DETECTING HEALTH MISINFORMATION ONLINE USING DEEP LEARNING METHODS

Dione Morales

Bachelor of Engineering
Computer Engineering Stream

School of Engineering
Macquarie University

November XX, 2018

Supervisors: Associate Professor Adam Dunn and Dr Rex Di Bona

ACKNOWLEDGMENTS

I would like to acknowledge the countless number of people that have helped me get to where I am today.

I would also like to thank my supervisors, Rex Di Bona and Adam Dunn, for giving me the freedom and support to be able to work on my current topic. I am deeply grateful for the help and opportunities they have given me.

STATEMENT OF CANDIDATE

I, Dione Morales, declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the School of Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment an any academic institution.

Student's Name: Dione Morales

Student's Signature: Dione Morales (electronic)

Date: September 9, 2018

ABSTRACT

With the popularity and ubiquity of social media platforms in today's society, the amount and rate at which information propagates online greatly outnumbers the resources available that can evaluate the quality and credibility of the information that gets shared. This is becoming an increasingly growing issue due to the clickbait model that is commonly adopted by social media platforms and the lack of rigour surrounding the publishing of content online, causing an increase in the number of articles that contain misinformed content. This project aims to investigate the performance of deep learning techniques in evaluating the credibility of information of health-related articles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the popularity and ubiquity of social media platforms in today's society, the amount and rate at which information propagates online greatly outnumbers the resources available that can evaluate the quality and credibility of the information that gets shared. This is becoming an increasingly growing issue due to the clickbait model that is commonly adopted by social media platforms and the lack of rigour surrounding the publishing of content online [47], causing an increase in the number of articles that contain misinformed content [48] [49].

The lack of resources that attempt to minimise the spread of misinformation can be attributed to the expert-level knowledge required to determine the credibility of information since a variety of factors such as the actual information, information sources, conflicts of interest and writing style of the article must be evaluated to be able to determine its quality and credibility. This issue is further exacerbated in specific domains such as for health-related content, as the spread of misinformation can have a detrimental effect on people and their communities that don't have access to these kinds of limited resources.

Thus, by developing a method that is capable of automatically assessing the quality and credibility of an article - various methods that intervene between an article and a user can then be developed to minimise the spread and effects of misinformation as the reliance on expert-level knowledge is no longer present.

1.1 Project Overview

This section details the scope of the project and its associated outcomes outlining the various tasks that must be accomplished to successfully complete the project.

One of the key components required to minimise the propagation of misinformation online is to have the ability of automatically evaluating and quantifying the credibility of articles. However, traditional automated methods - such as machine learning-based techniques, still require the domain knowledge of experts to be able to develop the features required by the model. Thus, this project aims to investigate the performance of

Deep Learning-based (DL) techniques in evaluating the credibility of information within domain-specific articles via the classification of set criteria that have deemed to be highly correlated with articles that have low credibility. Specifically, this project will focus on evaluating the credibility of online health articles related to vaccination due to the commonly misinformed and controversial views associated with its effects [15].

1.2 Motivation

The propagation of misinformed content can cause people within specific communities to adopt beliefs and practices that can have harmful effects to themselves and also to the people around them. In the context of health-related content specifically, these beliefs and practices can take form in the misuse or mistreatment of medicine or illnesses. A popular example where this issue has emerged is with the misinformed views by certain communities on the ramifications of vaccinating children.

The effects of vaccination and their relation to causing autism within children is a popular example where the belief of inaccurate facts has had a detrimental effect to specific communities and their surrounding environments. The measles outbreak in Minnesota which occurred in April 2017 is a specific case where the misinformed views on the effects of vaccination caused people within a Somali-American community located in the United States to forego the vaccination of their children causing an outbreak of measles, a disease which was declared to be eliminated from the United States in the year 2000 [21]. Within this specific instance, it was determined that the outbreak of measles was caused by a decline in the coverage of vaccination due to concerns of its effects in relation to the causation of autism. This can be seen in the decreasing trend of the percentage for 24 month-old children that received the measles-mumps-rubella vaccine which fell to a low of 40% in 2014 where it was at a 90% high 10 years prior.

1.3 Aims

With the primary objective of this project being the evaluation on the effectiveness of deep learning models in determining the credibility of online vaccine-related health articles. Due to the complexity of this project, a set of activities - divided into main goals and stretch goals, have been defined to ensure that the completion of this project remains feasible in the given time frame. The completion of all activities categorized as main goals will signal the realization of the primary objective and the completion of the project. Stretch goals are activities of interest that have been identified as non-essential to the completion of the primary objective but will be worked on after the completion of the project.

