FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

MisInfoBot: fight misinformation about COVID on social media

Tomás Nuno Fernandes Novo

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Carlos Manuel Milheiro de Oliveira Pinto Soares, PhD

Co-Supervisor: João António Fernandes da Costa, MEng

July 29, 2021

MisInfoBot: fight misinformation about COVID on social media

Tomás Nuno Fernandes Novo

Mestrado Integrado em Engenharia Informática e Computação

July 29, 2021

Resumo

A presença de misinformation nos Social Media é um fenómeno ameaçador cada vez mais comum que espalha informações incorretas para os seus muitos usuários. No atual contexto pandémico da COVID-19, os riscos deste tipo de informações e contradições são significativos devido ao seu potencial de causar perigosas reações emocionais, como sucedeu aquando da incerteza sobre a eficácia das máscaras protetoras e o seu uso. A evolução do Processamento de Linguagem Natural (NLP) e das suas técnicas associadas, como Deep Learning e Transfer Learning, possibilitam que computadores processem, interpretem e analisem linguagem humana escrita, concedendo a chance de lidar com a disseminação de informações incorretas.

Esta tese propõe uma nova abordagem baseada em Supervised Learning Classification e Information Retrieval que visa detectar automaticamente informações falsas sobre COVID-19 em tweets. A abordagem concebida foi implementada em dois métodos distintos. O primeiro método alia o fine-tuning de um deep-language model pré-treinado num corpus específico para classificação de tweets como verdadeiros ou falsos com o retorno de informações presentes num corpus de refêrencia. Todos os modelos usados são variações do deep language model BERT.

Com o fim de avaliar empiricamente a nossa abordagem, reunímos tweets com informações distintas sobre o vírus rotulados como verdadeiros ou falsos para criar um dataset de tweets. Coletámos também dados do Q&A da OMS sobre o coronavírus objetivando criar um corpus de referência. Por fim, também anotámos manualmente a relação (corroboração, contradição, neutralidade) entre cada frase de nosso corpus de referência, não só para servir de input para o modelo do segundo método, como também para atuar como ground thruth como objetivo de avaliar a componente de retorno de informações.

Com os resultados obtidos, concluímos que o pré-treino dos deep language models influencia diretamente a sua capacidade de classificação, tanto para a classificação de tweets como falsos como para a detecção de contradições. Porém, o pré-treino desses modelos não é suficiente para calcular a semelhança entre os embeddings de palavras dos tweets e frases dos nossos datasets, uma vez que as métricas obtidas para o retorno de informações pertinentes são bastante insatisfatórios. O fine-tuning dos deep language models apresentou melhores resultados para a classificação de tweets como falsos do que para o reconhecimento de contradições. Apesar de enfrentar várias limitações, principalmente no retorno de informações, propomos diversas ideias para melhorar o trabalho que desenvolvemos.

Keywords: Natural Language Processing, Contradiction Detection, Deep Learning, Information Retrieval, COVID-19, Social Media

Abstract

Misinformation is a threatening, increasingly common phenomenon in Social Media that spreads incorrect information to its many users. In the current pandemic context of COVID-19, the risks of misinformation and contradictions are significant due to their potential to cause dangerous emotional reactions, as happened when there was no certainty about the effectiveness of protective masks and their use. The evolution of Natural Language Processing (NLP) and associated techniques like Machine Learning and Transfer Learning make it possible for computers to process written human language, interpret it, and analyse it, providing the chance to address the spread of misinformation.

This thesis proposes a novel approach based on Supervised Learning Classification and Information Retrieval that automatically detects false information regarding COVID-19 on tweets. This conceived approach was implemented through two methods. The first method combines the fine-tuning of a deep language model pre-trained on a vast corpus for false tweet classification with retrieving information in a reference corpus through an Information Retrieval system. The second method relies on a pre-trained deep language model fine-tuned to identify contradictions between tweets and documents from a corpus of reference. All the models used are variations of the deep language model BERT.

In order to empirically evaluate our approach, we gathered distinct tweets labelled as true or false with information regarding the virus to create a dataset of tweets. We also collected data from the World Health Organization's Q&A about coronavirus to create a reference corpus. Finally, we also manually annotated the relation (entailment, contradiction, neutrality) between each sentence from our corpus of reference to serve as input for the model of the second method and act as ground truth for evaluating the retrieval of information. With the results obtained, we concluded that the pre-training of the deep language models directly influences its classification capability for both false tweet classification and contradiction detection. However, the pre-training of these models is not sufficient to calculate the similarity between the embeddings of words of tweets and sentences, as the metrics obtained for the retrieval of information are pretty bad. The fine-tuning of these deep language models presented better results for the tweet classification than for recognising contradictions. We faced several limitations, especially on retrieving information and propose several ideas to improve the study we developed.

Keywords: Natural Language Processing, Contradiction Detection, Deep Learning, Information Retrieval, COVID-19, Social Media

Acknowledgements

First of all, I want to express my condolences to all the people in the world that lost familiars or friends to the battle against COVID-19. May their souls rest in peace.

I want to express my gratitude to my supervisors, professor Carlos Soares and engineer João Costa. All the pieces of advice, counselling and feedback provided were indispensable to the execution of this research. Also, I want to thank Fraunhofer Portugal for the opportunity.

I couldn't reach where I am and become who I am without the love of my mother and my father, the ones that I love more than anything and that always gave me the strength never to give up and follow my dreams. Without all my family, which I love more than anything, this would be impossible for me to accomplish.

Also, thanks to all my friends that supported me in this hard path.

A special thank you to my friend Pedro Cabeça. Thanks for all the motivation you gave me. I hope you're taking care of me from up above. I will forever live with you in my heart.

Tomás Novo

"The only real revolution happens right inside of you."

J. Cole

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Main Objectives . . . 3
- 1.3 Document Structure . . . 3

2 Background and Related Work **5**
- 2.1 Supervised Learning for Classification . . . 5
  - 2.1.1 Classification Algorithms . . . 6
  - 2.1.2 Transfer Learning . . . 8
  - 2.1.3 Performance Metrics . . . 11
- 2.2 Natural Language Processing . . . 12
  - 2.2.1 Text Representation . . . 14
  - 2.2.2 Textual Information Retrieval . . . 19
  - 2.2.3 Contradiction Detection . . . 20
- 2.3 Covid-19 Misinformation Fight . . . 22
- 2.4 Summary . . . 23

3 Methodology **25**
- 3.1 Problem Formulation . . . 25
- 3.2 Method 1 – Classify & Retrieve . . . 26
- 3.3 Method 2 – Contradiction Retrieval . . . 27
- 3.4 Summary . . . 27

4 Datasets and Experimental Setup **29**
- 4.1 Datasets . . . 29
  - 4.1.1 Reference Corpus . . . 29
  - 4.1.2 Tweets Dataset . . . 30
  - 4.1.3 Annotations Dataset . . . 32
- 4.2 Experimental Setup . . . 32
  - 4.2.1 Models . . . 32
  - 4.2.2 Tokenization . . . 34
  - 4.2.3 Information Retrieval . . . 34
  - 4.2.4 Training, Evaluation and Testing - Performance estimation . . . 36
  - 4.2.5 Experiments . . . 38
  - 4.2.6 MisInfoBot - Twitter Setup . . . 39
  - 4.2.7 Environment and Frameworks . . . 39
- 4.3 Summary . . . 40

5 Results & Analysis **41**
5.1 *RQ1* – Pre-trained deep language models for false tweet prediction 41
5.2 *RQ2* – Pre-trained deep language models fine-tuning for false tweet prediction . . 42
5.3 *RQ3* – Similarity caption with TF-IDF / Pre-trained representations 42
5.4 *RQ4* – Pre-trained deep language models for contradiction detection 44
5.5 *RQ5* – Pre-trained deep language models fine-tuning for contradiction detection . 45
5.6 Analysis 47

6 Conclusions **49**
6.1 Answer to Hypothesis 50
6.2 Contributions 51
6.3 Future Work 51

A Results **53**
A.1 Classify & Retrieve method – Results Zero-Shot Learning 53
A.2 Classify & Retrieve method – Fine-Tuning Results 54
A.3 Classify & Retrieve Method – Training Validation 56

References **59**

List of Figures

1.1 Fake tweet 2

2.1 Decision Tree example 6
2.2 Architecture of an Artificial Neural Network 8
2.3 Structures of artificial networks used in NLP 9
2.4 A ROC curve plotted through Recall vs Fallout at different classification thresholds 13
2.5 Word2Vec both models: CBOW and Skip-Gram 16
2.6 Transformer model architecture 17
2.7 BERT tasks architecture 18

3.1 Method 1 – Architecture 26
3.2 Method 2 – Architecture 28

4.1 Organization of WHO's Coronavirus Q&A 30
4.2 Count of tokens per tweet of tokenizers of Classify & Retrieve method 35
4.3 Count of tokens per pair of tokenizers of Contradiction Retrieval method 36

5.1 Classify & Retrieve – ROC Curves 43
5.2 Contradiction Retrieval – M2M1 ROC Curve 46

A.1 Classify & Retrieve – Evaluation Accuracy 56
A.2 Classify & Retrieve – Evaluation Precision 56
A.3 Classify & Retrieve – Evaluation Sensitivity 57
A.4 Classify & Retrieve – Evaluation F1 Score 57

List of Tables

2.1 Confusion Matrix. . . . 11
2.2 Formulas of Performance Metrics . . . 12
2.3 Formulas of Performance Metrics . . . 20

4.1 Huggingface Deep Language Models . . . 33
4.2 Fine-Tuning Hyperparameters . . . 37
4.3 Classify & Retrieve – Tweets Dataset Split . . . 37
4.4 Contradiction Retrieval – Pairs Dataset Split . . . 37
4.5 Versions of used software . . . 39

5.1 Zero-Shot Learning results on fake post classification . . . 42
5.2 Fine-tuned models results on fake post classification . . . 44
5.3 Mean Average Precision results of IR for RQ3 . . . 44
5.4 Contradiction Retrieval – Zero-Shot Learning M2M1 results on contradiction detection . . . 45
5.5 Zero-Shot Learning M2M2 results on contradiction detection . . . 46
5.6 Zero-Shot Learning results on contradiction detection . . . 46
5.7 M2M1 Fine-Tuning metrics on contradiction detection . . . 46
5.8 M2M1 Fine-Tuning Confusion Matrix on contradiction detection . . . 46
5.9 Mean Average Precision on IR of contradictions . . . 47

A.1 M1M1 - Zero-Shot Learning Confusion Matrix . . . 53
A.2 M1M2 - Zero-Shot Learning Confusion Matrix . . . 53
A.3 M1M3 - Zero-Shot Learning Confusion Matrix . . . 53
A.4 M1M4 - Zero-Shot Learning Confusion Matrix . . . 54
A.5 M1M5 - Zero-Shot Learning Confusion Matrix . . . 54
A.6 M1M6 - Zero-Shot Learning Confusion Matrix . . . 54
A.7 M1M1 - Fine-Tuning Confusion Matrix . . . 54
A.8 M1M2 - Fine-Tuning Confusion Matrix . . . 54
A.9 M1M3 - Fine-Tuning Confusion Matrix . . . 55
A.10 M1M4 - Fine-Tuning Confusion Matrix . . . 55
A.11 M1M5 - Fine-Tuning Confusion Matrix . . . 55
A.12 M1M6 - Fine-Tuning Confusion Matrix . . . 55

Abbreviations

API – **A**pplication **P**rogramming **I**nterface
AUC – **A**rea **U**nder **C**urve
BERT – **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
biLM – **bi**directional **L**anguage **M**odel
BoW – **B**ag **o**f **W**ords
CBoW – **C**ontinuous **B**ag **o**f **W**ords
CNN – **C**onvulutional **N**eural **N**etwork
CWE – **C**ontradiction-specific **W**ord **E**mbeddings
DL – **D**eep **L**earning
DT – **D**ecision **T**rees
ELMo – **E**mbeddings from **L**anguage **Mo**dels
FN – **F**alse **N**egatives
FP – **F**alse **P**ositives
GloVe – **Glo**bal **Ve**ctors for Word Representation
GNN – **G**raph **N**eural **N**etwork
GPU – **G**raphics **P**rocessing **U**nit
IDF – **I**nverted **D**ocument **F**requency
IE – **I**nformation **E**xtraction
IR – **I**nformation **R**etrieval
LSTM – **L**ong **S**hort-**T**erm **M**emory
MAP – **M**ean **A**verage **P**recision
ML – **M**achine **L**earning
MLM – **M**asked **L**anguage **M**odel
MT – **M**achine **T**ranslation
NEL – **N**amed **E**ntity **L**inking
NER – **N**amed **E**ntity **R**ecognition
NLP – **N**atural **L**anguage **P**rocessing
NLG – **N**atural **L**anguage **G**eneration
NSP – **N**ext **S**entence **P**rediction
POS – **P**art **O**f **S**peech
PPDB – **P**araphrase Database
QA – **Q**uestion **A**nswering
Q-A – **Q**uestions - **A**nswers
RNN – **R**ecurrent **N**eural **N**etwork
ROC – **R**eceiver **O**perating **C**haracteristic
RTE – **R**ecognizing **T**extual **E**ntailment
RvNN – **R**ecursive **N**eural **N**etwork
SVM – **S**upport **V**ector **M**achine

TE – **T**extual **E**ntailment
TF – **T**erm **F**requency
TF-IDF – **T**erm **F**requency-**I**nverse **D**ocument **F**requency
TL – **T**ransfer **L**earning
TN – **T**rue **N**egatives
TP – **T**rue **P**ositives
WHO – **W**orld **H**ealth **O**rganization

Chapter 1

Introduction

Social Media are technologies that significantly impacted how people get information and spend their time [3]. Their rise made a huge global impact on politics and political deliberation, work and patterns of communication. Also, it changed the interactions with information of civic life, communities, dating, health, levels of stress, news consumption, parenting and teenage life by its various users [66]. Twitter is an important communication platform that has increasingly infused itself in the daily life of its many users life [59], with 336 million active users in 2018 [89].

However, these technologies also have their negative aspects, such as **misinformation**. This word was adopted to express false claims [80]. Social Media are one of the leading causes of its online spread [4], giving power to it as it is repeated and passed along from one person to another [21]. Twitter is not an exception, as due to its unmoderated nature, misinformation can spread easily and fast, reaching and endangering people worldwide [35].

Information integrity assumes vital importance in the safety and well-being of Social Media users, especially in critical events like the COVID-19 pandemic context. The novel Coronavirus (2019-nCoV) or the severe acute respiratory syndrome Coronavirus 2 (SARS-CoV-2) first appeared at the end of 2019 in Wuhan, China. This virus has a high spreadability, even higher than its ancestors SARS-CoV and Middle East respiratory syndrome Coronavirus (MERS-CoV) [82], alarming health institutions all around the globe [58]. As the pandemic spread worldwide, drastic measures were taken by countries to achieve the critical balance between economic activities and the spread of the virus. Some public health systems almost collapsed due to the increment of severe cases that demanded intensive care and even ventilation. This increase led to extreme measures like lockdowns, causing a positive impact on the restraint of the spread of Coronavirus but substantial negative implications in the countries' economies. Social distancing, personal hygiene, and mask utilisation to limit the virus spreading became imperative worldwide. Nevertheless, these intrusive measures and many other factors instigated the spread of wrong information about this virus in Social Media, endangering public health [34].

With this in mind, COVID-19 misinformation in Social Media needs to be fought through its

identification and unmasking in a way that can also promote education: through its expose against veracious information from reliable sources.

1.1 Motivation

> *" We're not just fighting an epidemic; we're fighting an infodemic "* - Tedros Ghebreyesus, Director-General of the World Health Organization (WHO), 2020

Although some platforms address this challenge of misinformation identification with different strategies, this problem is far from solved [5]. Several attempts lack efficiency given the difficulties of classifying information as truthful or not. Sometimes this distinction is made manually, making scalability impossible due to the dimension of these technologies.

An essential factor to be taken into account, as it determines the classification of a statement as true or not, is time. A clear example of the conditioning capacity of this factor is the various contradictions regarding the use of masks [16] throughout the epidemic: before it was stated that masks affected curbing the spread of the virus, it was claimed that they did not have such an effect. Figure 1.1 exposes one example of this phenomenon in the form of a tweet related to COVID-19. This tweet wrongly [1] claims that bleach and garlic cure COVID-19. In an environment with so many users, the spread of false information in the form of fake prevention measures for this dangerous virus, for example, can cause drastic consequences like the increase of cases.

Figure 1.1: Fake tweet

The issue of recognizing misinformation in Social Media may be addressed thanks to the evolution of techniques to process natural human language resultant of the use of this type of technology. This processing is called **Natural Language Processing (NLP)** and is often combined with **Deep Learning (DL)** and **Transfer learning** techniques in order to boost its performance. One relevant way to unmask misinformation is to identify and confront it against truthful information from reliable sources. **Information Retrieval (IR)** is one of NLP many tasks that, as the name denotes, has the goal of retrieving pertinent information regarding an inputted text. By

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub

accessing trustworthy information contained in a corpus, contradictions between its contents and textual information on Social Media may be identified. This way, education can be promoted to Social Media users, as misinformation is identified and faced against trusty information.

1.2 Main Objectives

The principal objective of this dissertation is to address the problem of misinformation related to coronavirus on Social Media by developing **MisInfoBot**, an automated bot for Twitter. It aims to recognize fake information and confront it by providing reliable claims. We also aspire to educate Twitter users, taking advantage of the educational potential that this social network offers. Hence, we formulated the following hypothesis:

Hypothesis – *Is it possible to use NLP methods to recognize misinformation and provide reliable information?*

To engage misinformation about COVID-19 and achieve this goal, we present an approach that aims to recognize misinformation present in tweets and retrieve truthful information. This approach was implemented through two methods. The first method has two components: the classification of tweets as true or fake through the fine-tuning of deep language models pre-trained in different corpus and retrieving information present in a reliable certificated corpus. The second method directly detects contradictions between tweets and information present in the same corpus of the first approach.

The main contribution of this thesis is the creation of a bot for Twitter that analyses the possibility of the text of a tweet to hold misinformation and retrieves reliable information. An important contribution of this work is the application of the proposed approach by using actual tweets and documents from the World Health Organization. Other relevant contributions are the obtainment of sequence classification results by testing several fine-tuned pre-trained deep language models and obtaining information retrieval results for the word embeddings produced by these models. The creation of a manually annotated dataset of semantic relations between the dataset of tweets and the reference corpus we used is also a pertinent contribution.

