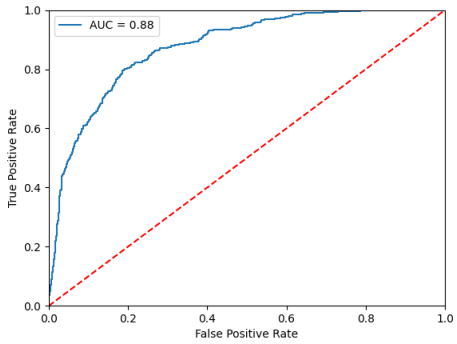
Table 5.3: Entailment to Misinformation Topics sub-task performance metrics

5.1.3 Complete method using Contradictions to Reference Documents

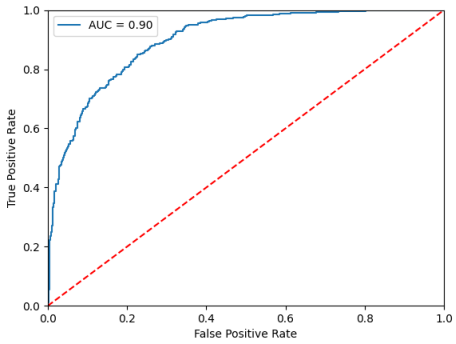
We take the models fine-tuned in Section 5.1.1 and Section 5.1.2.1, choosing the best performing among them for each step. In this case, it was model DM2 for the Detecting Misinformation sub-task. Since the results for the Contradictions to Reference Documents approach were very similar, we tested the complete method using all possible models for that step. We combined the predictions of both sub-tasks and use multi-label metrics to evaluate the results, comparing them with the ground-truth of the original labels. The performance metrics can be seen in Table 5.4. We can see that DM2RR2 has the best results for this approach.

Model	Accuracy	Precision	Recall	F1	Hamming Loss
DM2RR1	61.19%	60.71%	62.04%	61.37%	1.62%
DM2RR2	62.41%	61.67%	63.01%	62.33%	1.58%
DM2RR3	61.36%	60.13%	61.44%	60.77%	1.65%

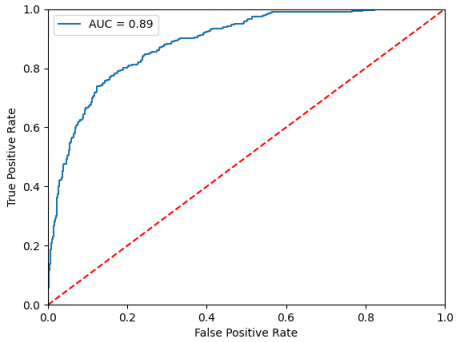
Table 5.4: Complete method with Contradictions to Reference Documents performance metrics



(a) RR1



(b) RR2



(c) RR3

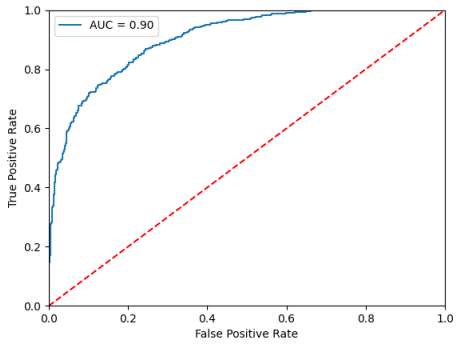
Figure 5.2: ROC Curves of Retrieving Rebuttal using contradiction sub-task

5.2 Research Questions

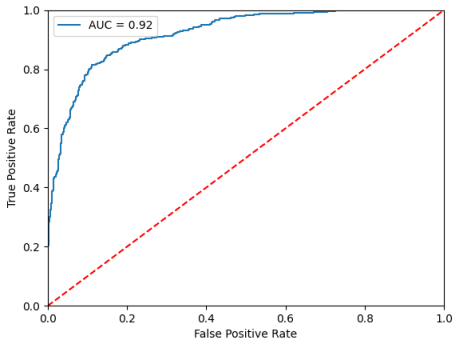
5.2.1 RQ1 - Can the Contradictions to Reference Documents approach present a better result than the original approaches developed by Misinfobot?

In order to answer to this question, we compare need to compare the results of Section 5.1.3 with the same combination of models, tested using in Misinfobot's [49] dataset. We use the DM2RR2 combination of the previous models since that combination had the best results. We pre-trained DM2 since Misinfobot's dataset touched on new misinformation topics about COVID. However, we did not pre-train RR2 in Misinfobot's dataset since that dataset too unbalanced for effective contradiction detection fine-tunning. The Annotations of Contradictions dataset should also work since the model only needs to know how to detect contradictions (there isn't a high need for the dataset to be topic specific, only task specific).

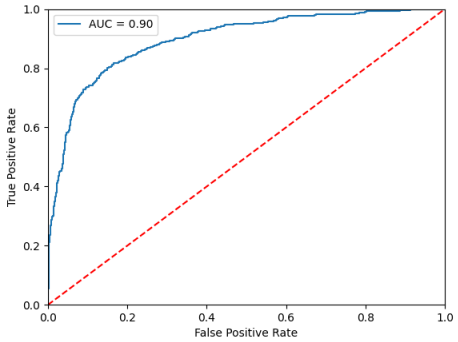
In Table 5.5 we can see the results for this method. We also show the results if, in the second step, we output all Reference Documents detected as contradictions instead of only outputting the one with the highest probability of being a contradiction. This difference in results shows us that the model can accurately predict misinformations and that the lower values of the metrics are only due to the Statement Document having many Reference Documents associated with it. Since we



(a) RR1



(b) RR2



(c) RR3

Figure 5.3: ROC Curves of Retrieving Rebuttal using entailment sub-task

Model	Accuracy	Precision	Recall	F1	Hamming Loss
DM2RR2	41.88%	18.04%	47.22%	26.10%	0.11%
DM2RR2 without max	67.94%	16.89%	83.80%	28.12%	0.09%

Table 5.5: Complete method with Contradictions to Reference Documents using Misinfobot's dataset performance metrics

are only taking the Reference Document with the highest probability of being a contradiction, the more Reference Documents associated with the Statement Document there are, the more significant the difference between the prediction of the model and the ground-truth, even if the model is perfect and detects all contradictions. Therefore, the increase in metrics shows that the model can accurately predict contradictions, but we are ignoring them due to only taking one Reference Document.

Comparing the results with the ones obtained by Misinfobot [49], we can see that it has obtained a lower accuracy and precision. However, it has obtained a much higher recall (sensitivity) and F1-score. This means our model obtains more false positives (precision) but fewer false negatives (recall). Since the goal of the model is to output only one Reference Document, the increase in false negatives does not affect us since we will only be choosing the one with the highest

probability of contradicting the misinformation. The lower accuracy is expected since the reason Misinfobot got such high accuracy is due to the extreme class imbalance in the dataset.

Overall, we can conclude that we managed to get better results than Misinfobot. This is possibly due to several factors, such as a more extensive dataset for Statement Documents(tweets) and also the avoidance of class imbalance for the dataset of the Rebuttal Retrieval step by only creating relevant pairs.

5.2.2 RQ2 - Can we produce better results by using the Entailments to Misinformations approach?

In order to find the answer to this question, we take the models fine-tuned in Section 5.1.1 and Section 5.1.2.2, choosing the best performing among them for each step. In this case, it was model DM2 for the Detecting Misinformation sub-task. Since the results for the Entailment to Misinformation Topics approach were also very similar, we tested the complete method using all possible models for that step. We combined the predictions of both sub-tasks and use multi-label metrics to evaluate the results, comparing them with the ground-truth of the original labels.

We then calculated the new metrics for them, which can be seen in Table 5.6, and compare them to the results obtained in Section 5.1.3. We can see that DM2RR1 has the best results overall, which shows that the best language model for contradiction detection isn't necessarily the best model for finding entailments. The higher metrics in relation to the complete method with Contradictions to Reference Documents can be attributed to the fact that the tweets are more similar in form and content to the misinformation topics compared to the Reference Documents. This shows that using entailment to misinformation topics can help avoid Misinfobot's [49] problem of the difference between text characteristics of the Statement Documents (tweets) and Reference Documents.

Model	Accuracy	Precision	Recall	F1	Hamming Loss
DM2RR1	75.31%	74.88%	76.51%	75.69%	1.02%
DM2RR2	75.01%	74.74%	76.36%	75.54%	1.02%
DM2RR3	72.99%	72.46%	74.04%	73.24%	1.12%

Table 5.6: Complete method using entailment to misinformations performance metrics

5.3 Summary

We fine-tuned all models and calculated their performance metrics, which were overall good.