1.3 Document Structure

This document is structured in six chapters with the aim of providing an easy and organized reading. Chapter 2 presents fundamental concepts related to this dissertation, providing the existing background on Supervised Learning for Classification, Deep Learning, Transfer Learning, NLP, Text Representation, Textual Information Retrieval, Contradiction Detection and current procedures followed by Social Media platforms to fight against misinformation related to COVID-19. Chapter 3 specifies the developed approach and the two methods that implement it in order to achieve the main objectives of this dissertation. Chapter 4 not only describes the creation, details

and manipulation of the datasets utilized in this study but also addresses the followed experimental setup and its characteristics. Chapter 5 presents the outputs obtained from the executed experiments and an analysis of these outcomes. Chapter 6 contains the conclusions drawn from this work, summarizing its outcomes and contributions and possible improvements for futures scientific researches regarding its main topic. The chapters are followed by an Appendix that contains useful information regarding the experiments and the results of this thesis and by References taken into account to develop this study.

Chapter 2

Background and Related Work

This chapter contextualizes this dissertation by addressing the definition and background of fundamental related key concepts. Section 2.1 addresses Supervised Learning and Classification, presenting the principal methods used in classification tasks and describing several categories and approaches of Transfer Learning, as well as several metrics to evaluate the performance of classification models. Section 2.2 provides a brief explanation about Natural Language Processing and its applications, as well as several techniques to represent text mathematically, with particular emphasis on the BERT deep language model, as all models utilized rely on it. This section also addresses Textual Information Retrieval as well as metrics to evaluate an IR system and provides several definitions for the concept of *contradiction*, specifying different approaches to detect contradictions. Section 2.3 addresses methods that Social Media and researchers developed in order to identify misinformation regarding COVID-19. Finally, section 2.4 sums up the addressed topics of all covered sections.

2.1 Supervised Learning for Classification

Machine learning (ML) is an Artificial Intelligence sub-field dedicated to the scientific study of algorithms utilized to improve the resolution of a task based on previous experiences [22]. The use of **Supervised Learning** as an ML methodology is the dominant approach to address NLP problems, revealing impressive results. In Supervised Learning, models are trained to output predictions of undetermined target functions that can be represented by an input labelled dataset used to train the model and its outputs. A common procedure in training is to split the initial dataset for training, testing and validation. The training is made with the objective of refining the outputted predictions and generalize well to previously unseen data. Over-training should be avoided, as the model will memorize examples, lose generalizability and output wrong predictions for new inputs [71]. **Classification**, a Supervised Learning technique, has as goal the creation of a classification rule that, having as a base a training set where both class labels and features are

given, allows the prediction of classes of new objects whose features are available [39]. The problem of detecting contradiction in tweets may be addressed as a **Classification** task. Several algorithms can be used for this purpose, as discussed in the following section.

2.1.1 Classification Algorithms

Of the many algorithms used in classification problems, **Decision Trees (DT)**, **Naive Bayes**, **Support vector machines (SVM)**, and **Deep Learning** algorithms are commonly used. We'll use the classification of text as contradictory or not as an example with the purpose of simplifying the explanation of the addressed algorithms. In order to predict or not a contradiction, all presented models take into account the features from the used dataset.

2.1.1.1 Decision Trees

This classification model assumes the shape of a tree in order to classify instances by sorting them accordingly to the values of their features [42]. The tree is outputted considering the dataset features: starting by the root node on top of the tree, each node will represent a condition based on which the tree will split upside-down into branches. If a node does not split, we reached a leaf, i.e., a decision: either it is a contradiction or not [29].

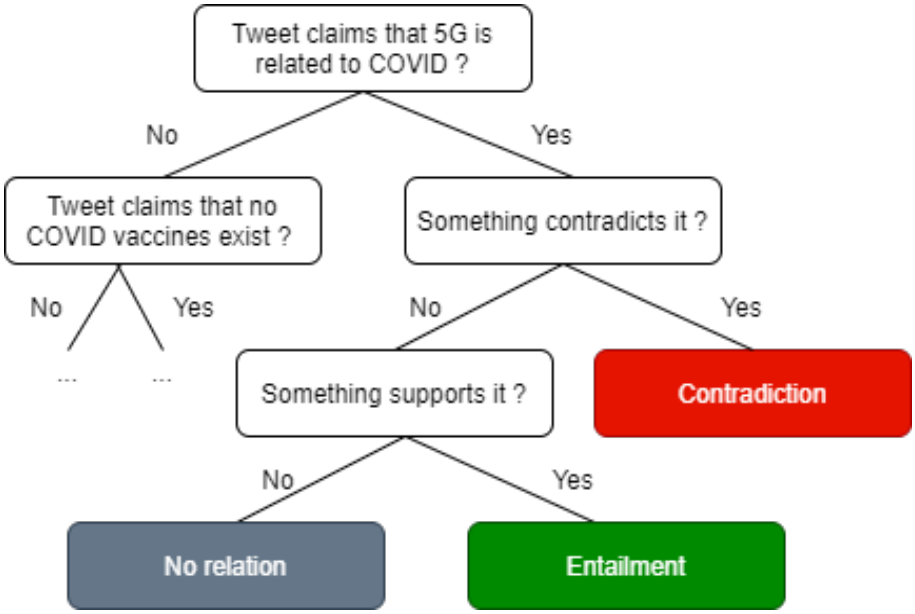


Figure 2.1: Decision Tree example

2.1.1.2 Naive Bayes

This classification method is probabilistic and relies on the Bayes theorem, which provides the probability of an event A (the existence of contradiction in a tweet, for example) given the occurrence of event B (the addressing of a specific Coronavirus topic by a tweet, for example), as

equation 2.1 shows.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{2.1}$$

Bayesian models for classification naively assume that features are independent in class prediction [24, 72]. The application of this probabilistic classifier in Web-Scale taxonomy presents poor performance due to its intrinsic limitations in problems involving contradiction pairs [94].

2.1.1.3 Support Vector Machines

Having n input features, a hyperplane splits a space with n dimensions into two, one containing positive examples and the other the negative ones. In the impossibility of separating the instances linearly, non-linear frontiers are modelled by implicitly projecting the data to a different space through a kernel trick where the inputs are mapped into high-dimensional feature spaces [69]. A regularization mechanism named *soft margin* deals with non-separable classes. An example of this classifier's application is in recognition of textual entailment, like the example of [10].

2.1.1.4 Deep Learning

The human brain inspires **Deep Learning** models as they use algorithms named **Neural Networks** whose structure has multiple layers, as illustrated by Figure 2.2. Neural Networks are composed of a large number of neurons, which are simple processing units. The neurons are organized in layers that act in cooperation. Weights connect every neuron of a layer with every other neuron of its preceding and succeeding layers. By adjusting these weights, the network can address a classification task [62, 95]. The network itself is a graph with multiple types of layers:

- **Input Layer** - single layer that receives input data
- **Hidden Layers** - variable number of layers that exist layers of input and output and that are responsible for intermediate calculations. Choosing the number of hidden layers and respective nodes has an impact on the model's performance.
- **Output Layer** - outputs the obtained prediction

Different existing architectures of neural networks can be chosen and therefore applied according to the task. For NLP tasks, the use of **Convolutional Neural Networks (CNN)**, **Recurrent Neural Networks (RNN)** and **Recursive Neural Networks (RvNN)** [63, 68] have been showing motivating results. The architectures of these networks are illustrated in Figure 2.3.

Convolutional Neural Networks are artificial networks with a specific layer that performs convolutions: a linear function applied to a matrix that gives the name to this model that also has pooling and fully connected layers.

Recurrent Neural Networks are a hierarchical network in which neuronal connections form a directed cycle, using information sequentially across the network's neurons. This sequential processing consists of a sub-sequential execution of the same task across the network.

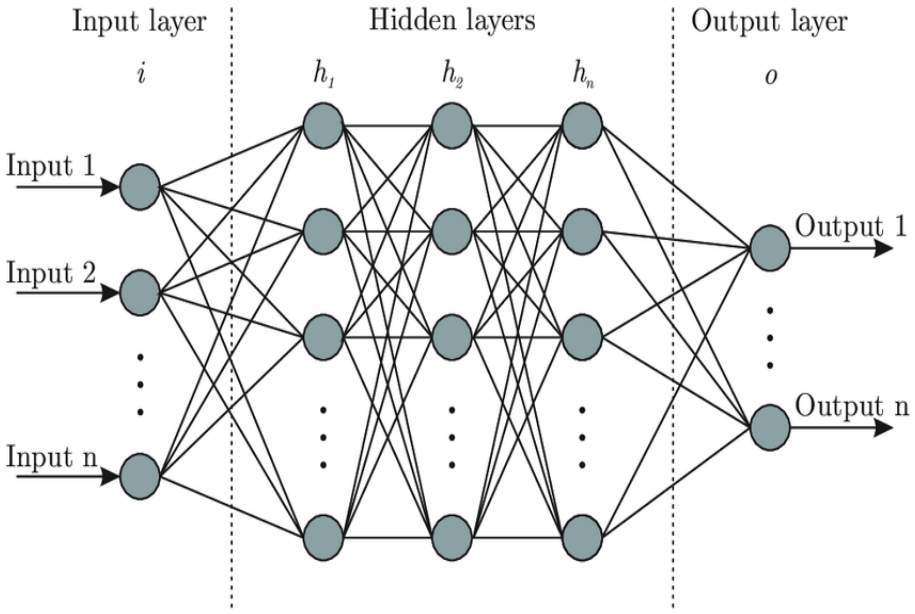


Figure 2.2: Architecture of an Artificial Neural Network. Extracted from [9].

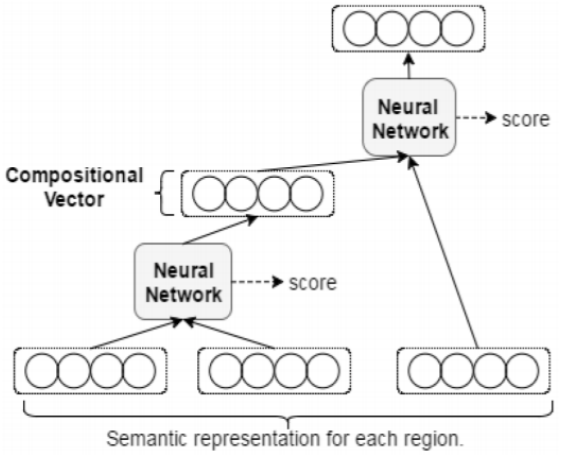
Hence, dependencies with the successive antecedent computations are generated every time the task is executed. By processing data sequentially, this type of network reveals itself useful in NLP tasks as the context of a text may be affected by the sequence of its words.

Recursive Neural Networks are a model that has a topological structure that is similar to a tree, as it recursively reproduces weights successively from low level to higher levels.

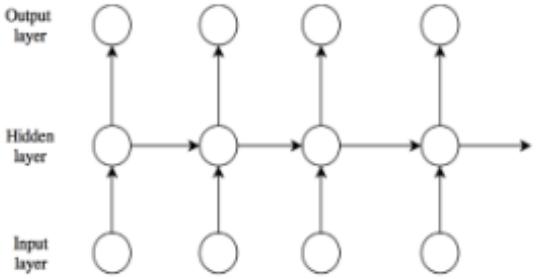
Neural Networks have variables that determine the network structure and the training of the network, the **hyperparameters**. The number of layers and units of each layer of a network is an essential hyperparameter. Another important hyperparameter is the number of epochs. One **epoch** consists of the forward and backward transmission of a whole dataset through a Neural Network. During an epoch, each dataset sample updates the inner parameters of the model. Hence, the chosen epochs number is significant as it is the number of times that the entire dataset will pass through the network. The **batch size** is also a crucial hyperparameter as it comprises the number of samples that the algorithm will process before assigning new values to the internal parameters of the network.

2.1.2 Transfer Learning

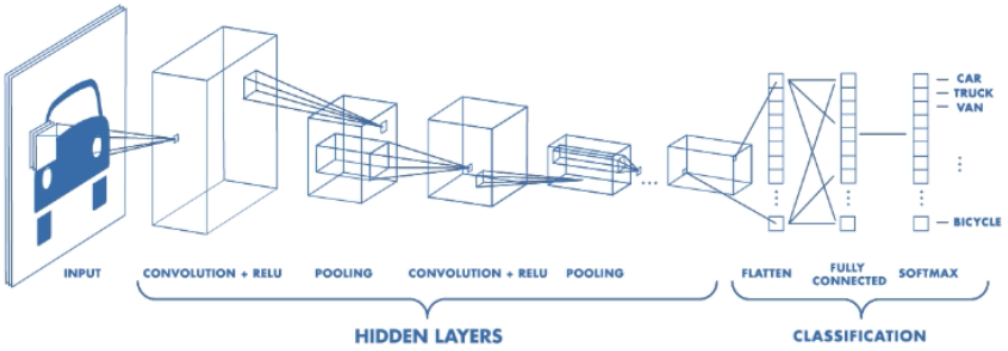
ML improved the results in many distinct knowledge engineering areas. However, most ML methods assume that data for training and testing are extracted from the same distribution and feature space. Changing the distribution can imply the rebuilding from scratch of many models and the use of new data for training. Rebuilding a model is expensive and sometimes even impossible due to the need to collect new training data [64]. The performance of a learner for a target domain trained from a source domain related to the target domain can be boosted by *Transfer Learning*.



(a) Architecture of an RvNN



(b) Architecture of an RNN



(c) Architecture of a CNN

Figure 2.3: Structures of artificial networks used in NLP. Obtained from [11, 68].

To improve a learner, this technique relies on the existence of some relation between domains [93]. Hence, it is possible to reuse the knowledge that was already learned between tasks and use it to approach new issues by taking into account training and testing sets from distinct distributions, domains and tasks. The transfer of knowledge, when done successfully, can improve the learning performance by avoiding expensive efforts of data-labeling [74]. Over the last years, various architectures and methods for transfer learning applied to NLP tasks emerged, improving their state-of-the-art. Three categories of transfer learning techniques can be distinguished, taking into account the different sources and distinct target tasks and domains:

- **Inductive Transfer Learning –** independently of their domains, target and source tasks are distinct. The target domain requires labelled data to induce a predictive model. Plus, labelled spaces may differ from source to target.
- **Transductive Transfer Learning –** there is a similarity between target and source tasks while their domains are distinct. The unavailability of labelled data in the target domain, unlike the source domain, has many data available.
- **Unsupervised Transfer Learning –** there is a difference between target and source tasks, but their domains are explicitly or implicitly related. Labelled data is unavailable in none of the domains, which leads to the conclusion that this technique targets unsupervised learning tasks.

On the other hand, there are several approaches according to the different types of knowledge that can be transferred, which are:

- **Feature-representation-transfer –** consists of the encoding of knowledge utilized to transfer across domains into a learned feature representation. Finding a good feature representation will reduce the divergence between domains and errors in classification models, improving the target task's performance. Distinct types of data of the source domain imply specific strategies to obtain good feature representations. The BERT language representation model, presented in section 2.2.1.3, uses this type of transfer learning in its pre-training.
- **Instance-transfer –** relies on the assumption that some data parts in the source domain may be reused through re-weighting in order to learn the target domain. The reuse of the data of the source domain is not done directly, while in the target domain the reuse of specific parts of data alongside few labelled data is common.
- **Parameter-transfer –** follows the assumption that parameters or hyperparameters prior distributions are shared by both tasks. This enables the encoding of transferred knowledge in the shared parameters or prior hyperparameters distributions.
- **Relational-knowledge-transfer –** follows the assumption that source domain data has a relationship similar to the relationship that data of the target domain has. This approach relies on the transfer of the data relationship of the source domain to the target domain.

Table 2.1: Confusion Matrix.

		Predicted Class		
		Contradiction	Not Contradiction	Total
Actual Class	Contradiction	**TP**	**FN**	**P'**
	Not Contradiction	**FP**	**TN**	**N'**
	Total	**P**	**N**	**T**

After considering each category's previous definitions, as in a Supervised Learning approach, labelled data is required on both source and target domains, the Inductive Transfer Learning technique should be followed. However, as in this work the transfer of knowledge is made at pre-training, and as the two pre-training tasks of BERT are unsupervised, as explained in section 2.2.1.3, Unsupervised Transfer Learning and Feature-representation-transfer are the techniques that we shall take into account.

We have discussed different types of models and their quality without explaining how they are evaluated. This is done in the following section.

2.1.3 Performance Metrics

Performance metrics are used to evaluate the classification performance of an NLP task. This evaluation measures the effectiveness of the classifier's predictions by comparing the model predictions versus the ground-truth labels in a **Confusion Matrix**, an essential concept in the evaluation of a model's performance. Table 2.1 provides an illustration of this concept in a problem of contradiction detection. The predicted class represents the classifications of the classifier as a contradiction or not, and the actual class consists of the ground-truth data.

As we are focusing on analyzing the presence of Contradictions, this is our positive class, while Non-Contradictions is the negative one. Synthesizing the content of Table 2.1, it can be said that **TP (True Positives)** stands for the number of correct classifications of text as positive for contradiction made by the model. Oppositely, **FP (False Positives)** is the number of incorrect predictions of text as positive by the algorithm. Similarly, **TN (True Negatives)** is the number of correct classification of text as negative and **FN (False Negatives)** stands for the number of incorrect classification of text as negative for the presence of contradiction by the model. Summing these four values, the total amount of predictions done by the classifier is obtained. With these values, several essential metrics for the classifier can be calculated in order to evaluate its performance, such as the ones presented in Table 2.2.

Analyzing the metrics of this table, **Accuracy** is the proportion of correct results among all the analyzed cases. However, only taking a look at the Accuracy value may lead to incorrect conclusions when the dataset is not balanced as it assumes equal costs for both **FP** and **FN**. **Precision/Specificity** takes into account the proportion of the truly predicted positives among all predictions as positive (class of interest). A high precision value signifies that the identification of the class of interest is often correct. Similarly, **Fallout/Sensitivity** takes into account the proportion

of the truly predicted negatives among all predictions as negative. By calculating **Recall**, we obtain the proportion of actual positives that were properly classified. The **F1 Score** is the harmonic mean between Precision and Recall. An also important metric to take into account that is not present in the table is **ROC (Receiver Operating Characteristic) Curve**. This plot, represented by Figure 2.4, demonstrates the distinguishing ability of a binary classifier system according to the variation of its discrimination *threshold*. This threshold consists of the value of classification probability that a class score must exceed to recognize a sample as belonging to the same class. By plotting the Recall against the Fallout at various threshold settings, we obtain a ROC curve. The AUC (Area Under the Curve) metric demonstrates the probability of a classifier to rank a positive sample chosen arbitrarily higher than an arbitrarily chosen negative one. Hence, AUC illustrates how much a model is capable of distinguishing classes. The highest de AUC, the highest the capability of the classifier to determine classes.