The best performing model for Detecting Misinformation was DM2 (covid- twitter-bert-v2), and the best complete model for the Contradictions to Reference Documents approach was DM2RR2.

We managed to prove that this approach can get better results than the original Misinfobot's approach, and that the Entailments to Misinformations approach can get even better results, since the difference between text characteristics is smaller.

Chapter 6

Conclusions

With the growth in Social Media, we have seen a growth in the spread of misinformation. This misinformation has the potential to be dangerous since some people use it to make decisions, and solutions for it are still being developed. To do that, we created a bot that identifies misinformation on Twitter and rebuts it with the related reference information.

Misinfobot [49] created two approaches, one which separated the task into two steps (Classify and Retrieve) and one which used Contradiction Detection. We created an approach that uses Supervised Learning Classification and combines the two approaches developed by Misinfobot. We also separate the problem into two steps. The first step uses a Misinformation Detection model (used in Misinfobot's first approach) to mark the received statement document as true or fake. The second step receives all items labeled as misinformation (as fake) by the first step to a Contradiction Detection model (used in Misinfobot's second approach). We used two approaches for this step. In one, this model compares each tweet with each Reference Document and outputs the probability of contradiction, entailment, or neutrality between them, choosing the one with the highest probability of contradiction. In the other approach, we try to find entailment between the tweets and misinformation topics. Both steps use pre-trained models, which were fine-tuned on the datasets used to test the models. After evaluating them empirically on the new datasets created and on Misinfobot's datasets, we can conclude that both approaches have a high potential in fighting misinformation. The first approach (using contradiction to Reference Documents) can outperform Misinfobot since it avoids the class imbalance in the contradiction detection model. The second approach (using entailment to misinformation topics) can even outperform the first approach due to avoiding the difference in text characteristics between the Statement and Reference Documents.

Overall, we can conclude that NLP can be effective to detect and rebut misinformation, and that both the classification of misinformation and contradiction detection can be viable approaches for that task. However, the results also show that there should be margin for growth with other approaches.

6.1 Contributions

This dissertation has the following contributions:

- The creation of a Twitter Bot that uses the method developed to identify misinformations and rebut them with the related reference information
- A new innovative approach to misinformation rebuttal by using entailment to misinformation topics
- The creation of an annotated dataset with the semantic relationship between reference documents and tweets pertaining to COVID-19 vaccination
- Empirical Evaluation of the method proposed on the COVID-19 vaccination context

6.2 Future Work

Despite the positive results obtained, there is still margin for improvement. Other methods could be used that could yield different results. In regards to our particular approach, the following could be explored in order to improve it:

- This approach should be tested on misinformation topics other than COVID-19 to truly see its effectiveness
- Some misinformations are reliant on time (being true or false depends on the date that we are discussing it), and need to be addressed before this can be a fully viable method;
- Different language models could possibly yield better results. Likewise, maybe using Population-based training and a higher interval of possible hyperparameters could lead to better results;
- Some bias could have been introduced in the pairs dataset since they were manually annotated, or there could already exist bias in the original dataset that they were based on. Increasing the dataset size and reducing the bias could produce better results;
- Adding an intermediary step between the two steps (Misinformation Detection and Rebuttal Retrieval) could produce better results. One possibility for this step is a model that categorizes the statement documents by topic before trying to find contradictions;
- This approach requires the manual creation of datasets, which can be time-consuming. Self-supervised approaches could be incorporated to account for that.

Appendix A

Confusion Matrices

A.1 Confusion Matrices For Detecting Misinformation

In this section we show the confusion matrices obtained as a result of the sub-task described in Section 3.2.1.

		Actual value		
		Fake	Real	Total
Predicted Value	Fake	268	96	364
	Real	190	779	969
	Total	458	875	1333

Table A.1: Confusion Matrix for DM1

		Actual value		
		Fake	Real	Total
Predicted Value	Fake	330	87	417
	Real	128	788	906
	Total	458	875	1333

Table A.2: Confusion Matrix for DM2

		Actual value		
		Fake	Real	Total
Predicted Value	Fake	265	87	352
	Real	193	788	981
	Total	458	875	1333

Table A.3: Confusion Matrix for DM3

		Actual value		
		Fake	Real	Total
Predicted Value	Fake	262	81	343
	Real	196	794	990
	Total	458	875	1333

Table A.4: Confusion Matrix for DM4

A.2 Confusion Matrices For Retrieving Rebuttal using Contradictions

In this section we show the confusion matrices obtained as a result of the sub-task described in Section 5.1.2.1.

		Actual value			
		Entailment	Neutral	Contradiction	Total
Predicted Value	Entailment	0	1	0	1
	Neutral	12	732	183	927
	Contradiction	3	79	311	393
	Total	15	812	494	1321

Table A.5: Confusion Matrix for RR1 using Contradictions

		Actual value			
		Entailment	Neutral	Contradiction	Total
Predicted Value	Entailment	0	0	0	0
	Neutral	13	736	164	913
	Contradiction	2	76	330	408
	Total	15	812	494	1321

Table A.6: Confusion Matrix for RR2 using Contradictions

A.3 Confusion Matrices For Retrieving Rebuttal using Entailment

In this section we show the confusion matrices obtained as a result of the sub-task described in Section 5.1.2.2.

		Actual value			
		Entailment	Neutral	Contradiction	Total
Predicted Value	Entailment	5	4	5	14
	Neutral	8	718	145	871
	Contradiction	2	90	344	436
	Total	15	812	494	1321

Table A.7: Confusion Matrix for RR3 using Contradictions

		Actual value			
		Entailment	Neutral	Contradiction	Total
Predicted Value	Entailment	352	78	5	435
	Neutral	140	732	7	879
	Contradiction	2	2	3	7
	Total	494	812	15	1321

Table A.8: Confusion Matrix for RR1 using Entailments

		Actual value			
		Entailment	Neutral	Contradiction	Total
Predicted Value	Entailment	306	37	3	346
	Neutral	186	774	8	968
	Contradiction	2	1	4	7
	Total	494	812	15	1321

Table A.9: Confusion Matrix for RR2 using Entailments

		Actual value			
		Entailment	Neutral	Contradiction	Total
Predicted Value	Entailment	362	74	4	440
	Neutral	128	737	8	873
	Contradiction	4	1	3	8
	Total	494	812	15	1321

Table A.10: Confusion Matrix for RR3 using Entailments

References

[1] Diaa Salama Abdelminaam, Fatma Helmy Ismail, Mohamed Taha, Ahmed Taha, Essam H. Houssein, and Ayman Nabil. Coaid-deep: An optimized intelligent framework for automated detecting covid-19 misleading information on twitter. *IEEE Access*, 9:27840–27867, 2021.

[2] Hyunju Ahn, Jiyong An, Seungchan An, Seokho Jeong, Jungmin Kim, and Sangbeom Kim. Huggingface model ans/vaccinating-covid-tweets . https://huggingface.co/ans/vaccinating-covid-tweets, 2021.

[3] Noralhuda N. Alabid and Zainab Dalaf Katheeth. Sentiment analysis of twitter posts related to the covid-19 vaccines. *Indonesian Journal of Electrical Engineering and Computer Science*, 24:1727, 12 2021.

[4] Md Zahangir Alom, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen, Abdul A. S. Awwal, and Vijayan K. Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3), 2019.

[5] Miguel A. Alonso, David Vilares, Carlos Gómez-Rodríguez, and Jesús Vilares. Sentiment analysis for fake news detection. *Electronics*, 10:1348, 6 2021.

[6] João Araújo. https://gitlab.com/jfcaraujo/misinfobot2-dataset.

[7] Yejin Bang, Etsuko Ishii, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. *Model Generalization on COVID-19 Fake News Detection*, volume 1402 CCIS, pages 128–140. Springer, Cham, 2 2021.

[8] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[9] Ekaba Bisong. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA, 2019.

[10] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[11] Britanica. https://www.britannica.com/list/the-top-covid-19-vaccine-myths-spreading-online.

[12] MU Health Care. https://www.muhealth.org/our-stories/covid-19-vaccine-myths-vs-facts.

[13] CDC. https://www.cdc.gov/media/releases/2020/p0714-americans-to-wear-masks.html/.

[14] CDC. https://www.cdc.gov/coronavirus/2019-ncov/vaccines/facts.html.

[15] CDC. https://www.cdc.gov/coronavirus/2019-ncov/vaccines/distributing/about-vaccine-data.html.

[16] CDC. https://www.cdc.gov/coronavirus/2019-ncov/vaccines/faq-children.html.

[17] CDC. https://www.cdc.gov/vaccinesafety/concerns/autism.html.

[18] CDC. https://emergency.cdc.gov/newsletters/coca/020122.html.

[19] CDC. https://www.cdc.gov/coronavirus/2019-ncov/your-health/reinfection.html.