Table 2.2: Formulas of Performance Metrics

$$Accuracy = \frac{TP+TN}{T}$$	(2.2)
$$Precision/Specificity = \frac{TP}{P}$$	(2.3)
$$Recall/Sensitivity = \frac{TP}{P'}$$	(2.4)
$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$	(2.5)
$$Fallout/FalsePositiveRate = \frac{FP}{N'}$$	(2.6)

2.2 Natural Language Processing

Natural Language Processing (NLP) consists of a set of computation procedures that analyze and represent natural human language for applications or tasks [46], such as:

Dialogue Systems – establishment of a conversation with a computer system [12]

Information Extraction (IE) – processing of unstructured machine-readable documents in order to extract structured information from them [61]

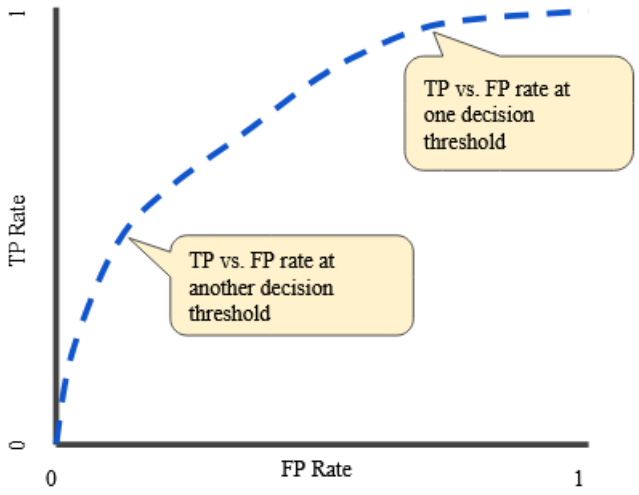


Figure 2.4: A ROC curve plotted through Recall vs Fallout at different classification thresholds. Reproduced from [15]

Machine Translation (MT) – the conversion of one natural language into another, producing fluent output text that preserves the meaning of the input text [83]

Named-entity linking (NEL) – matching of entities in a text such as individuals, companies, locations, etc. with their corresponding entities in a knowledge base [79]

Named-entity recognition (NER) – identification of named entities present in unstructured text and their classification into a category such as a person, location, company, etc. [43]

Part-of-speech tagging (POS tagging) – grammatical classification of words (as noun, verb, adjective, determinant, adverb, etc.) through the analysis of their context and definitions [36]

Question-Answering (QA) – answering natural language questions made by humans [54]

Summarization – extraction of the synopsis from texts while preserving its essential information and meaning [81]

Textual entailment (TE) – verifies if what is claimed in a fragment of text is followed by another text [27]

Information Retrieval (IR) is another relevant NLP task as it allows the retrieval of relevant information contained in documents, as presented in Section 2.2.2.

The processing of data in the form of natural human language has the potential to induce the creation of many original applications and revolutionize interactions with apps, websites and devices. The evolution of Social Media has reshaped the amount and the types of NLP data available. The impressive and fast developments in ML and a colossal growth in this availability of data and in computing power made possible new linguistic interactions and created new possibilities for novel applications.

One complex application of NLP is the creation of many helpful automated bots. A subfield of NLP named *Natural Language Generation (NLG)* focuses on the production of meaningful information through natural language by using knowledge about the same language and the application domain to generate different types of text [70] automatically. This sub-field allowed the creation of several thousands of distinct chatbots specified for different tasks [18] and even voice-driven digital assistants like Siri (Apple), Cortana (Microsoft), Alexa (Amazon), etc., by combining the processing of human language with the use of Deep Learning models.

However, most algorithms do not process text directly as their performances depend on mathematical calculus. Instead, these models often use numerical representations of words. As these representations affect directly executed calculations of the model, concise text manipulations are necessary to represent words with quality. Hence, several conversion techniques of words to their numerical representation are presented in the Section below.

2.2.1 Text Representation

Methods that are already a little outdated but were important in the evolution of text representations are Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). Recent developments like **distributed** representations assume relevance in representing text as they made the previously referred methods obsolete.

Bag-Of-Words (BoW) representation was a popular method used in NLP tasks such as information retrieval and document classification. In this representation, a set of words and the associated frequency of each word represent a document. This way, words can be represented by a vector that can act as input for classifications [96] and other tasks. Hence, when used in Supervised Learning classification, the features to train a model are word frequencies. Many text classification methods used this representation due to its simplicity for purposes of classification [60]. Nevertheless, this method has the limitation of not taking grammar neither the order of text words into account [37].

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure used to examine the relevance of keywords to documents in a corpus. It is a text representation based on statistics that express the keywords' relevance to documents by measuring the **Term Frequency (TF)** and **Inverse Document Frequency (IDF)**. TF measures the quantity of times that a term appears in a document while IDF assigns different weights according to the frequency of words (frequent words have lower weight and infrequent words greater weight) across the set of all documents. TF-IDF results of the multiplication of these two values [30]. However, this representation presents limitations, as semantics are not taken into account.

However, the main issues of these two representations are the dimensionality, as data becomes sparse with its increase and its difficulty in extracting meanings as each word is represented individually. Through the grouping of identical words, the performance of learning algorithms in NLP tasks can be improved by **distributed** representations of words in the space of vectors [52]. By mapping words into a vector space in which each dimension consists of a feature, it is possible to find similarities between words as related words own more similar representations, addressing

the obstacles that the previous representations could not surpass.**Word Embeddings** are a type of representations that have the capacity to encode words' syntactic and semantic relationships [6]. There are two types of Word Embeddings: **Non-Contextualized** or **Global** (word representation is independent of its context) and **Contextualized** (takes context of the word into account in its representation), addressed in Sections 2.2.1.1 and 2.2.1.2, respectively.

2.2.1.1 Non-Contextualized Word Embeddings

Of the non-contextualized word embedding techniques stands out **Word2Vec**[1], a Google's neural network for text processing that has two learning models as base [49]:

Continuous Bag of Words (CBoW) this model predicts a target word having its nearby words in consideration, i.e., its context. Prediction is not influenced by order of the words that form the context, originating a bag of words that uses a continuous distributed representation of the context, unlike the standard BoW. Each input word is encoded as a one-hot vector so a weight matrix can map it to the network's hidden layer. This layer's neurons copy to the output layer the sum of the inputs' weight, outputting values resulting from a Softmax function, which transforms a vector received as input to a probability distribution [75]. The architecture of this model is presented in Figure 2.5a.

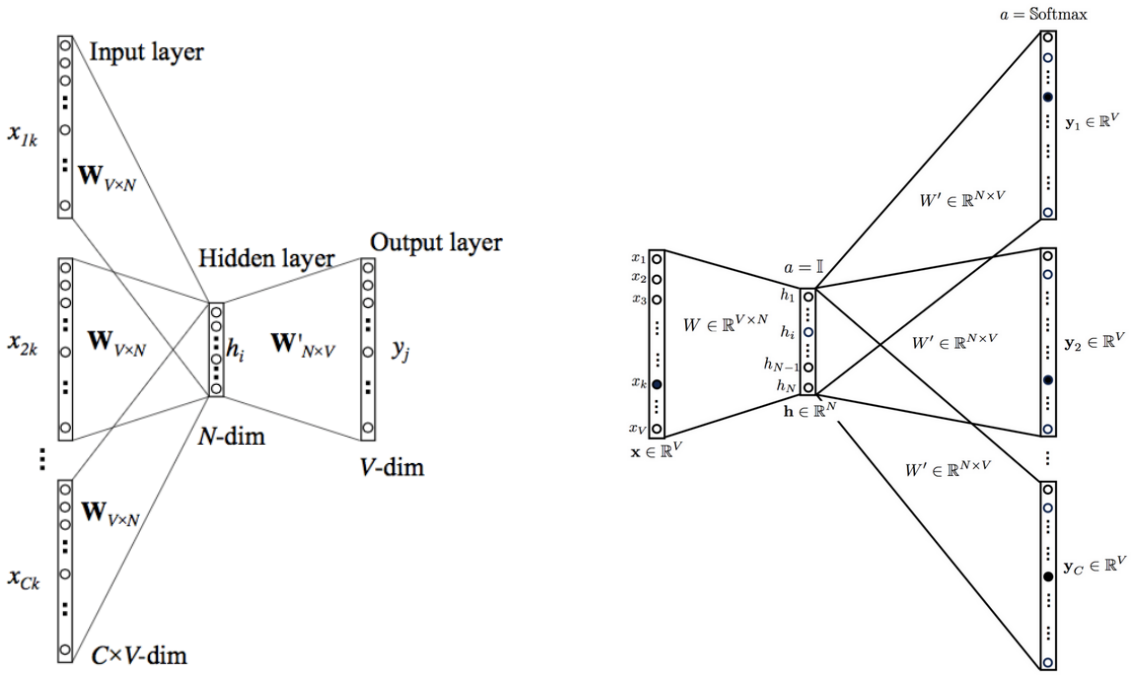
Skip-Gram given an input word, this model can predict target context words. Words are represented through a one-hot vector, considering a window of surrounding words (the context). Skip-Gram outputs the probability distributions for the context words. Figure 2.5b presents this models' architecture.

Each model has its advantages and disadvantages [38]: CBOW is faster, and its representations for words that appear frequently are better than Skip-Gram, which proves to have better performance with fewer data and presents better representations for unusual words. Coupled with RNNs and CNNs, Word2Vec is commonly used in the creation of chatbots and many other applications.

Another important non-contextualized word embedding is **Global Vectors for Word Representation (GloVe)**[2]. This word vector technique is built on the idea that semantic relationships between words can be derived from a co-occurrence matrix obtained from a corpus. It consists of a count model that considers how many times a word has co-occurred with other words [65]. The advantage of GloVe over Word2vec is that it not only relies on the local context information of words but also on word co-occurrence, i.e., it relies on local and global statistics to generate word vectors [25].

Thereby, non-contextualized word embedding techniques output a matrix that maps words into vectors typically used by neural networks. However, their representation is independent of the context, which may be a problem as the context assumes importance and should be tacked into account when representing a word. For example, in the sentences "You are right" and "Turn right",

[1] https://code.google.com/archive/p/word2vec/
[2] http://nlp.stanford.edu/projects/glove/



(a) CBOW model (b) Skip-Gram model

Figure 2.5: Word2Vec both models: CBOW and Skip-Gram. Obtained from [73]

the word *right* assumes different meanings. With this in mind, context is an important factor to consider when representing words.

2.2.1.2 Contextualised Word Embeddings

In this type of embedding, the representation of a word depends on its context, an indispensable factor that non-contextualized word embeddings do not consider.

Considering the context, **Language Models** are statistical models that calculate the probability of a word appearing next in a sentence, taking its words into account [85]. **Long Short-Term Memory (LSTM)** models consist of a unique RNN that maintains long-term dependencies due to its internal memory, revealing itself valuable in word representation modelling. Despite their potential, LSTM neural models are not fast since they cannot be parallelized and perform process data sequentially [84, 86].

Embeddings from Language Models (ELMo)[3] is a deep contextualized word representation resulting of a bidirectional LSTM pre-trained on a vast corpus. The representations provided by this model are contextualized as they consider semantics and syntax, recognized by the LSTM, in order to represent words. The generated vectors of words are learned functions of computed inner states of a deep bidirectional language model (biLM) that perform a linear combination of each internal hidden layer's vectors. Furthermore, ELMo can be easily incorporated into existing NLP models [67].

[3] https://allennlp.org/elmo

Transformer[4] is an encoder-decoder architecture developed by Google that has mechanisms of attention as a base, represented by Figure 2.6: the left half represents the encoder and the right half the decoder. This model relies on **multi-headed self-attention**. The attention mechanism allows the model to identify relevant parts in a sequence. This mechanism is used in different ways with diverse objectives: In "encoder-decoder attention" layers, the previous decoder layer provides the queries while the memory keys and values are results of the output of the encoder, allowing every position in the decoder to take into account every position in the input sequence. This way, to compute the next representation for a given word in a sentence, this model compares it to every other word, resulting in an attention score for the other words. The scores define the contribution of the other words of the phrase in the initial word's representation. Regarding computational complexity, recurrent layers of RNNs are slower than self-attention layers. Experiments on Machine Translation tasks revealed performances in which these models required less time to train and were more parallelizable [92]. However, for tasks of fact-checking or text classification, this model reveals itself insufficient. Nevertheless, this model served as a basis for developing the BERT model presented in the following subsection.

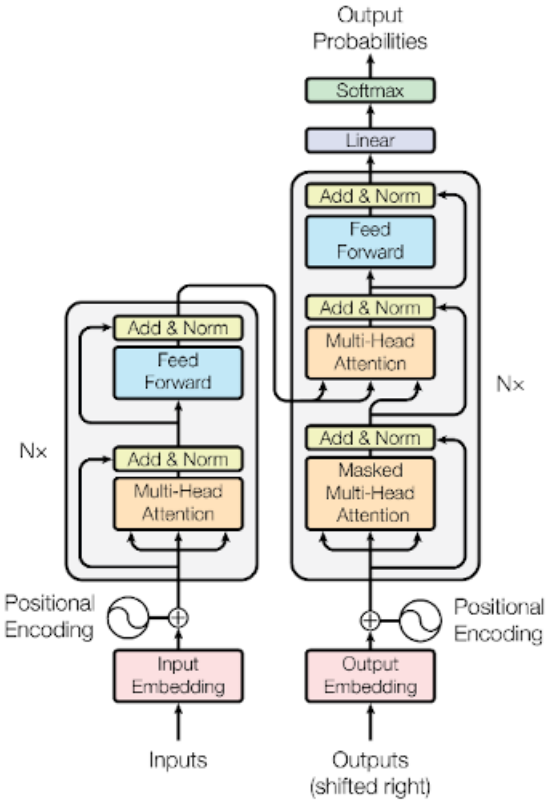


Figure 2.6: Transformer model architecture. Extracted from [92]

[4] https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

2.2.1.3 Bidirectional Encoder Representations from Transformers - BERT

The **BERT**[5] language representation model was designed by Google to pre-train bidirectional representations from unlabelled text by jointly taking right and left context into account in all layers, resulting in a simple and empirically powerful model. BERT can be fine-tuned with only one additional layer for outputs in order to address supervised-learning classification tasks. The model can have different sizes and thus different configurations despite being based on the Transformer's encoder. One of the qualities of this model is its easy adaptation to the resolution of several tasks [20]. Regarding **Input/Output Representations**, BERT input representation can represent in one token sequence a single or a pair of sentences in an unambiguous way. A special classification token [CLS] is always the initial token of sequences. The final hidden state corresponding to this token is utilized as the aggregate sequence representation for classification tasks. A particular token [SEP] is utilized to distinguish sentences, and to every token is added a learned embedding to indicate to which sentence the token bellows. The model **pre-training** is executed on a large corpus (800M words from BooksCorpus [97] and 2,500M English words from Wikipedia) to identify language patterns by using two unsupervised tasks: **Next Sentence Prediction (NSP)** and **Masked Language Model**. In NSP, BERT predicts how likely a sentence is to follow another to improve its capability to understand relationships between sentences. In the second task, a percentage of tokens of input is randomly masked and then predicted. BERT's **Fine-tuning** is made by plugging in the specific inputs and outputs of the desired task BERT and fine-tuning every parameter end-to-end. By fine-tuning this model, BERT can be easily adapted to many tasks. Figure 2.7 presents pre-training and fine-tuning architectures in BERT.

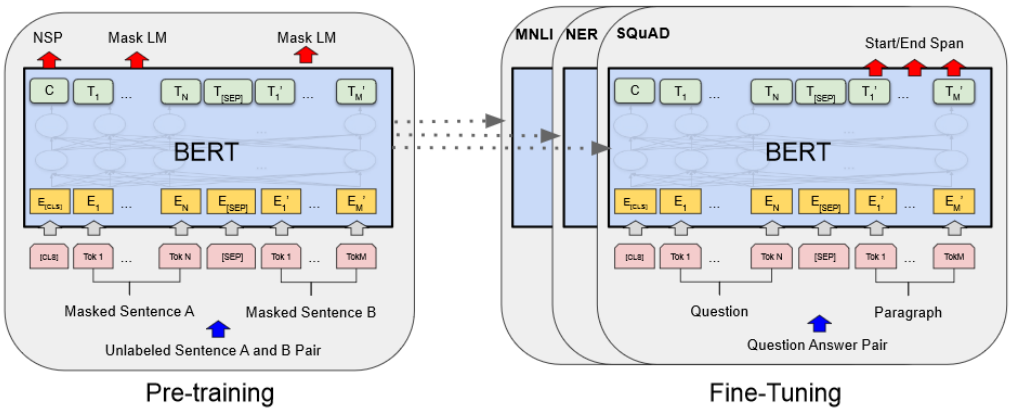


Figure 2.7: BERT tasks architecture. Extracted from [20].

On their works, [48] propose roBERTA, a version of the BERT model that includes longer training (more batches and data) and on longer sequences, dynamically switches the masking pattern that is applied in data used in training and does not include Next Sentence Prediction.

[5] https://github.com/google-research/bert

The use of textual representations are essential to address many NLP problems, and the quality of this representation may affect the performance on tasks like Information Retrieval. Text representations assume relevance in this particular task, as we present in the Section below.

2.2.2 Textual Information Retrieval

Given a query, an *Information Retrieval (IR)* system should be able to identify relevant information to the query from its collection of documents. With the objective of indexing and obtaining valuable information, each document in a corpus is represented as a vector of features and its associated weights. These features represent the specific content of each document. By converting custom queries to an identical representation, the similarity of features between documents' and queries' representations and consequently their relevance can be analyzed in order to return the best information possible [55]. The words of two distinct texts may be identical lexically or semantically [28].

2.2.2.1 Lexical Similarity

The lexical similarity results of the similarity between the sequence of characters of two words. String-Based algorithms usually measure this similarity. By operating on sequences of strings and on the composition of characters, these algorithms measure similarity between strings. Hence, these methods allow the measuring of similarity or dissimilarity/distance between two strings. The measurement may be done based on characters or on terms.

One of the principal Character-Based measures of similarity is **N-gram**, which relies on comparing the sub-sequence of n characters/words of two string sequences. Some of the main Term-Based measures of similarity are **Cosine similarity** (based on the measurement of the cosine of the angle between vectors) and **Euclidean distance** (square root of the sum of squared differences between elements of two vectors).

2.2.2.2 Semantic Similarity

The semantic similarity may be identified if two words are equal, if two words are opposite, if two words are utilized in an identical context, if two words are used in an identical way or if one word results in a variation of the other. To measure this similarity, Corpus-Based and Knowledge-Based algorithms are used. The first one measures words similarity relying on the information present on a vast corpus. The second identifies the degree of word similarity through the use of information from semantic networks.