[20] Ben Chen, Bin Chen, Dehong Gao, Qijin Chen, Chengfu Huo, Xiaonan Meng, Weijun Ren, and Yang Zhou. *Transformer-Based Language Model Fine-Tuning Methods for COVID-19 Fake News Detection*, volume 1402 CCIS, pages 83–92. Springer Science and Business Media Deutschland GmbH, 2021.

[21] Sourya Dipta Das, Ayan Basak, and Saikat Dutta. *A Heuristic-Driven Ensemble Framework for COVID-19 Fake News Detection*, volume 1402 CCIS, pages 164–176. Springer, Cham, 2 2021. heuristic post processing pode ser interessante.

[22] Sourya Dipta Das, Ayan Basak, and Saikat Dutta. A heuristic-driven ensemble framework for covid-19 fake news detection. *Communications in Computer and Information Science*, 1402 CCIS:164–176, 2 2021.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[24] Cambridge Dictionary. https://dictionary.cambridge.org/dictionary/english/misinformation.

[25] Mohamed K. Elhadad, Kin Fun Li, and Fayez Gebali. *An Ensemble Deep Learning Technique to Detect COVID-19 Misleading Information*, volume 1264 AISC, pages 163–175. Springer, 2021.

[26] FDA. https://www.fda.gov/drugs/drug-safety-and-availability/fda-cautions-against-use-hydroxychloroquine-or-chloroquine-covid-19-outside-hospital-setting-or/.

[27] National Center for Complementary and Integrative Health. https://www.nccih.nih.gov/health/covid-19-and-alternative-treatments-what-you-need-to-know.

[28] European Centre for Disease Prevention and Control. https://www.ecdc.europa.eu/en/covid-19/questions-answers/questions-and-answers-vaccines.

[29] Tedros Adhanom Ghebreyesus. https://twitter.com/DrTedros/status/1486723273362575361, January 2022.

[30] Google. https://code.google.com/archive/p/word2vec/.

[31] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

[32] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2020.

[33] Yale New Haven Health. https://www.ynhhs.org/patient-care/covid-19/Vaccine/Vaccine-myths.

[34] John Hopkins. https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/covid-19-vaccines-myth-versus-fact.

[35] New Jersey COVID-19 Information Hub. https://covid19.nj.gov/faqs/coronavirus-information/about-the-virus/is-covid-19-a-biological-weapon.

[36] Rina Kumari, Nischal Ashok, Tirthankar Ghosal, and Asif Ekbal. Misinformation detection using multitask learning with mutual learning for novelty detection and emotion recognition. *Information Processing & Management*, 58:102631, 9 2021.

[37] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[38] Moritz Laurer. Huggingface model MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli . https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli, 2021.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[40] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[41] Matplotlib. https://matplotlib.org/3.5.0/gallery/misc/packed_bubbles.html?highlight=bubble%20chart#.

[42] Nebraska Medicine. https://www.nebraskamed.com/COVID/antibody-dependent-enhancement-in-vaccines.

[43] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), apr 2021.

[44] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*, 2020.

[45] Usman Naseem, Imran Razzak, Matloob Khushi, Peter W. Eklund, and Jinman Kim. Covidsenti: A large-scale benchmark twitter data set for covid-19 sentiment analysis. *IEEE Transactions on Computational Social Systems*, 8:1003–1015, 8 2021.

[46] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020.

[47] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[48] Shuteng Niu, Yongxin Liu, Jian Wang, and Houbing Song. A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2):151–166, 2020.

[49] Tomás Nuno Fernandes Novo. MisInfoBot: fight misinformation about COVID on social media. Master's thesis, FEUP, 2021.

[50] Fahmi Nurfikri. https://towardsdatascience.com/an-illustrated-guide-to-artificial-neural-networks-f149a549ba74.

[51] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. https://github.com/keras-team/keras-tuner, 2019.

[52] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2021.

[53] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 10 2010.

[54] Politifact. https://www.politifact.com/factchecks/2021/oct/21/carrie-madej/transhumanism-nanotechnology-covid-19-vaccine-cons/.

[55] Politifact. https://www.politifact.com/factchecks/2020/dec/22/tweets/viral-tweet-cites-made-cdc-covid-19-survival-rates/.

[56] Politwoops. https://projects.propublica.org/politwoops/tweet/1287955156387201024.

[57] Poynter. https://www.poynter.org/?ifcn_misinformation=covid-19-vaccines-will-make-men-gay.

[58] Poynter. https://www.poynter.org/?ifcn_misinformation=children-are-already-protected-against-the-covid-19-because-of-their-natura

[59] Khubaib Ahmed Qureshi, Rauf Ahmed Shams Malick, Muhammad Sabih, and Hocine Cherifi. Complex network and source inspired covid-19 fake news classification on twitter. *IEEE Access*, 9:139636–139656, 2021.

[60] Khubaib Ahmed Qureshi, Rauf Ahmed Shams Malick, Muhammad Sabih, and Hocine Cherifi. Complex network and source inspired covid-19 fake news classification on twitter. *IEEE Access*, 9:139636–139656, 2021.

[61] The Click Reader. https://www.theclickreader.com/introduction-to-convolutional-neural-networks/.

[62] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[63] Reuters. https://www.reuters.com/article/factcheck-vaccine-cytotoxic-idUSL2N2O01XP.

[64] Reuters. https://www.reuters.com/article/factcheck-vaccines-mrna-idUSL1N2L9187.

[65] Reuters. https://www.reuters.com/article/factcheck-coronavirus-britain-idUSL1N2SY1SP.

[66] Reuters. https://www.reuters.com/article/factcheck-covid-mrna-gene-idUSL1N2PH16N.

[67] Joshua Roesslein. Tweepy: Twitter for python! *URL: https://github.com/tweepy/tweepy*, 2020.

[68] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how BERT works. *CoRR*, abs/2002.12327, 2020.

[69] Snopes. https://www.snopes.com/fact-check/bill-gates-covid-vaccines/.

[70] Mayo Clinic Health System. https://www.mayoclinichealthsystem.org/hometown-health/featured-topic/covid-19-vaccine-myths-debunked.

[71] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.

[72] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 847–855, New York, NY, USA, 2013. Association for Computing Machinery.

[73] Aysenur Topbas, Akhtar Jamil, Alaa Ali Hameed, Syed Muzafar Ali, Sibghatullah Bazai, and Syed Attique Shah. Sentiment analysis for covid-19 tweets using recurrent neural network (rnn) and bidirectional encoder representations (bert) models. *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, pages 1–6, 10 2021.

[74] Twitter. https://s22.q4cdn.com/826641620/files/doc_financials/2021/q4/Final-Q4'21-Selected-Metrics-and-Financials.pdf.

[75] UNESCO. https://en.unesco.org/sites/default/files/unesco_social_media_posts_on_disinformation_during_covid19.pdf.

[76] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, and N.I. Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177:232–243, 11 2020.

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html.

[79] Apurva Wani, Isha Joshi, Snehal Khandve, Vedangi Wagh, and Raviraj Joshi. *Evaluating Deep Learning Approaches for Covid19 Fake News Detection*, volume 1402 CCIS, pages 153–163. Springer, Cham, 2 2021. unsupervised learning in the form of language model pre-training and distributed word representations using unlabelled covid tweets corpus.

[80] WebArchive. https://web.archive.org/web/20200728005026/https://twitter.com/stella_immanuel/status/1287

[81] Maxwell Weinzierl and Sanda Harabagiu. Identifying the adoption or rejection of misinformation targeting covid-19 vaccines in twitter discourse. In *Proceedings of the Web Conference 2022*, New York, NY, USA, 2022. Association for Computing Machinery.

[82] Maxwell A. Weinzierl and Sanda M. Harabagiu. Automatic detection of covid-19 vaccine misinformation with graph link prediction. *Journal of Biomedical Informatics*, 124:103955, 12 2021.

[83] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3:9, 12 2016.

[84] WHO. https://www.who.int/news-room/questions-and-answers/item/coronavirus-disease-(covid-19)-hydroxychloroquine.

[85] Wikipedia. https://en.wikipedia.org/wiki/COVID-19_vaccine_misinformation_and_hesitancy.

[86] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018.

[87] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[88] Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Pengchong Jin Le Hou, Fan Yang, Frederick Liu, Jaeyoun Kim, and Jing Li. TensorFlow Model Garden. https://github.com/tensorflow/models, 2020.

[89] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. Semantics-aware bert for language understanding, 2019.

[90] Anand Zutshi and Aman Raj. *Tackling the Infodemic: Analysis Using Transformer Based Models*, volume 1402 CCIS, pages 93–105. Springer, Cham, 2 2021.