2.2.2.3 Metrics

In order to evaluate an Information Retrieval system and its effectiveness and reliability, several metrics can be used to examine its performance in a set of queries and documents [76]. The main metrics [2] utilized to evaluate IR systems are presented in Table 2.3. **Precision at K** takes into

account all the K documents retrieved by the system, measuring the relevant documents between the retrieved documents for a query. **Recall at K** measures the relevant documents retrieved for a query, considering all documents. **F-Score** is the weighted harmonic mean of Precision and Recall. **Average Precision** measures the average of precision of multiple queries, while **Mean-Average Precision** is the mean of Average precision across multiple queries.

Table 2.3: Formulas of Performance Metrics

$$Precision\ at\ \mathrm{K} = \frac{Number\ of\ Relevant\ Documents\ in\ \mathrm{K}\ documents\ retrieved}{K} \quad (2.7)$$
$$Recall\ at\ \mathrm{K} = \frac{Number\ of\ Relevant\ Documents\ in\ \mathrm{K}\ documents\ retrieved}{Number\ of\ total\ relevant\ documents} \quad (2.8)$$
$$F\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.9)$$
$$Average\ Precision = \frac{Sum\ of\ Precisions}{\text{Number of Total Relevant Documents Retrieved}} \quad (2.10)$$
$$MAP = \frac{\Sigma_{q=1}^{Number\ of\ Total\ Queries} Average\ Precision(q)}{Number\ of\ Total\ Queries} \quad (2.11)$$

The retrieval of information is often associated with an NLP task named **Fact Checking**, which aims the verification of factual information through comparisons with several sources to preserve truth. However, in Social Media, some claims contradict truthful sources. An overview of several methods already developed to identify contradiction is presented below.

2.2.3 Contradiction Detection

The definition of contradiction varies among the various authors that have tried to detect it successfully in diverse ways. For [32], contradiction results from the incompatibility of information in two distinct texts and can be identified through linguistic information (negation, antonymy and information semantic and pragmatic) by facing the recognition of contradiction as a problem of classification that operates on the result of textual entailment of two input texts. Their work provided the first empirical results for the detection of contradiction, with good outcomes

on experiments in two corpora: the first with contradictions resulting of the negation of each entailment from the PASCAL Recognizing Textual Entailment (RTE) dataset and another based on paraphrases. [19] also used RTE datasets, annotating them and balancing them (between entailments and non-entailments), as well as collecting real-life contradictions from various sources. For the authors, a contradiction can be defined as the improbable occurrence of two sentences regarding the same event being true simultaneously. They distinguish two types of contradiction: the first one occurs via negation, antonymy and mismatch of dates and numbers and is easier to identify than the other contradiction type, the ones that result from using factive or modal words, structural and subtle lexical contrasts and world knowledge. The outcome of their work was a system where the mismatch between text and hypothesis is the base of where contradiction features are extracted, through the conversion of the same text and hypothesis to typed dependency graphs generated by Stanford parsers, then applying logistic regression to classify the pairs. With this in mind, seven features that reflect contradiction patterns are considered: Polarity features – identify presence/absence of negation linguistic markers through negation dependencies in the typed dependency graph or through specific negation linguistic markers such as "no", "not", or "few"; Matches or disagreement between dates, numbers, and time; Antonymy features – comparisons with WordNet's antonyms and contrasting words [53] and verbs of opposition from VerbOcean [14]; Structural features – determination of syntactic structures through overlaps of subjects with objects and vice-versa; Factivity features – based on words of factivity; Modality features – catch of patterns of modal reasoning through the presence of markers of modality like "can or "maybe"; Relational features – detection of relations between elements in the text. The outputs reflected a lack of feature generalization on the second defined type of contradictions, as expected.

Another definition of contradiction is adopted by [23] as they define the identification of contradiction as the detection of statement pairs that convey information about actions or events that are impossible to hold simultaneously. However, their work takes uncertainty assessments into account. Linguistic patterns or words that express speculation, beliefs or thoughts such as "I believe", "I assume", "it seems" or lexical clues like "probably", "might be", "it is unlikely", etc., hedges (mitigating words like "certainly", "possibly", etc. that modify the uncertainty assigned to propositions), modal verbs and forms of passive-active language allow to recognize uncertainty. Their contradiction detection model relies on a disconnected analysis of factual information conveyed by sentences and uncertainty assessments appended to the same information. Thereby, contradictions are detected through degrees of conflict and disagreement between sentences. A conflict between two sentences is a relation where statements approach the same topic with their content revealing opposition or contradiction but have similar or equivalent uncertainty assessments. In contrast, in disagreement, the shared topic's content is identical in the two statements but without agreement on the uncertainty assessments. The contradictory or opposition content can be identified at a lexical level by taking negation, antonymy, numerical mismatches or world knowledge into account.

Time is a factor that is taken into account to define contradiction in the works done by [87], distinguishing two types of contradiction: synchronous and asynchronous. The approach they

developed relies on the fine-grained extraction of sentiments: an author's expressed sentiment about a topic is a number between the interval [-1,1], indicating the opinion polarity. Negative values represent negative opinions, while positive values define positive ones. The absolute value portrays the strength of opinion. Sentiments can be aggregated by calculating the mean value of individual sentiments in a compilation of documents that address the same topic. A high variance of this value or its proximity to 0 denounces contradiction regarding a topic. Thus, to detect contradictions, first, each sentence's topic needs to be detected so that the sentiments for each pair of sentence-topic can be detected. The sentiments about the topic are then analyzed across multiple texts. The conception of a time-tree structure to store contradictions regarding a topic grants scalability to their developed method as nodes that correspond to time windows and sum up sentiments information present in all analyzed documents.

For [44], when sentences are unlikely to be correct simultaneously, it results in a contradiction. Facing the detection of contradictions as a classification problem, their classifier relied on the relationship between semantic relations representations from input texts, i.e., word embeddings. However, classic context-based models for word-embedding like Word2Vec or Glove map words with identical context into closed vectors: words like "crowded", "empty", and "overfull", which have different meanings, have similar representations. To tackle this issue, [56] applied knowledge at a lexical level like PPDB (Paraphrase Database) [26] and WordNet [53] to utilize pairs of antonym and synonymy to review the embedding of words. [13] and [47] also used WordNet jointly with Thesaurus to develop methods that extract a limited quantity of antonym pairs from this lexical resources to generate constraints of semantic so that vector representations of words could have a more accurate similarity. Also, [77] extracted antonyms from Wikipedia by utilizing patterns but had difficulties with data sparsity. Having these approaches as base, [44] developed a method that automatically creates a corpus from PPDB and WordNet with many contrasting pairs of words phrases. A feedforward neural network that learnt contradiction-specific word embedding (CWE) was developed and optimized by minimizing the gap of semantics between pairs of paraphrases and its maximization for pairs of contradiction. The input sentences' semantic relations are represented through the learnt embeddings that act as features for a developed Convolutional Neural Network model to detect contradiction.

2.3 Covid-19 Misinformation Fight

The presence of misinformation in Social Media platforms is a consequence of the freedom of expression they grant. To unmask false information, these platforms follow different approaches.

Facebook – the development of classifiers based on computer vision assist temporary bans in ads and commerce listings for protective masks and coronavirus' related products, sometimes even in a proactively way. Warning labels are often assigned to fake content regarding the virus by independent fact-checking partners, reducing its distribution and showing these labels, proving to be an effective technique. SimSearchNet, a CNN model, was also built

to recognise near-exact duplicates that share fake information through images [91]. In their works, [45] created Jennifer, a Facebook chatbot based on QA pairs that address various topics related to the pandemic.

Instagram, TikTok, Youtube – works with third-party fact-checkers to recognise, revise, and label false information, reducing its distribution. Users are encouraged to read the latest news from official sources like the WHO website when the content is related to the virus. To tackle misinformation on YouTube, [50] developed a multi-label NLP classifier relying on Transfer Learning to detect misinformation expressed as conspiracy comments.

Twitter – to identify the spread of false information through the analysis of tweets regarding coronavirus and conspiracy theories related to 5G, [31] performed two tasks: the identification of text-based misinformation and the recognition of structure-based misinformation. BoW and BERT word embeddings were used for the first and Graph Neural Networks (GNNs) for the second. Twitter also uses the technique of applying labels to tweets that hold misleading information or disputed claims about the virus when warnings are not considered as enough. Still, the chance to reply, retweet, or like to tweets can be blocked, and the tweet can even be removed if the misinformation shared is severe. The labels contain a link that redirects users to more information. This work is done by a Curation team that organises and presents existing content by finding and highlighting text, videos, images and live streams existing in tweets [90, 88]. However, everything indicates that this process is carried out manually. Twitter, as one of the most popular Social Networks, has many features. Many interactions can take place when a user writes a tweet and posts it. These interactions consist of reacting to a tweet, writing a reply to it, or retweeting the initial tweet, sharing it with the possibility of writing a commentary. Tweets hold distinct data types (images, links, text, videos, etc.), and misinformation on Twitter may spread through different types of data and interactions. This spread is not only made by its many users but also by automated accounts the bots. Regarding the COVID-19 "*infodemic*", several bots combine the users' susceptibility to trust and share fake information with sharped strategies to achieve its dissemination [33]. The naivety and bad intentions of users requires prudence in misinformation fighting, as several users believe that they're sharing truthful information and tend to share even more misinformation when corrected by other users. This motivates the creation of false information and even toxicity in the interactions between users as they don't like to be corrected by other users [1].

2.4 Summary

In this Chapter, we discussed the background and state-of-the-art on the topics related to this project.

We addressed distinct algorithms used in the classification tasks of Supervised Learning. Of all presented methods, Neural Networks are the ones that assume the most relevance for this project

as several types of these algorithms are often used in NLP tasks. This Chapter also presented Transfer Learning and distinct ways to transfer the knowledge earned by a model to another domain, avoiding expensive re-building from scratch and efforts. To evaluate the performance of a classifier, several metrics can be calculated by relying on confusion matrices, as shown in this Chapter. Accuracy, Precision, Recall, F1, Fallout and AUC assume the most relevance as they expose the classifying capability of a model.

As we want to unmask misinformation present on Twitter in the form of text, Natural Language Processing reveals itself indispensable. The evolution of text representations, as demonstrated, was significant in order to represent words with more quality. BERT assumes relevance between all presented representations as it is a deep language model that can be fine-tuned to solve specific NLP tasks. Regarding the retrieval of information, several similarities and distances can be calculated in order to use a sentence as a query in a corpus of relevant documents. Some metrics to evaluate the retrieval performance were also presented in this Chapter. Some approaches to detect contradictions were also provided, and several approaches to fight misinformation regarding COVID-19 on Social Media.

Chapter 3

Methodology

This chapter describes the developed methodologies to fight misinformation related to COVID-19 on Twitter. Section 3.1 describes the approach we propose and its related concepts that gave origin to two distinct methods, explained in detail in Sections 3.2 and 3.3, respectively. Section 3.4 summarizes the information presented in this chapter.

3.1 Problem Formulation

Our approach relies on the implementation of a Twitter bot that aims to tackle misinformation present in tweets' content at a textual level and consists of two major sub-tasks: the recognition of the presence of misinformation in a tweet and the retrieval of information that goes against it. Thus, this is precisely what we aim for with the approach we implemented. By analysing the textual content of the text of a tweet, we want to verify if its claims contradict any document that exists in a corpus and retrieve contradicted documents. If the document is contradictory, the tweet is classified as misinformation as its text goes against claims from reliable sources.

Formalizing mathematically the problem we want to address with our approach, having a collection of N texts , $T = T_1, T_2, ..., T_N$ and a group of Y relevant documents $RD = RD_1, RD_2, ..., RD_Y$, we want to retrieve a subset of K documents $SD = SD_1, SD_2, ..., SD_K$ such that $SD \subset RD$ with documents that are contradicted by a text N.

We implemented our approach through two methods. The first solution addresses the discussed problem by individually executing a classification and a retrieval task in order to detect contradictions against the corpus. In contrast, the second methodology tries to address them both simultaneously. Both methods rely on Supervised Classification models trained on labelled datasets to acquire a domain and task knowledge. On both methods, we utilized several deep language models pre-trained in large specific corpus to take advantage of their pre-training by transferring their learned knowledge and amplifying our variety of obtained results. In order to test our approach,

we explored the COVID-19 pandemic context as a case scenario, as we used tweets related to the virus and information provided by WHO regarding the virus.

3.2 Method 1 – Classify & Retrieve

The first method that we developed intends to directly address the truthfulness of a text by classifying it as true or false and retrieving information that contradicts it. With this in mind, the structure of this method is formed by two major components: a model to classify the veracity of a document and an Information Retrieval system.

Regarding the first component, the Supervised Learning classifier model was fine-tuned to categorize text as true or false. Hence, we are facing a binary classification problem in which the model calculates the probability of a text to contain misleading information, attributing a class (0 if fake, 1 if true) to it. By assuming a threshold of 0.5, this assignment is made according to the class with the highest probability value. If the model considers that the text of a tweet holds false information and thus classifies the tweet as false, the tweet will be served as input for the second component of this method.

The Information Retrieval system has access to a reference corpus that contains information regarding several topics. When a text is classified as false, this component uses text as queries in this corpus to retrieve a document containing a contradiction relation regarding the text used as query. MisInfoBot then uses the retrieved sentences in order to formulate an answer.

Figure 3.1 illustrates the architecture for this method. Mathematically formalizing the problem for this solution, our model receives a collection of N texts, $T = T_1, T_2, ..., T_N$ as input and outputs a label $l \in 0, 1$ representing, respectively, the classification of a text as true or false, respectively. The texts classified as untrue by the model form M queries $Q = Q_1, Q_2, ..., Q_M$, such that $Q \subset T$. Having a group of Y relevant documents $RD = RD_1, RD_2, ..., RD_Y$, Q act as a query in the IR system in order to obtain a subset of K documents $SD = SD_1, SD_2, ..., SD_K$ such that $SD \subset RD$ such that each k document contradicts text m.

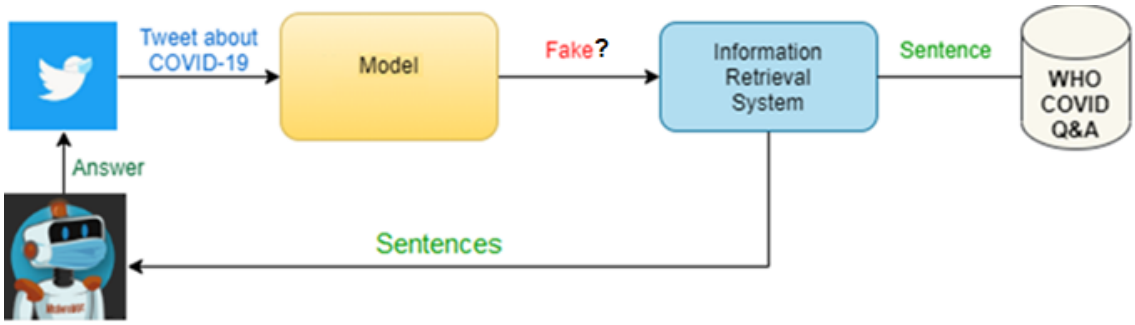


Figure 3.1: Method 1 – Architecture

3.3 Method 2 – Contradiction Retrieval

Another proposed method for MisInfoBot relies on a model that simultaneously performs a classification and a retrieval task for the received sequences. The model starts by receiving pairs of text from two different sources. One of the sources is relevant documents from a corpus. Considering the other source, for each text of this source, the model will generate as many pairs as documents exist in the corpus of reference. Hence, for a single text, many pairs will be created. The model then recognizes the relation between each one of these pairs. By calculating the probability of this relationship being a contradiction, we can obtain the most contradictory values obtained and use this value as a reference in order to output the text from the documents that contradict a text the most. Hence, the classifications predicted by the models have direct input in the retrieval as the model retrieves only contradictions.

We fine-tuned the model to classify the relationship between the tweet-sentence pairs. The model calculates the probability of each of these relations for each pair. In this method, as explained previously, the deep language model receives a tweet-sentence pair and predicts the relation holding between the tweet and the sentence. Hence, we can address this problem as a multi-class classification in which the model calculates the probability of a pair to contain a specific relation, assigning a class (0 if entailment, 1 if contradiction, 2 if neutrality) to it. However, we can also address this problem as a binary classification in which the model calculates the probability of a pair to contain a contradiction or not, assigning a class (0 if Not Contradiction, 1 if Contradiction) to it. This assignment is made according to the class with the highest score value.

In the case that the model categorizes any pair as contradictory, it will output the most contradicted sentences. Similarly to the first method, MisInfoBot will retrieve these sentences as answers to the received tweets.

Figure 3.2 illustrates the architecture of this method. Mathematically formalizing the problem for this method, the deep language model receives a collection of N texts , $T = T_1, T_2, ..., T_N$ and a group of Y relevant documents $RD = RD_1, RD_2, ..., RD_Y$ as inputs and creates $N \times Y$ pairs $P = P_{11}, P_{12}, ..., P_{NY}$, creating Y pairs for each text N. For each pair, the model outputs the probability of the document to contradict, or not, the text, respectively, i.e, $P(l|P_N Y)$. Hence, for each text T, the model outputs a a subset of K documents $SD = SD_1, SD_2, ..., SD_K$ such that $SD \subset RD$ and each document k contradicts text t.

3.4 Summary

The proposed approach aims to create a system that can detect the existence of misinformation in a text by detecting contradictions against text present in documents from a reference corpus and retrieving the contradicted documents. We propose two methods in order to implement this approach, relying on Supervised Learning classification models.

The first method objectives a separated execution of the two sub-tasks of classification and retrieval by performing them on two distinct components: a model and an IR system. In the case

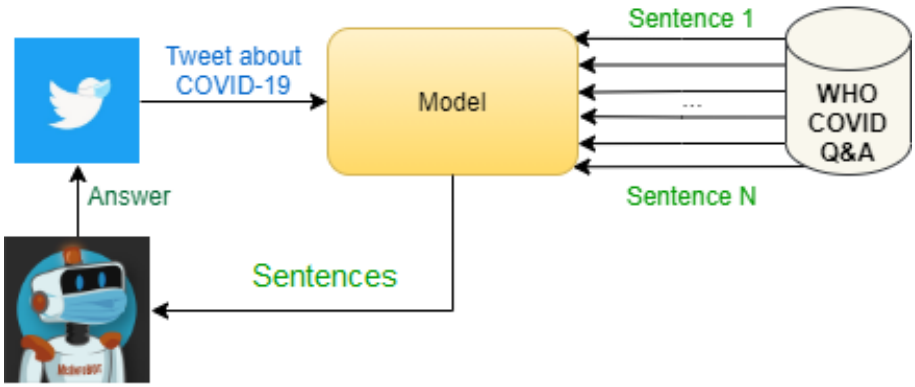


Figure 3.2: Method 2 – Architecture

that a text is classified as false by the model, the retrieval system provides relevant documents regarding the topic of that text. By classifying a tweet as false and outputting relevant sentences, this method tries to provide information that goes against what is claimed in the tweet relying on the two referred components.

The second method executes a direct retrieval of contradiction between text and text from documents from a reference corpus. By identifying the relation between them, this method retrieves the most contradictory documents for a text, simultaneously performing the two sub-tasks of our approach.

Using real tweets and a corpus of reference with relevant documents regarding the COVID-19 topic, our approach tries to tackle misinformation about this virus on Twitter by identifying contradictions against documents of the corpus.

Chapter 4

Datasets and Experimental Setup

This chapter describes the data we used and its respective gathering and pre-processing procedures and describes the executed experiments to empirically evaluate the two methods that implement the proposed approach described in the last chapter. Section 4.1 describes the main characteristics of the data we utilized and relates its preparation. Section 4.2 presents the experimental methods followed to implement our two methodology approaches. Finally, section 4.3 sums up the crucial information of these two sections.

4.1 Datasets

With the objective of exploring the methodology we proposed, we created several datasets: a **reference corpus** that contains processed sentences from the WHO's Q&A about COVID-19, a **tweets dataset** with tweets labelled as true or false and an **annotations dataset**, a dataset that we manually annotated that contains the relation (contradiction, entailment, neutrality) between the sentences and the tweets from the other two datasets. All datasets are available publicly online. [1]

4.1.1 Reference Corpus

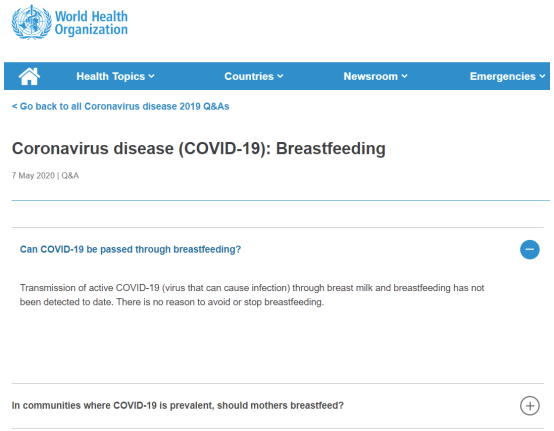
In order to create a reliable corpus that would hold reliable information regarding the virus, we gathered all the questions and answers provided by the World Health Organization on their Q&A [2] about COVID-19. As figure 4.1 exhibits, the WHO organizes this information into 47 topics and subtopics, where each subtopic is a question. We collected each answer for each subtopic, resulting in 391 answers.

Despite holding information regarding the virus, this dataset presents its limitations as WHO does not address all topics related to coronavirus that can originate misinformation. For example,

[1] https://github.com/TomasNovo/MisInfoBot-Datasets

[2] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub

(a) WHO's Coronavirus Q&A Topics



(b) WHO's Coronavirus Q&A Subtopics and answers

Figure 4.1: Organization of WHO's Coronavirus Q&A

conspiracy theories about the virus, common on tweets, are topics not addressed by the WHO. Also, as this dataset was created in February, the durability of its integrity is not guaranteed: as discoveries about the virus are made every day, the information about the virus, and thus, related information on WHO's Q&A, may become outdated. Hence, it is important to clarify that this thesis assumes that the reference corpus is correct and that dealing with the evolving nature of knowledge is out of the scope of this project.

Some of the answers to the COVID-19 topics we collected were extensive and had more than one sentence. In order to simplify them, we pre-processed all answers to obtain segmented sentences. This segmentation consisted of the separation of the sentences that formed each answer. With this division, we obtained 2412 sentences. These sentences were then filtrated by removing duplicates and re-directions to additional info, in sentences such as "find out more", "click here", "more information", "see Q&A", "find out more about", etc. This pre-processing resulted in 2265 sentences that formed our reference corpus dataset.

For the IR component of the first method, the segmented sentences were also submitted to a pre-processing so that the success of the queries could be improved. Hence, to boost the performance of our IR system, we: transformed the text of each sentence to lowercase, removed punctuation characters, unfolded apostrophes ("isn't" turned into "is not". "doesn't" into "does not", etc.) and applied word lemmatization.

4.1.2 Tweets Dataset

In order to create a consistent dataset of tweets, several datasets from distinct sources were analysed. All investigated datasets held tweets related to COVID-19 labelled *a priori* as true or false. We focused on collecting tweets directly related to health topics regarding Coronavirus (like preventing measures, vaccines, the effectiveness of surgical masks) and topics regarding conspiracy theories about the virus. Keeping in mind that this collection would directly impact the results of

our approach as we are only considering tweets regarding specific topics, we wanted to address topics relevant to our work. Hence, many tweets from the reviewed datasets weren't used for distinct reasons:

- As some labels (true or false) were already outdated due to the constant updates regarding the virus, we did not utilise them. For example, some tweets claimed that there was no vaccine for Coronavirus, and at that timestamp, it was effectively true, but vaccines were developed in this timespan.

- The content of several tweets was related to the death of famous people, irrelevant to our work and, thus, discarded.

- A large amount of tweets held political information related to the virus, like claims from politicians and countries that were not pertinent to our work and, thus, were not added to our dataset.

With this in mind, after this exclusion criteria, we collected *423* tweets from distinct sources ([7], [17], [51], [78]) in order to provide variety to our dataset. Of all the tweets, *263* (62%) are labelled as fake and *160* (38%) as true.

Twitter imposes a limitation of 280 maximum characters per tweet. Nevertheless, each tweet has its own text and, thus, its characteristics. In order to use our dataset of tweets, the respective content of these tweets was submitted to some important procedures:

Anonymization – we anonymized the content of each tweet in order to don't compromise the policies from the Twitter Developer Agreement and Policy [3], Twitter Automation Rules [4] and Twitter Rules [5]. The name of every user (started by '@') was replaced by '@user'

Cleaning – as the content of some tweets held links redirecting to other tweets (retweets) or multimedia content, each link was replaced by "http", as links do not have relevance to our work. Also, hashtags (started by '#') were separated as we processed them in order to obtain individual words.

The tweets resulting of this pre-processing formed the inputs of both of our two methods' models.

For the IR component of the first method, tweets classified as false by the classifier were also submitted to the same pre-processing of the segmented sentences. This was made to better capture the similarities between the tweets and the sentences and, thus, boost the retrieval of information.

[3] https://developer.twitter.com/en/developer-terms/agreement-and-policy.html
[4] https://help.twitter.com/en/rules-and-policies/twitter-automation
[5] https://help.twitter.com/en/rules-and-policies/twitter-rules

4.1.3 Annotations Dataset

We manually annotated the semantic relation of each tweet of our dataset of tweets with each segmented sentence of our reference corpus: *-1* if the sentence contradicted the tweet, *1* if the sentence supported the tweet and *0* if there was no relation between them. With this annotation, we obtained a matrix composed of these three numbers representing the relation of each tweet-sentence pair. Thus, by considering every cell of the matrix a pair, we obtained *958095* pairs. These pairs and respective labels were used to train the deep language model of our second method and acted as ground truth for the IR module of the first method.

Regarding the annotation process, several steps were followed. The first consisted of identifying the topic of a tweet. After this identification, all sentences regarding that topic were analysed, and the ones containing other irrelevant topics were annotated with 0. Next, the analysed sentences were annotated according to the label of a tweet: if the tweet was fake, we annotated contradiction, and if true, we annotated entailment. Neutrality was also annotated when the topic was addressed but had no relationship to the tweet or when the topic had no relation with the tweet. In some rare cases, as some tweets contained truthful and fake information simultaneously, the three relations were annotated according to the sentence. As a single user carried out the annotations, some bias may be induced in the data as something defined as contradictory or corroboratory may vary from person to person.

4.2 Experimental Setup

4.2.1 Models

The pre-training of the deep language models is an important step to achieve good results. Of the many models provided by HuggingFace Transformers, we selected those in which the pre-training revealed relevant and related to our methods as they were trained in specific domains. Thus, we selected models pre-trained on tweets, documents about Coronavirus, tweets about Coronavirus and contradictions/entailments. Table 4.1 provides descriptions about the models that we selected and then fine-tuned. For simplicity purposes, we named each model. The models that names start with M1 were used in the Classify & Retrieve method, while the models that start with M2 were utilized in the Contradiction Retrieval method.

All chosen models are variations of two major models: BERT-Base Uncased [20] (12 layers, 768 hidden layers, 12 attention heads, 110M parameters), roBERTa-Base [48] (12 layers, 768 hidden layers, 12 attention heads, 125M parameters). We also experimented a roBERTa-Large (24 layers, 1024 hidden layers, 16 attention heads, 355M parameters) model for the second method.

For the Classify & Retrieve method, the deep language models that we utilized in the classifying component predicted two classes for the received dataset of tweets for the first method: False and True. For the Contradiction Retrieval method, the deep language models that we utilized to the classifying component predicted two relations for the received dataset of pairs: Contradiction and Not Contradiction. Hence, all models of both methods perform a binary classification.

Table 4.1: Huggingface Deep Language Models

	Name	Description
digitalepidemiologylab/covid-twitter-bert-v2	M1M1	second version of a BERT-large-uncased model pre-trained on a large corpus of tweets with keywords related to coronavirus. The corpus consisted of 97M unique tweets that reached a final sample of 22.5M tweets after filtering and preprocessing. The evaluation of this model had downstream text classification tasks on Twitter from SemEval challenges [57].
gsarti/covidbert-nli	M1M2	this BERT model was trained during 6 hours on a NVIDIA Tesla P100 GPU on the CORD19 dataset of scientific articles related to COVID. This pre-trained model uses the native wordpiece vocabulary of BERT and is fine-tuned on a corpus of inference with the library of sentence-transformers to generate universal sentence embeddings by utilizing the average pooling strategy softmax loss.
lordtt13/COVID-SciBERT	M1M3	based on SciBERT: A Pretrained Language Model for Scientific Text [8] and its unsupervised pretraining was made on a vast corpus with multiple scientific publications with several domains. The evaluation of this model was made with distinct scientific domains in many NLP tasks such as dependency parsing, sequence tagging and sentence classification.
mrm8488/bioclinicalBERT-finetuned-covid-papers	M1M4	a Masked Language BERT model fine-tuned in papers about coronavirus.
cardiffnlp/twitter-roberta-base	M1M5	a roBERTa-base model trained on more than 55M tweets of TweetEval. This model is also a Masked Language model.
deepset/roberta-base-squad2-covid	M1M6	a Question Answering model trained on annotations of the CORD19 dataset. Its performance was evaluated by applying 5-fold cross-validation on the dataset, and this model results from the third fold of the cross-validation.
wikibert-base-parsinlu-entailment	M2M1	BERT model pre-trained on Persian language to recognize entailment and contradiction. [40]
ynie/roberta-large_conv_contradiction_detector_v0	M2M2	roBERTa model pre-trained on contradictions.

4.2.2 Tokenization

Tokenization is an important procedure so that data can serve as input for BERT models. As referred, we utilized several deep language models that result from variations of BERT and roBERTa models. Each model that we utilized has an associated tokenizer as they had distinct pre-trainings: despite two models having the same architecture (for example, M1M1 and M1M2 are BERT variations), the pre-training in a distinct corpus may affect the tokenizer's behaviour as distinct tokens can be learned. Hence, we tested the tokenizers associated with each deep language model that we utilized in our methods to analyze the tokenization differences.

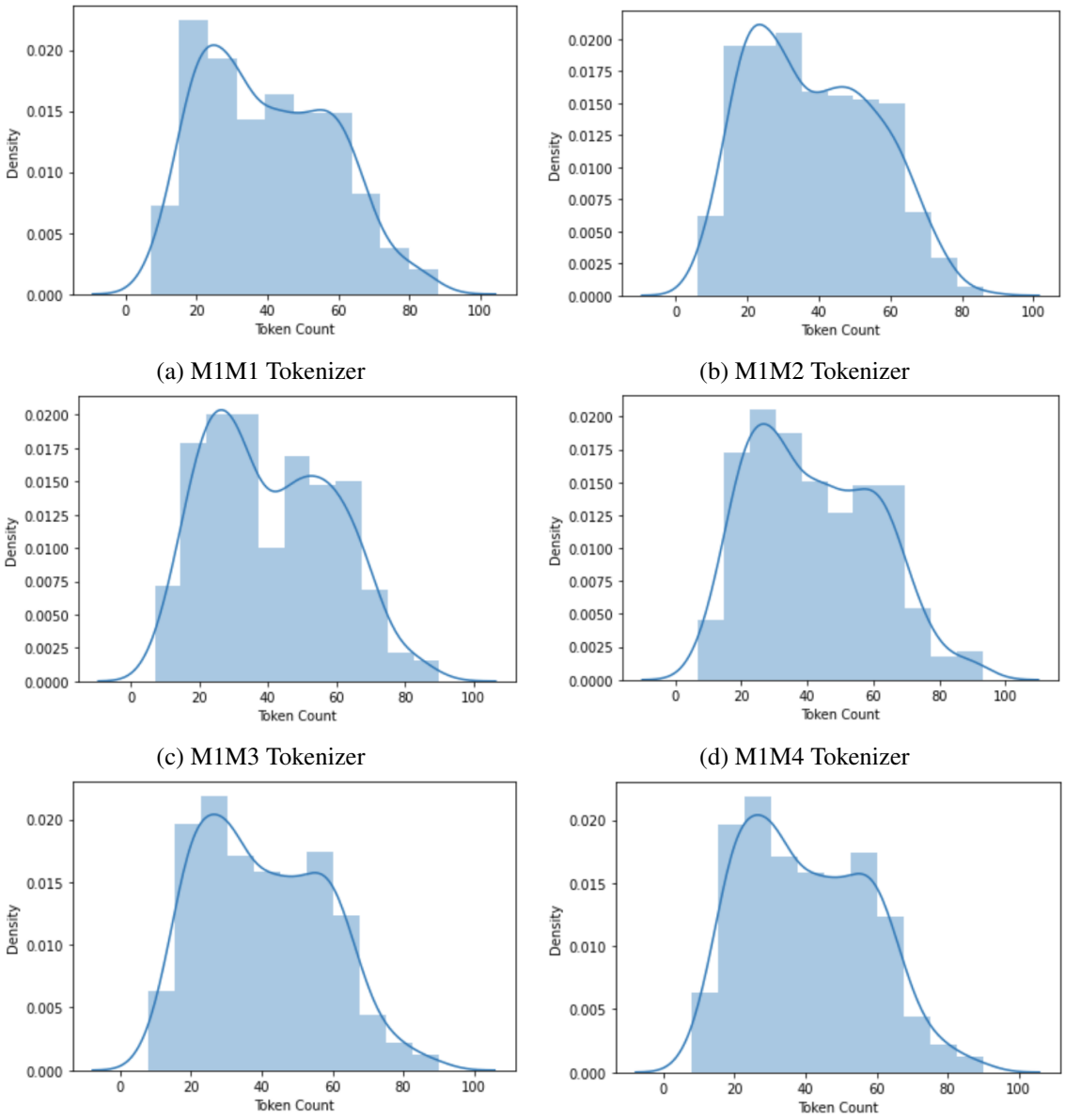
RoBERTa's tokenization differs from BERT's as it utilizes a variant named Byte-Pair Encoding (BPE). BPE is an algorithm that concatenates characters based on their frequencies. By starting with two-byte characters and considering n-gram pairs of tokens and their respective frequencies, several new longer tokens are added to the vocabulary of the model. Hence, BERT's tokenizer preferably merges two succeeding tokens with two consecutive "##" while roBERTa's uses a specific Unicode character, 'Ġ'.

With this in mind, as the two methods we developed use tweets and tweet-sentence pairs, respectively, we submitted them to tokenization utilizing each model's tokenizer and respective configurations. This allowed us to choose a maximum sequence length for this tokenization, an important hyperparameter. Figures 4.2 and 4.3 present histograms that contain the number of tokens per tweet and per tweet-sentence pair for the first and second methods, respectively. It is possible to visualize that for the first method, the maximum number of tokens is inferior to 128. Thus, this was the chosen value for the maximum sequence length for this method. For the second method, we chose to set this hyperparameter at 256. Despite the loss of some information, as few pairs ($< 1\%$) have more than this value of tokens, this allowed us to utilize larger batch sizes, an important hyperparameter related to the number of samples, and save some time on expensive computational efforts.

Tokenizers were also utilized in the creation of the tweet-sentence pairs for the **Contradiction Retrieval** method. Hence, each pair formed by a tweet and a sentence is unified through a separator token from the tokenizer of the deep language models for this method. As the two utilized models for this method are variations of BERT and roBERTa models, we had to generate different pairs with different separator tokens for the experiments with the used models as they differ in this tokenizing configuration.

4.2.3 Information Retrieval

One crucial objective of our work is to retrieve information regarding a topic of a tweet. As we developed two distinct methods to address misinformation in text, their characteristics in retrieving information are distinct.



(a) M1M1 Tokenizer

(b) M1M2 Tokenizer

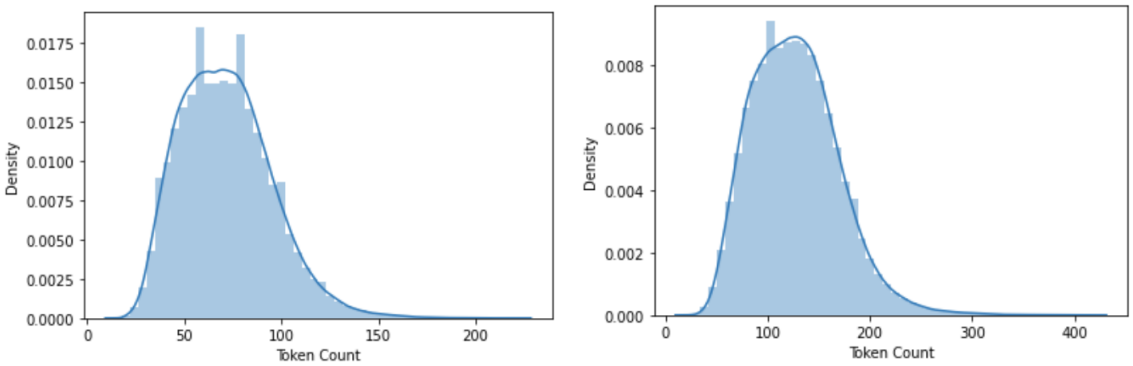
(c) M1M3 Tokenizer

(d) M1M4 Tokenizer

(e) M1M5 Tokenizer

(f) M1M6 Tokenizer

Figure 4.2: Count of tokens per tweet of tokenizers of Classify & Retrieve method



(a) M2M1 Tokenizer (b) M2M2 Tokenizer

Figure 4.3: Count of tokens per pair of tokenizers of Contradiction Retrieval method

4.2.3.1 Classify & Retrieve

For the first method, the Information Retrieval System is triggered when a tweet is classified as fake. This system relies on the calculation of similarity between the tweet and the sentences from our reference corpus. Hence, we tested two different representations for the IR component: one based on TF-IDF and the other on BERT Embeddings. We relied on the calculus of the **Cosine Similarity** for both representations.

We relied on deep language models in order to convert the tweets and the sentences from our reference corpus to their respective embeddings. Hence, we utilized a BERT-Base Uncased model and all the deep language models that we utilized in the classifying component of the Classify & Retrieve method in order to effectuate this conversion.

4.2.3.2 Contradiction Retrieval

On the second method, the deep language model receives tweet-sentence pairs as input and predicts the probability of each pair to hold a contradiction. As for a tweet exists as many pairs as many sentences in our corpus of reference, several pairs, and thus several sentences can be classified as contradictory regarding the tweet. Hence, the model uses the predicted values to rank and retrieve the sentences classified as holding a contradiction against the tweet.

4.2.4 Training, Evaluation and Testing - Performance estimation

As explained in Chapter 3, both methods we developed to implement our approach rely on the fine-tuning of pre-trained deep language models to perform a sentence classification task. Regarding the fine-tuning, Table 4.2 illustrates the chosen neural network hyperparameters utilized in the two methods. The evaluation of the training of the used models is made at the end of each **epoch**. The chosen values of batch sizes were selected so that they would not computationally compromise our experiments. The chosen optimizer for the fine-tuning was AdamW [41], an Stochastic Gradient Descent type (keeps a singular learning rate for every weight update and preserves it during training) optimization algorithm that adapts the learning rate as the training occurs.

Table 4.2: Fine-Tuning Hyperparameters

Hyperparameters	Classify & Retrieve	Contradiction Retrieval
Training Epochs Number	15	15
Train Batch Size	8	8
Validation Batch Size	16	16
Maximum Sequence Length	128	256
Learning Rate	5e-5	5e-5
Warm Up Steps	500	500
Weight Decay	0.01	0.01

We divided the tweets dataset into distinct datasets to try different experiments: **train**, **evaluation** and **test**. Nevertheless, the proportion of true and false tweets is considered in this split as the sampling was stratified. Table 4.3 illustrates the three subsets that resulted from the split of the dataset of tweets that we used in the Classify & Retrieve method. It is possible to observe that every subset has more false tweets than true ones, representing more than 60% of the total tweets of each subset. Hence, every subset is unbalanced as they have more tweets from the False class.

For the second method, we created tweet-sentence pairs by joining every sentence of our reference corpus with every tweet of the datasets created for the first method. The pairs in each set match exactly the tweets in the corresponding set of the split of the Classify & Retrieve method. Analysing this table, we can notice that the neutrality relation exists in incomparable dimensions regarding the contradiction and entailment relations, as we are talking about 99% versus less than 1%.

Table 4.3: Classify & Retrieve – Tweets Dataset Split

Dataset	Total Samples	False Samples	True Samples
Train	253 (60%)	153 (60%)	100 (40%)
Validation	85 (20%)	58 (68%)	27 (32%)
Test	85 (20%)	52 (61%)	33 (39%)

Table 4.4: Contradiction Retrieval – Pairs Dataset Split

Dataset	Total Samples	Entailment Samples	Contradiction Samples	Neutrality Samples
Train	573045 (60%)	1387 (0.25%)	597 (0.10%)	571061 (99.65%)
Validation	192525 (20%)	345 (0.17%)	258 (0.13%)	191922 (99.70%)
Test	192525 (20%)	447 (0.23%)	159 (0.07%)	191919 (99.70%)
Undersampled Train	1791	597 (33%)	597 (33%)	597 (33%)

To address this unbalance, we created an undersampled subset of training in order to avoid an unbalanced classification tending for the class with the highest number of samples. The number of samples of each class for this undersampled subset is the number of contradiction samples of the original dataset of train, validation or test. As the normal train dataset had 597 contradiction

samples, the undersampled dataset for train has 597 samples of each class. The undersampling is made at the datasets of pairs and not at the datasets of tweets because as a tweet originates as many pairs as many sentences exist in the reference corpus, the resulting dataset would still be unbalanced. Hence, doing it at pair level allows a possibility to try to generate an equilibrium in the training of the classifiers in order to identify contradictions.

Table 4.4 presents the subsets that resulted from the aggregation of subsets of tweets from the first method and the sentences from our corpus of reference and undersampling.

4.2.5 Experiments

After the split of the datasets into training, testing and evaluation, distinct experiments were done in order to answer to the following research questions:

RQ1 – *Can we use pre-trained deep language models to accurately predict false tweets?*

RQ2 – *Can we improve the accuracy of false tweet prediction by fine-tuning pre-trained deep language models?*

RQ3 – *Can we use a simple, distance-based approach to retrieve documents from the reference corpus that are relevant to a fake tweet?*

RQ4 – *Can we use pre-trained deep language models to accurately predict contradictions between tweets and documents from a reference corpus?*

RQ5 – *Can we improve the accuracy of prediction of contradictions between tweets and documents from the reference corpus by fine-tuning pre-trained deep language models?*

Regarding ***RQ1***, we submitted every deep language model to a **Zero-Shot Learning** experiment with the evaluation dataset of tweets in order to evaluate their false tweets recognition ability based on their pre-training.

To answer ***RQ2***, we **fine-tuned** every model to classify tweets as false by training them on GPUs with the tweets training dataset and with the hyperparameters established for this method. The evaluation and test subsets of tweets were used to evaluate the model during training and to obtain results and metrics, respectively.

For ***RQ3***, we used the test dataset of tweets so we could obtain performance metrics for this system, simulating the case that the classifier would have 100% of accuracy. We tried different ways to identify the **cosine similarity** between our tweets and our sentences: through embeddings produced by the deep language models and through the use of text directly through TF-IDF.

Regarding ***RQ4***, we submitted the deep language models selected to detect contradiction to a **Zero-Shot Learning** experiment with the evaluation dataset of tweet-sentence pairs.

To answer ***RQ5***, we **fine-tuned** every model to classify sentences by training them on GPUs with the undersampled training dataset of pairs and hyperparameters. The evaluation and test subsets of tweets were used to evaluate the model during training and to obtain results and metrics, respectively.

Table 4.5: Versions of used software

Software	Version
Huggingface Transformers	0.0.8
Matplotlib	3.2.2
Numpy	1.20.0
Pandas	1.2.4
PyTorch	1.7.1
Scikit-learn	0.24.2
Seaborn	0.11.1
SpaCy	3.0
Tweepy	3.10.0
Ubuntu	20.04.2 LTS

4.2.6 MisInfoBot - Twitter Setup

To implement our Twitter bot, we took advantage of the functionalities furnished by Twitter's Application Programming Interface (API). [6] This API allows developers/researchers to perform actions on Twitter without accessing its User Interface (UI). These actions are the many interactions that Twitter has as features, allowing access to information (friends, followers, tweets, retweets, retweeters, likes, likers, etc.) of particular accounts and information present on tweets (text, hashtags, keywords, images, videos, etc.) through the execution of specific methods present in the API's documentation. [7] With this in mind, we created an account for our bot in order to utilize this API for our research: to access content present in tweets, as all the datasets of tweets that we investigated only contained the ID of each tweet that formed it for anonymization purposes; to create a handler for our bot when invoked (i.e., when a tweet holds *@MisInfoBotCOVID*), with the intention of retrieving relevant information as an answer to a tweet, informing if it holds (or not) misinformation.

4.2.7 Environment and Frameworks

After researching distinct software to implement our approaches, we selected several tools. To execute the proposed experiments, we used Google Colaboratory. This platform allows performing runtimes of 12 hours of Python 3.7.10 with free access to GPU (Graphics Processing Unit) hardware acceleration on an Ubuntu operating system. Table 4.5 provides the utilized software and its respective version for reproducibility.

We used Hugging Face Transformers [8], a state-of-the-art library for NLP based on PyTorch/TensorFlow that has several implementations of distinct deep language models in a large repository. This library allows the use of knowledge learnt by several pre-trained models. Hence, we used some models existent with different pre-training, as this library also allows us to avoid expensive

[6] https://developer.twitter.com/en
[7] https://developer.twitter.com/en/docs
[8] https://github.com/huggingface/transformers

pre-training efforts. Each utilized model was fine-tuned to classify sentences/pairs of sentences. All used models were implemented with PyTorch. We also took advantage of the tokenizers provided by this library in order to encode our datasets.

4.3 Summary

In this chapter, we described in detail the datasets we used to empirically evaluate our approach and the respective gathering and pre-processing procedures.

By providing the environment and frameworks that we utilized, we produce opportunities to reproduce our work. The choice of the models we utilized was justified, and descriptions of the same models were provided. The associated tokenizers of each model and respective tokenization distinctions in this process were also presented. Explanations about how the retrieval of relevant information is also provided in this chapter and how we set up our bot by accessing Twitter's API.

Regarding the distinct executed experiments, we presented how the division of our datasets was made and described the conduction of the experiments itself. For these experiments, we formulated five Research Questions: ***RQ1***, ***RQ2***, ***RQ3***, ***RQ4***, and ***RQ5***. ***RQ1*** addresses the use of pre-trained deep language models to classify tweets as false and ***RQ2*** the fine-tuning of these models for the same purpose. ***RQ3*** inquires about the use of TF-IDF or pre-trained representations to capture similarities between tweets and sentences and retrieve information. ***RQ4*** addresses the use of pre-trained deep language models to classify pairs of text as contradictory and ***RQ5*** the fine-tuning of these models for the same purpose.

Chapter 5

Results & Analysis

This chapter presents and discusses the results of the empirical validation of the proposed methods.

To fight misinformation, we developed two distinct methods that perform two sub-tasks: Supervised Learning Classification and Information Retrieval. The Classify & Retrieve method performs these tasks individually in two different components, while the Contradiction Retrieval method executes both classification and retrieval simultaneously. With this in mind, we decided to evaluate each component of the first method individually. For the second method, as it is composed of only a classification model, the analysis of the classification and IR results is made together. The model acts as both classifier and retriever.

In Section 2.1.3, we described the main metrics used to evaluate the performance of a Supervised Learning Classifier. Hence, in the following sections, the classification results obtained are analysed and discussed according to those metrics. On the Classify & Retrieve method, we present results obtained as a binary classification problem in which the model calculates the probability of a tweet to be true or false. On the Contradiction Retrieval method, we face the problem as a multi-class classification. The model calculates the probability of a pair to contain an entailment, contradiction or neutrality relation. However, the results are presented as a binary classification of two classes: Contradiction and Not Contradiction, as its more aligned with the problem we are addressing. Hence, we consider Not Contradiction any relation of entailment or neutrality.

Nevertheless, the outcome of our approach is the retrieval of sentences. Thus, we will rely on the metrics presented in Section 2.2.2.3 in order to evaluate the outputs of the two methods that we implemented.

5.1 *RQ1 – Can we use pre-trained deep language models to accurately predict false tweets?*

In order to examine the classifying ability of the used models without any training, we submitted these models to a Zero-Shot Learning experience. The results of Zero-Shot Learning are in

Table 5.1.

In this experiment, the models classified the tweets based on their pre-training. All accuracy values of the six models hold between the interval 43%-65%. Model **M1M6** had the best performance in distinguishing the false class as it had the highest value of Precision and F1 Score and the highest Specificity value. Model **M1M2** could not distinguish the existence of truthful tweets. Regarding the rest of the models, **M1M4** outperformed models **M1M5**, **M1M3** and **M1M1**. As seen, the pre-train of the models in a corpus directly influence in the classification of a tweet as true or fake.

Table 5.1: Zero-Shot Learning results on fake post classification

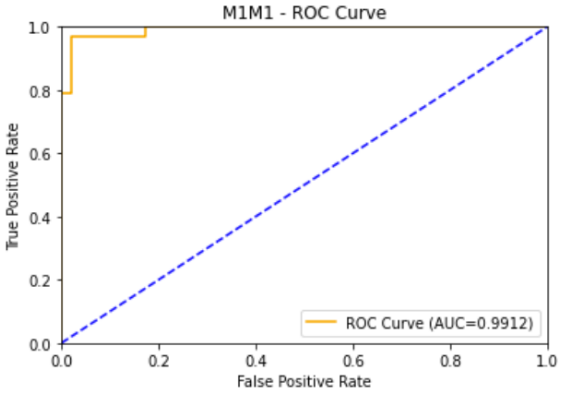
	Accuracy	Precision	Sensitivity	F1	Specificity
M1M1	0.4353	0.6190	0.4483	0.5200	0.2558
M1M2	0.6000	0.6538	0.8793	0.7500	0.0000
M1M3	0.4471	0.6410	0.4310	0.5155	0.2826
M1M4	0.5412	0.6939	0.5862	0.6455	0.3333
M1M5	0.6471	0.2000	0.0370	0.0625	0.6750
M1M6	0.6471	0.6944	0.8621	0.7692	0.3846

5.2 *RQ2 – Can we improve the accuracy of false tweet prediction by fine-tuning pre-trained deep language models?*

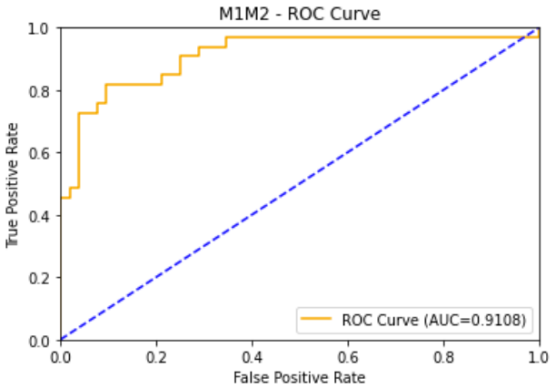
After the Zero-Shot Learning experiment, we fine-tuned the pre-trained models on the problem of which posts are false by training them with the train dataset of tweets with the hyperparameters for the Classify & Retrieve method present in Table 4.2. The principal metrics of the performance of every model in this experiment and their ROC Curves are presented in Table 5.2 and Figure 5.1, respectively. It is possible to see that the performance of every model improved with the training. All models had values of AUC above 80%, proving that the fine-tuning of each one was effective in order to classify tweets as true or false. As model **M1M1** was pre-trained on tweets that address coronavirus, it is the most familiarized with the words of our dataset of tweets and reveals itself as the most capable of distinguishing false tweets from the truthful ones and is the one with the best performance.

5.3 *RQ3 – Can we use a simple, distance-based approach to retrieve documents from the reference corpus that are relevant to a fake tweet?*

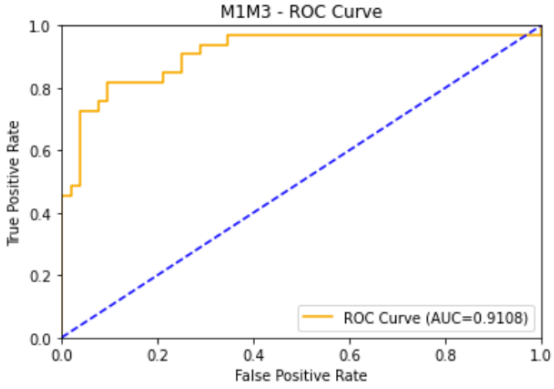
In this experiment, we used the 52 false tweets of the test dataset, simulating the scenario that the fine-tuned deep language model would have 100% of accuracy.



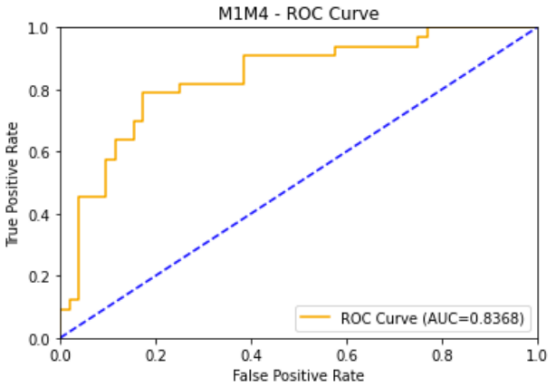
(a) M1M1 – ROC Curve



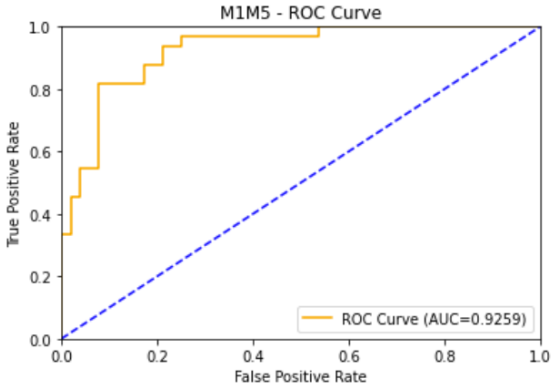
(b) M1M2 – ROC Curve



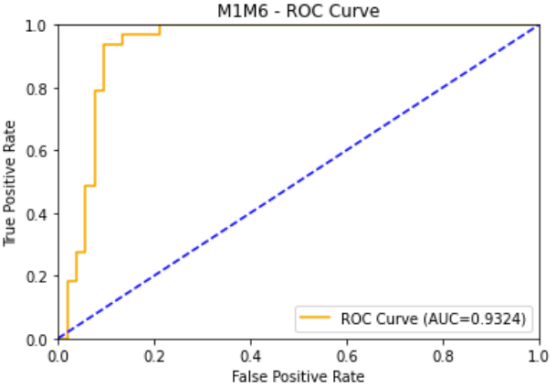
(c) M1M3 – ROC Curve



(d) M1M4 – ROC Curve



(e) M1M5 – ROC Curve



(f) M1M6 – ROC Curve

Figure 5.1: Classify & Retrieve – ROC Curves

Table 5.2: Fine-tuned models results on fake post classification

	Accuracy	Precision	Sensitivity	F1	Specificity
M1M1	0.9294	0.9792	0.9038	0.9400	0.8649
M1M2	0.8000	0.8723	0.7885	0.8283	0.7105
M1M3	0.8706	0.9362	0.8462	0.8889	0.7895
M1M4	0.8118	0.8600	0.8269	0.8431	0.7429
M1M5	0.8588	0.8846	0.8846	0.8846	0.8182
M1M6	0.8353	0.8519	0.8846	0.8679	0.7676

Both implemented systems (one based on TF-IDF and the other in word embeddings) revealed to have really bad performances in the retrieval of useful information. We tested the two systems with $K = 5$, $K = 10$ and $K = 50$ and for the three values, the metrics are unsatisfactory as the highest value of Mean Average Precision was 2.69% at $K = 5$ for TF-IDF. This system presented better results than the one based on the embeddings produced by all the analysed deep language models, the six utilized in the first component of this method, and also a BERT Base model, as Table 5.3 demonstrates. This happens probably due to the linguistic differences between the tweets and the sentences. The text of the sentences is much more informative, formal and extensive than text on tweets. The embeddings of words of the tweets and the sentences produced by the deep language models are very distinct despite sometimes addressing the same topic.

Table 5.3: Mean Average Precision results of IR for RQ3

	K = 5	K = 10	K = 50
TF-IDF	0.0269	0.0154	0.0046
M1M1	0.0153	0.0096	0.0031
M1M2	0.0077	0.0659	0.0034
M1M3	0.0154	0.0077	0.0035
M1M4	0.0077	0.0077	0.0038
M1M5	0.0115	0.0058	0.0031
M1M6	-	0.0019	0.0015
BERT Base	0.0077	0.0096	0.0031

5.4 *RQ4 – Can we use pre-trained deep language models to accurately predict contradictions between tweets and documents from a reference corpus?*

As explained in the experiments Section regarding this method, we submitted the two used deep language models (**M2M1** and **M2M2**) to a Zero-Shot Learning experience with the evaluation dataset of pairs (192525 pairs) to evaluate the utilized models' classification aptitude to detect contradictions.

Tables 5.4 and 5.5 contain the confusion matrices of this experiment for models **M2M1** and **M2M1**, respectively. Table 5.6 exhibits the principal classification metrics of the two deep language models, in order to verify their ability to detect contradictions without fine-tuning. Analysing the Tables, we can verify that model **M2M2** had more difficulties to identify contradictions than **M2M1** if both models relied only on their pre-trainings. Model **M2M1** also classified much more samples as contradictory than model **M2M2** (60116 samples vs 5056 samples), presenting a higher Precision value but a lower Sensitivity as consequence when comparing to **M2M2**.

5.5 *RQ5 – Can we improve the accuracy of contradictions prediction between tweets and documents from the reference corpus by fine-tuning pre-trained deep language models?*

After obtaining the results for the Zero-Shot Learning experiment, as presented in the previous Section, we fine-tuned the models with the hyperparameters for the Contradiction Retrieval method present in Table 4.2.

Unfortunately, it was impossible to obtain results for the model M2M2 for computational reasons, as it consists of a roBERTa-Large model with 355M parameters and even with minimal batch size, the training of this model was not possible to achieve as it was too heavy for our environmental resources.

We fine-tuned the deep language model M2M1 so it could classify sequences by training it on our dataset of tweet-sentence pairs. As this dataset of pairs has a large number of samples and is heavily unbalanced, we undersampled this dataset, as explained previously.

Tables 5.8 and 5.7 illustrate the performance of this model through the obtained confusion matrix and metrics, respectively. The ROC Curve of this performance is presented in Figure 5.2. Analysing these results, we can conclude that the AUC value is under 0.5, meaning that the predictions made by the model have the same performance as predicting the class of a sample with a random guess. The values of Precision and Sensitivity denounce this behaviour of the model, despite its training.

Regarding the results of the Information Retrieval of this method, of the 3610 contradictions predicted by the model, only 112 were actual contradictions, from the total of 159 of the test dataset. This means that the Precision value of the outputted contradictory sentences by the model

Table 5.4: Contradiction Retrieval – Zero-Shot Learning M2M1 results on contradiction detection

		Predicted Class		
		Not Contradiction	Contradiction	Total
Actual Class	Not Contradiction	**132222**	**60045**	**192267**
	Contradiction	**187**	**71**	**258**
	Total	**132409**	**60116**	**192525**

Table 5.5: Zero-Shot Learning M2M2 results on contradiction detection

		Predicted Class		
		Not Contradiction	Contradiction	Total
Actual Class	Not Contradiction	**187228**	**5039**	**192267**
	Contradiction	**241**	**17**	**258**
	Total	**187469**	**5056**	**192525**

Table 5.6: Zero-Shot Learning results on contradiction detection

	Accuracy	Precision	Sensitivity	F1 Score	Specificity
M2M1	0.6871	0.2752	0.0012	0.0024	0.9986
M2M2	0.9726	0.0659	0.0034	0.0064	0.9987

Table 5.7: M2M1 Fine-Tuning metrics on contradiction detection

	Accuracy	Precision	Sensitivity	F1 Score	Specificity
M2M1	0.9815	0.7044	0.0309	0.0593	0.9998

Table 5.8: M2M1 Fine-Tuning Confusion Matrix on contradiction detection

		Predicted Class		
		Not Contradiction	Contradiction	Total
Actual Class	Not Contradiction	**188858**	**3508**	**192366**
	Contradiction	**47**	**112**	**159**
	Total	**188905**	**3620**	**192525**

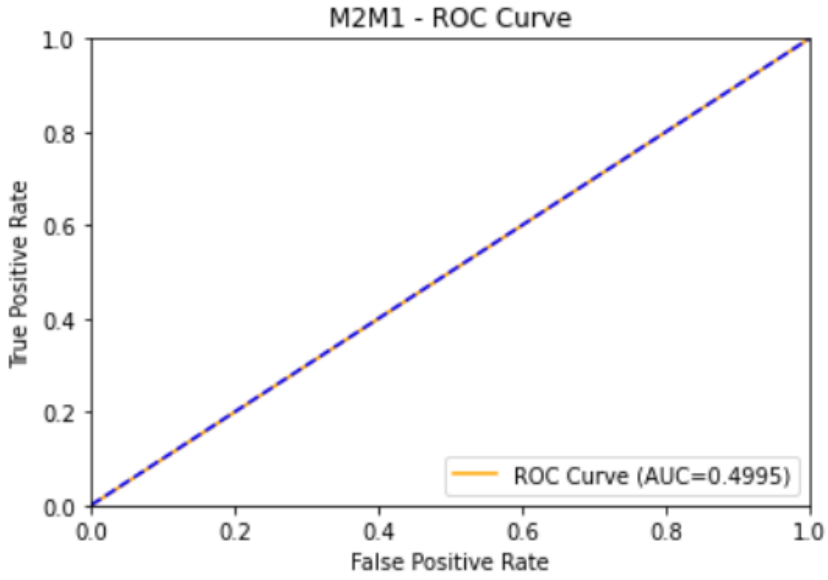


Figure 5.2: Contradiction Retrieval – M2M1 ROC Curve

assumes a value of 3.09%. As it predicted many of the real tested contradictions, it has a Recall of 70.44%. With these two values, we obtain the F1 Score, which for this performance is 5.92%. This model predicted contradictions for the 85 tweets that jointly with the sentences from our corpus of reference form our pairs test dataset Table 5.3 contains the obtained Mean Average Precision for this method.

Table 5.9: Mean Average Precision on IR of contradictions

	K = 5	K = 10	K = 50
M2M1	0.0118	0.0082	0.0031

5.6 Analysis

We obtained results of implementing a novel approach to fight misinformation regarding COVID-19 by combining a fine-tuned pre-trained deep language model with information retrieval. With this in mind, all obtained results assume relevance as misinformation is a phenomenon that needs to be fought.

The results obtained from the first method we developed that implements our approach had good results in the first component, the classifier (***RQ1***, ***RQ2***), and bad results in retrieving information. Every deep language model fine-tuned to differentiate false tweets from truthful tweets had good predictive performance, with every AUC above 0.80 for the test dataset, after the training, which means the models we used learned how to distinguish the two classes of tweets. However, our approach does not consider several factors as we do not consider irrelevant tweets, and the dataset we used is unbalanced with more false examples than real ones. Hence, we are only considering a hypothetical scenario as we are not addressing all the topics regarding Coronavirus or other topics on Twitter. The results we obtained in this particular scenario are just the first steps in a difficult fight against misinformation. Our dataset of tweets holds much more samples of the class we want to predict, which may also impact the accuracy of the results. As the annotations of the relation between the tweets and the sentences were done manually, they may have some bias induced, which may affect the performance of the models. Nevertheless, the classification results are far better than the information retrieval ones obtained.

Neither with TF-IDF nor with word embeddings (***RQ3***) the Information Retrieval System could present satisfactory results: all the obtained metrics are poor and insufficient. This may lead to the hypothesis that the language of the tweets from our dataset is clearly distinct from the language used in the sentences from our reference corpus: as the sentences are much more formal and informative, our information retrieval systems are incompetent at capturing the similarities between them.

Regarding the second method, as the unbalancing of our dataset of tweet-sentence pairs could and probably would influence the prediction ability of the deep language model, we performed an undersampling. Neither the Zero-Shot Learning nor the fine-tuning of the model for sequence

classification were effective (***RQ4***, ***RQ5***). Its AUC value is 0.5, accusing the lack of capability of the model to detect contradictions. Hence, this method needs to be improved with future work by increasing the number of samples of each relation of our training dataset. This increase may be achieved by adding tweets to our dataset of sentences and by performing the annotation processed we executed and described in Section 4.1.3. In this method, the information retrieval component depends directly of the classification of a pair as contradictory, as the model only retrieves pairs recognized as contradictory, relying in its classification capability. This imposes limitation to the results we obtained regarding the retrieval of information for this method. The metrics we obtained for the retrieval of information of this method also denounce its lack of capability in this component.

Both methods that implement our approach, despite interesting and relevant, are not enough as retrieving information is not effective. The classification of a tweet as true or false outperforms the detection of contradictions between the tweets from our dataset and the sentences from our corpus. The contrast of semantic and lexical nature between the tweets and the sentences is one of the main causes of this failure and the relatively low number of samples utilized in the experiences. The fact that some topics of the tweets from our dataset aren't addressed in our reference corpus also influences the obtained results.

Chapter 6

Conclusions

Misinformation is a phenomenon that takes place in Social Media that consists of the dissemination of false information. It is a dangerous phenomenon that takes advantage of the number of users of this type of platforms and the freedom of expression provided by them that can cause consequences, especially in the current pandemic context of COVID-19. Thus, this phenomenon needs to be tackled. Hence, the objective of this dissertation was to propose and develop an approach to identify misinformation present on Twitter through an automated bot for this platform and retrieve truthful information to unmask the presence of this phenomenon and educate the users of this platform.

With that intention, we developed an approach based on Supervised Learning Classification and Information Retrieval that aims to retrieve truthful information present in sentences for tweets suspicious of containing misinformation. For that purpose, we relied on deep language models pre-trained in distinct specific corpus, taking advantage of the knowledge learned. Hence, we implemented our approach with two different methods. The first method combines a deep language model fine-tuned for sequence classification that categorizes a tweet as fake or not and uses the fake tweet as a query in an Information Retrieval System with the objective of retrieving reliable information from a reference corpus. The second method also uses a deep language model fine-tuned for sequence classification, but it acts as a classifier and a retriever. This model identifies contradictions between a tweet and the documents of a reference corpus and retrieves the documents that contradict the tweet the most.

We collected and processed a dataset of real tweets labelled as True or False and gathered and processed sentences from the WHO Q&A about COVID-19 to create a corpus of reference. We then manually annotated the relation (entailment, contradiction, neutrality) between each tweet and each sentence. The datasets we utilized present some limitations, as we have an unbalanced dataset of tweets that only address specific topics regarding COVID-19, not covering the vast amount of topics that exist regarding this virus on Twitter. The manual annotations of the relations

between the tweets and the sentences may also have some bias induced, as a single user carried them.

After empirically evaluating the developed methods with the described experiments and datasets, we concluded that our approach is relevant and has potential but lacks efficiency in certain aspects.

Regarding the **Classify & Retrieve** method, the classification metrics are very satisfactory. Still, the retrieval of useful and accurate information was the heel of Achilles of this method, as its results were bad for the systems implemented for the purpose. Notorious distinctions between the text characteristics of the tweets and played a significant role in the bad results of the information retrieval role of this method.

For the **Contradiction Retrieval** method, the fine-tuned model **M2M1** had interesting metrics as it was capable of detecting the presence of a big part of the contradictions of the used test dataset. However, this model also detected contradictions where they did not occur, which leads to the conclusion that its performance can be improved as their class recognition is still arbitrary. Thus, improving the classification performance of this method also means improving the quality of its retrieved information, which was unsatisfactory.

6.1 Answer to Hypothesis

In Section 1.2 we raised the following hypothesis in order to make clearer the objectives of our work.

Hypothesis – *Is it possible to use NLP methods to recognize misinformation and provide reliable information concerning that topic ?*

After our work, we can conclude that NLP techniques can be effective to address misinformation on Twitter and other Social Media platforms. By processing human language present in text, several approaches can be developed with the objective of tackling this problem. In the experiments made for the **Classify & Retrieve** method, we obtained good results for the classification of a tweet as true or false, which means that exist patterns/keywords in the writing of misinformation that can be identified. The classification of text as true or false is a technique that reveals the potential to address misinformation. In **Contradiction Retrieval** method, the recognition of contradictions between tweets and documents of a corpus of reference presented worse results than the classification of tweets as true or false. With this in mind, with some work in order to improve the classifier model of the Classify & Retrieve method, the recognition of misinformation regarding COVID-19 may be possible in Social Media platforms.

However, the metrics we obtained from the experiments involving retrieving information expose the difficulties of capturing the similarity between the used tweets and sentences. One hypothesis for this lack of efficiency in this retrieval is the language difference between the tweets and the sentences we used, as tweets are much more informal and short than the informative and formal sentences. Hence, the retrieval of information is a task that needs to be perfected with, for

example, other approaches to catch similarities different from the ones we tried in order to develop a system able to both recognize information and provide useful information regarding a topic.

6.2 Contributions

Highlighting the main contribution of this dissertation, we have:

- The creation of a Twitter bot that examines the presence of COVID-19 misinformation in a tweet and retrieves truthful information
- The application of the proposed approach through the use of real tweets and relevant documents from WHO's Coronavirus Q&A with the respective pre-processing
- The testing and collecting of results of distinct pre-trained deep language BERT models fine-tuned to classify sequences of a specific dataset, as well as the analysis of the Cosine similarity between embeddings generated by these models
- The conceiving of an annotated publicly available dataset of semantic relations (entailment, contradiction, neutrality) between tweets and a reference corpus that are also public

6.3 Future Work

Despite the interesting results obtained from the novel approach that we developed to tackle misinformation through Supervised Learning Classification and Information Retrieval through a Twitter bot, the outcomes of our work are just a first step in this serious fight against false information. Different approaches of the one that we proposed may be more or less effective, but contributing to this cause will only be a step further to tackle this phenomenon. Regarding our approach, many characteristics may be addressed in order to improve it. The ones we consider the most relevant are the following:

- **Tweets Dataset –** as explained in the description of the followed experimental setup for this dissertation, we filtered the tweets from the analysed datasets to made a custom dataset of tweets. This dataset was made with the objective of inducing useful knowledge regarding specific topics (true and false) of Coronavirus to our classifier. However, we create a hypothetical scenario with this choice of tweets, as Twitter holds countless tweets regarding the virus and topics that we have not introduced to our models. Hence, exploring the amplification of the themes addressed by increasing the number of tweets used may be a relevant step to improve the results.
- **Reference Corpus –** similarly to the point above, the addition of relevant documents from other sources would also be relevant as the reference corpus would cover more subjects. The use of sources with different types of language would also impact the performance of

the models. Also, other types of pre-processing, like a different segmentation of the corpus, could present an improvement of the results of the IR task of the Classify & Retrieve method.

- **Annotations Dataset –** as explained, the relation between every sentence of our corpus of reference and the tweets from our dataset was manually annotated, and some bias may be induced. Therefore, increasing the number of annotations and reducing the bias could lead to better results.

- **Models and Hyperparameters –** the description of each pre-trained deep language model that we utilized in both methods and why we opted for their use was clarified in Section 4.2.1. The use of different models with pre-trainings distinct from the ones we utilized could present interesting results, as well as distinct fine-tuning hyperparameters. In addition, a distinct undersampling technique could also present interesting outcomes for our second method.

- **Information Retrieval –** as illustrated in the experimental setup, we explored the calculus of the Cosine Similarity between tweets and the documents from our corpus of reference and between the embeddings of these elements in order to retrieve useful documents. The calculation of other types of similarities/distances could improve the results we obtained. Furthermore, other types of IR systems could also improve the results we obtained regarding this component.

- **Bot Interactions –** MisInfoBot only answers when invoked, i.e, when a tweet contains '@MisInfoBotCOVID'. Exploring other types of interactions and listeners will allow exploring more opportunities to address misinformation.

- **Open Source –** as misinformation exists about different topics and in many types and the fight against it is common, the possibility of open this project to interested researchers may bring relevant outcomes.

Appendix A

Results

A.1 Classify & Retrieve method – Results Zero-Shot Learning

This Section includes the confusion matrices obtained from the Zero-Shot Learning experiment of the six deep language models M1M1, M1M2, M1M3, M1M4, M1M5, M1M6 for the first implemented method.

Table A.1: M1M1 - Zero-Shot Learning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**26**	**32**	**58**
	True	**16**	**11**	**27**
	Total	**42**	**43**	**85**

Table A.2: M1M2 - Zero-Shot Learning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**51**	**7**	**58**
	True	**27**	**0**	**27**
	Total	**78**	**7**	**85**

Table A.3: M1M3 - Zero-Shot Learning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**25**	**33**	**58**
	True	**14**	**13**	**27**
	Total	**39**	**46**	**85**

Table A.4: M1M4 - Zero-Shot Learning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**34**	**24**	**58**
	True	**15**	**12**	**27**
	Total	**49**	**36**	**85**

Table A.5: M1M5 - Zero-Shot Learning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**54**	**4**	**58**
	True	**26**	**1**	**27**
	Total	**80**	**5**	**85**

Table A.6: M1M6 - Zero-Shot Learning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**50**	**8**	**58**
	True	**22**	**5**	**27**
	Total	**72**	**13**	**85**

A.2 Classify & Retrieve method – Fine-Tuning Results

This Section includes the confusion matrices obtained from the fine-tuning of the six deep language models M1M1, M1M2, M1M3, M1M4, M1M5, M1M6 used in the first implemented method.

Table A.7: M1M1 - Fine-Tuning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**47**	**5**	**52**
	True	**1**	**32**	**33**
	Total	**48**	**37**	**85**

Table A.8: M1M2 - Fine-Tuning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**41**	**11**	**52**
	True	**6**	**27**	**33**
	Total	**47**	**38**	**85**

Table A.9: M1M3 - Fine-Tuning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**44**	**8**	**52**
	True	**3**	**30**	**33**
	Total	**47**	**33**	**85**

Table A.10: M1M4 - Fine-Tuning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**43**	**9**	**52**
	True	**7**	**26**	**33**
	Total	**50**	**35**	**85**

Table A.11: M1M5 - Fine-Tuning Confusion Matrix

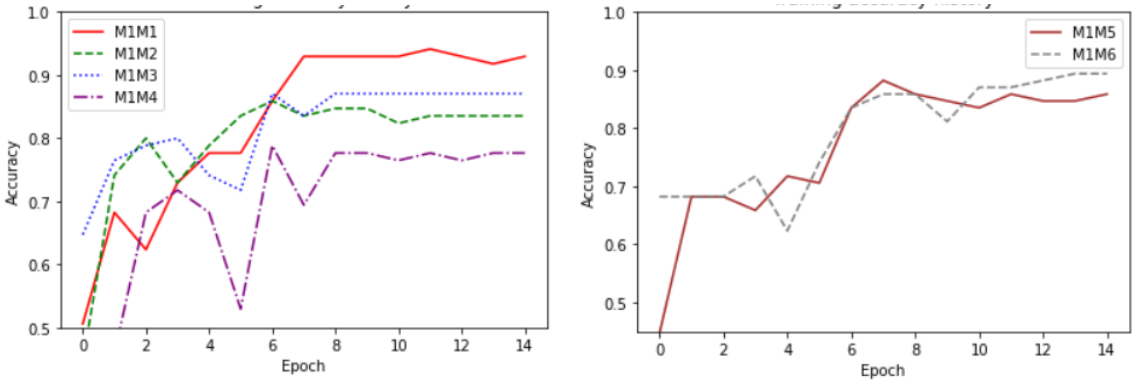
		Predicted Class		
		False	True	Total
Actual Class	False	**46**	**6**	**52**
	True	**6**	**27**	**33**
	Total	**52**	**33**	**85**

Table A.12: M1M6 - Fine-Tuning Confusion Matrix

		Predicted Class		
		False	True	Total
Actual Class	False	**46**	**6**	**52**
	True	**8**	**25**	**33**
	Total	**44**	**31**	**85**

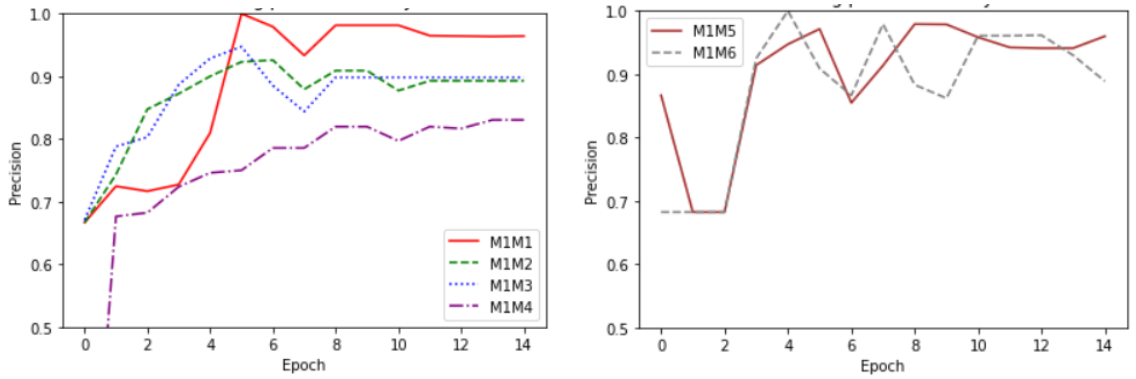
A.3 Classify & Retrieve Method – Training Validation

This Section includes the training validation metrics obtained from the fine-tuning of the six Deep Lan-guage Models (M1M1, M1M2, M1M3, M1M4, M1M5, M1M6) used in the first implemented method.



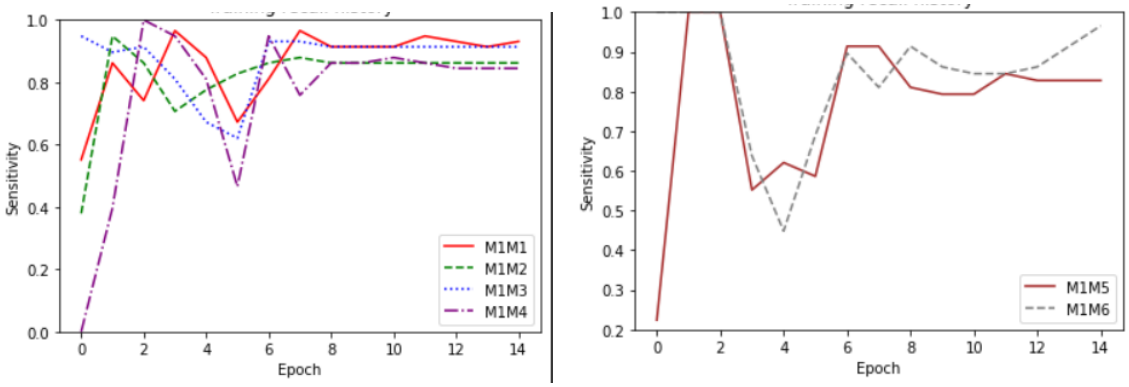
(a) BERT Models (b) RoBERTa Models

Figure A.1: Classify & Retrieve – Evaluation Accuracy



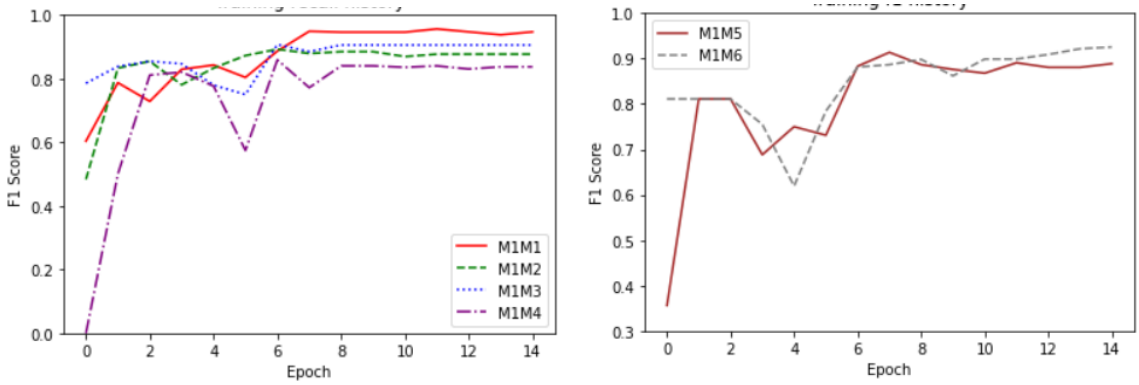
(a) BERT Models (b) RoBERTa Models

Figure A.2: Classify & Retrieve – Evaluation Precision



(a) BERT Models (b) RoBERTa Models

Figure A.3: Classify & Retrieve – Evaluation Sensitivity



(a) BERT Models (b) RoBERTa Models

Figure A.4: Classify & Retrieve – Evaluation F1 Score

References

[1] Does correcting online falsehoods make matters worse?

[2] Maristella Agosti, Fabio Crestani, and Gabriella Pasi, editors. *Lectures on information retrieval: Third European Summer-school, ESSIR 2000, Varenna, Italy, September 11-15, 2000: revised lectures*. Number 1980 in Lecture notes in computer science. Springer, Berlin ; New York, 2001. Meeting Name: European Summer School in Information Retrieval.

[3] Hunt Allcott, Luca Braghieri, Sarah Eichmeyer, and Matthew Gentzkow. The Welfare Effects of Social Media. *American Economic Review*, 110(3):629–676, March 2020.

[4] Hunt Allcott and Matthew Gentzkow. Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, 31(2):211–236, May 2017.

[5] Hunt Allcott, Matthew Gentzkow, and Chuan Yu. Trends in the diffusion of misinformation on social media. page 8.

[6] Felipe Almeida and Geraldo Xexeo. Word Embeddings: A Survey. page 10.

[7] Sumit Banik. Covid fake news dataset, November 2020.

[8] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. *arXiv:1903.10676 [cs]*, September 2019. arXiv: 1903.10676.

[9] Facundo Bre, Juan M Gimenez, and Víctor D Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158:1429–1441, 2018.

[10] Julio Javier Castillo and Pilot Task. *the Fifth Pascal Recognizing Textual Entailment (RTE-5) Evaluation Challenge and the new Textual Entailment Search*.

[11] Chandra Churh Chatterjee. Basics of the Classic CNN, July 2019.

[12] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *ACM SIGKDD Explorations Newsletter*, 19(2):25–35, November 2017.

[13] Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Hui Jiang, and Xiaodan Zhu. Revisiting Word Embedding for Contrasting Meaning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 106–115, Beijing, China, 2015. Association for Computational Linguistics.

[14] Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. page 8.

[15] Classification: ROC Curve and AUC | Machine Learning Crash Course.

[16] Joshua Cohen. The Complex Global Evolution Of Coronavirus Mask Rules. Section: Healthcare.

[17] Limeng Cui and Dongwon Lee. Coaid: Covid-19 healthcare misinformation dataset, 2020.

[18] Robert Dale. The return of the chatbots. *Natural Language Engineering*, 22(5):811–817, September 2016.

[19] Marie-Catherine de Marneffe, Anna N Rafferty, and Christopher D Manning. Finding Contradictions in Text. page 9.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[21] Nicholas DiFonzo and Prashant Bordia. *Rumor psychology: Social and organizational approaches*. Rumor psychology: Social and organizational approaches. American Psychological Association, Washington, DC, US, 2007. Pages: x, 292.

[22] Fhel Dimaano. What is Machine Learning?, October 2019.

[23] Valentina Dragos. Detection of contradictions by relation matching and uncertainty assessment. *Procedia Computer Science*, 112:71–80, January 2017.

[24] Rohith Gandhi. Naive Bayes Classifier, May 2018.

[25] Thushan Ganegedara. Light on Math ML: Intuitive Guide to Understanding GloVe Embeddings, December 2020.

[26] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The Paraphrase Database. page 7.

[27] Swapnil Ghuge and Arindam Bhattacharya. Survey in Textual Entailment. page 28.

[28] Wael H. Gomaa and Aly A. Fahmy. *A Survey of Text Similarity Approaches*.

[29] Prashant Gupta. Decision Trees in Machine Learning, November 2017.

[30] A. A. Hakim, A. Erwin, K. I. Eng, M. Galinium, and W. Muliady. Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In *2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–4, October 2014.

[31] Abdullah Hamid, Nasrullah Shiekh, Naina Said, Kashif Ahmad, Asma Gul, Laiq Hassan, and Ala Al-Fuqaha. Fake News Detection in Social Media using Graph Neural Networks and NLP Techniques: A COVID-19 Use-case. *arXiv:2012.07517 [cs]*, November 2020. arXiv: 2012.07517.

[32] Sanda Harabagiu. Negation, Contrast and Contradiction in Text Processing. page 8.

[33] McKenzie Himelein-Wachowiak, Salvatore Giorgi, Amanda Devoto, Muhammad Rahman, Lyle Ungar, H Andrew Schwartz, David H Epstein, Lorenzo Leggio, and Brenda Curtis. Bots and Misinformation Spread on Social Media: Implications for COVID-19. *Journal of Medical Internet Research*, 23(5):e26933, May 2021.

[34] John Hiscott, Magdalini Alexandridi, Michela Muscolini, Evelyne Tassone, Enrico Palermo, Maria Soultsioti, and Alessandra Zevini. The global impact of the coronavirus pandemic. *Cytokine & Growth Factor Reviews*, 53:1–9, June 2020.

[35] Kyle Hunt, Puneet Agarwal, and Jun Zhuang. Monitoring Misinformation on Twitter During Crisis Events: A Machine Learning Approach. *Risk Analysis*, n/a(n/a). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/risa.13634.

[36] S. G. Kanakaraddi and S. S. Nandyal. Survey on Parts of Speech Tagger Techniques. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–6, March 2018.

[37] Anne Kao and Steve R. Poteet. *Natural Language Processing and Text Mining*. Springer Science & Business Media, March 2007. Google-Books-ID: CVtxFWbKT7wC.

[38] Dhruvil Karani. Introduction to Word Embedding and Word2Vec, September 2020.

[39] Wahab Khan, Ali Daud, Jamal A. Nasir, and Tehmina Amjad. A survey on the state-of-the-art machine learning models in the context of NLP. *Kuwait Journal of Science*, 43(4), November 2016. Number: 4.

[40] Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabadi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, Erfan Sadeqi Azer, Niloofar Safi Samghabadi, Mahsa Shafaei, Saber Sheybani, Ali Tazarv, and Yadollah Yaghoobzadeh. ParsiNLU: a suite of language understanding challenges for persian. *arXiv*, 2020.

[41] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

[42] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, November 2006.

[43] J. Li, A. Sun, J. Han, and C. Li. A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[44] Luyang Li, Bing Qin, and Ting Liu. Contradiction Detection with Contradiction-Specific Word Embedding. *Algorithms*, 10(2):59, May 2017.

[45] Yunyao Li, Tyrone Grandison, Patricia Silveyra, Ali Douraghy, Xinyu Guan, Thomas Kieselbach, Chengkai Li, and Haiqi Zhang. Jennifer for COVID-19: An NLP-Powered Chatbot Built for the People and by the People to Combat Misinformation. page 9.

[46] Elizabeth D Liddy. Natural Language Processing. page 15.

[47] Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1501–1511, Beijing, China, 2015. Association for Computational Linguistics.

[48] 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. *arXiv:1907.11692 [cs]*, July 2019. arXiv: 1907.11692.

[49] L. Ma and Y. Zhang. Using Word2Vec to process big text data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2895–2897, October 2015.

[50] Juan Carlos Medina Serrano, Orestis Papakyriakopoulos, and Simon Hegelich. NLP-based Feature Extraction for the Detection of COVID-19 Misinformation Videos on YouTube. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, July 2020. Association for Computational Linguistics.

[51] Shahan Ali Memon and Kathleen M. Carley. CMU-MisCov19: A Novel Twitter Dataset for Characterizing COVID-19 Misinformation, September 2020. If you use this dataset, please cite our recently accepted paper on "Characterizing COVID-19 Misinformation Communities Using a Novel Twitter Dataset" at MAISON Workshop at CIKM 2020 as follows: "Shahan Ali Memon and Kathleen M. Carley. Characterizing COVID-19 Misinformation Communities Using a Novel Twitter Dataset, In Proceedings of The 5th International Workshop on Mining Actionable Insights from Social Networks (MAISoN 2020), co-located with CIKM, virtual event due to COVID-19, 2020." The preprint version of the paper can found at https://arxiv.org/abs/2008.00791.

[52] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. page 9.

[53] George A. Miller. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. Google-Books-ID: Rehu8OOzMIMC.

[54] Amit Mishra and Sanjay Kumar Jain. A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3):345–361, 2016.

[55] M. Mitra and B.B. Chaudhuri. Information Retrieval from Documents: A Survey. *Information Retrieval*, 2(2):141–163, May 2000.

[56] Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting Word Vectors to Linguistic Constraints. *arXiv:1603.00892 [cs]*, March 2016. arXiv: 1603.00892.

[57] 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.

[58] Vincent J. Munster, Marion Koopmans, Neeltje van Doremalen, Debby van Riel, and Emmie de Wit. A Novel Coronavirus Emerging in China — Key Questions for Impact Assessment. *New England Journal of Medicine*, 382(8):692–694, February 2020. Publisher: Massachusetts Medical Society _eprint: https://doi.org/10.1056/NEJMp2000929.

[59] Dhiraj Murthy. *Twitter: social communication in the Twitter age*. Digital media and society. Polity, Cambridge, 2013. OCLC: ocn805013923.

[60] K Nalini and Dr L Jaba Sheela. Survey on Text Classification. *International Journal of Innovative Research in Advanced Engineering*, 1(6):6, 2014.

[61] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A Survey on Open Information Extraction. *arXiv:1806.05599 [cs]*, June 2018. arXiv: 1806.05599.

[62] Artem Oppermann. What is Deep Learning and How does it work?, August 2020.

[63] D. W. Otter, J. R. Medina, and J. K. Kalita. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[64] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[65] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[66] Andrew Perrin. 65% of adults now use social networking sites – a nearly tenfold jump in the past decade. page 12.

[67] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv:1802.05365 [cs]*, March 2018. arXiv: 1802.05365.

[68] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Computing Surveys*, 51(5):1–36, January 2019.

[69] Ashis Pradhan. Support vector machine-a survey. *International Journal of Emerging Technology and Advanced Engineering*, 2(8):82–85, 2012.

[70] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Cambridge University Press, January 2000. Google-Books-ID: qnWQU9C8bDkC.

[71] Zuzana Reitermanova. Data splitting. In *WDS*, volume 10, pages 31–36, 2010.

[72] I Rish. An empirical study of the naive Bayes classifier. page 6.

[73] Xin Rong. word2vec Parameter Learning Explained. *arXiv:1411.2738 [cs]*, June 2016. arXiv: 1411.2738.

[74] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer Learning in Natural Language Processing. page 4.

[75] Yusaku Sako. "Is the term "softmax" driving you nuts?", August 2018.

[76] Mark Sanderson and Justin Zobel. Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. page 8.

[77] Roy Schwartz, Roi Reichart, and Ari Rappoport. Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 258–267, Beijing, China, 2015. Association for Computational Linguistics.

[78] Gautam Kishore Shahi, Anne Dirkson, and Tim A. Majchrzak. An exploratory study of covid-19 misinformation on twitter, 2020.

[79] W. Shen, J. Wang, and J. Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, February 2015. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[80] Jieun Shin, Lian Jian, Kevin Driscoll, and François Bar. The diffusion of misinformation on social media: Temporal pattern, message, and source. *Computers in Human Behavior*, 83:278–287, June 2018.

[81] Advaith Siddharthan. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298, 2014.

[82] Tanu Singhal. A Review of Coronavirus Disease-2019 (COVID-19). *The Indian Journal of Pediatrics*, 87(4):281–286, April 2020.

[83] Jonathan Slocum. A SURVEY OF MACHINE TRANSLATION: ITS HISTORY, CURRENT STATUS, AND FUTURE PROSPECTS. *Computational Linguistics*, 11(1):17, 1985.

[84] Kamilya Smagulova and Alex Pappachen James. A survey on lstm memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228(10):2313–2324, 2019.

[85] M. Sundermeyer, H. Ney, and R. Schlüter. From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):517–529, March 2015. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[86] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[87] Mikalai Tsytsarau, Themis Palpanas, and Kerstin Denecke. Scalable detection of sentiment-based contradictions. *DiversiWeb, WWW*, 1:9–16, 2011.

[88] Twitter Moments guidelines and principles.

[89] Twitter Revenue and Usage Statistics (2020), October 2018.

[90] Updating our approach to misleading information.

[91] Using AI to detect COVID-19 misinformation and exploitative content.

[92] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. page 11.

[93] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, December 2016.

[94] Congle Zhang, Gui-Rong Xue, Yong Yu, and Hongyuan Zha. Web-scale classification with naive bayes. In *Proceedings of the 18th international conference on World wide web - WWW '09*, page 1083, Madrid, Spain, 2009. ACM Press.

[95] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1253, 2018. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1253.

[96] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, December 2010.

[97] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.