Main Goals

- Implement and evaluate the performance of machine learning methods for assessing the quality and credibility of vaccine-related text.

- Implement a deep neural network method to assess the quality and credibility of vaccine-related text and compare the performance with the previous approaches.

Stretch Goals

- Evaluate the effect of transfer learning methods on the training time and performance of the proposed deep neural network method.

Chapter 2

Background and Related Work

A literature review has been conducted to develop an understanding on the research that has been done in the assessment of the credibility of information, specifically in the context of information related to health and the limitations and capabilities of machine learning techniques and how it differs from deep learning-based methods for the task of document classification.

2.1 Assessing the Quality and Credibility of Health Information

To have the capability of automating the process of evaluating the quality or credibility of online information, a definition that outlines of what is required by an article to be considered as a credible source of information must be developed. While there has been a significant amount of research that has been done on the development of tools and frameworks that aim to assess the credibility of online health information, there is currently no standardized method or benchmark that is universally used. Tools and frameworks that have been identified to be applicable in the context of this project are: DISCERN [3], HealthNewsReview [4] and Quality Index for health-related Media Reports (QIMR) [53].

2.1.1 Checklists and Criteria

DISCERN

DISCERN [3] is a questionnaire designed to assess the reliability of a publication, it consists of 16 questions each with a Likert scale, ranging from 1 (no) to 5 (yes) and is divided into 3 sections. The first section (questions 1 - 8) investigate the reliability of the information and is comprised of questions such as ‘Are the aims clear?’, ‘Is it balanced and unbiased?’ and ‘Does it refer to areas of uncertainty?’. The second section (questions 9 - 15) assesses the quality of information provided by the publication for treatment choices and is composed of questions such as ‘Does it describe the risks/benefits of each treatment?’ and ‘Does it provide support for shared decision-making?’. The final section (question 16) assesses the overall rating of the publication ('Based on the answers to all of the above questions, rate the overall quality of the publication as a source of information about treatment choices').

HealthNewsReview

HealthNewsReview [4] provides a set of 10 criteria designed to act as a framework for evaluating the credibility of health-related media. The criteria is based on the various elements that all health-related media should consist of, the criteria is composed of criterion such as 'Does the story compare the new approach with existing alternatives?', 'Does the story use independent sources and identify conflicts of interest?' and 'Does the story appear to rely solely or largely on a news release?'.

QIMR

QIMR [53] is a tool developed to monitor the quality of health research reports presented in the media. The tool considers 5 main factors that have been deemed to be correlated with the quality of research reports based on interviews with health journalists and researchers. The 5 main factors are: background information provided, sources of information used, manner in which results were analysed, context of the research and the validity of their methodology.

2.1.2 Studies Applying Checklists to Health Information Online

There has been an extensive amount of research conducted which aimed to assess the quality of content for various health-related domains using one of the aforementioned tools [17] [18] [26] [46]. A factor shared by these commonly performed studies however, is the use of experts to leverage the tool in assessing the quality and credibility of the information, indicating that commonly used tools such as DISCERN is designed to be used by domain experts in order to produce reliable and consistent results. This sentiment is shared by Batchelor et al. [11] who evaluated performance of the DISCERN tool when used by health professionals and patients. Due to the resource intensive nature of expertly performing this task, the number of articles evaluated, among the various studies examined have been limited ranging between 10 - 300 information sources. This highlights the issue that these tools are not designed to be capable of evaluating the quality and credibility of articles at the scale and speed required to match the pace of online activity and thus must be adapted for the automation of this process. While there has been work in adapting criteria to better encompass the factors that are correlated to the quality and credibility of information within a specific context, such as by Matsoukas et al. [30], who expanded the DISCERN tool to improve its capability in assessing the quality of online information, there has not been any published work found that aims to adapt these tools to provide the capability of autonomously assessing the quality and credibility of resources.

2.2 Document Classification Methods

The task of document classification is a heavily researched topic due to its wide number of applications in various domains. Formally, document classification is defined to be the task of finding a classifier $f : D \to L$ where $D = \{d_1, d_2, \ldots, d_n\}$ is a collection of documents and $L = \{l_1, l_2, \ldots, l_k\}$ is the set of possible labels that a document d can be classified as.

$$f : D \to L \text{ where } f(d) = l \tag{2.1}$$

Common applications of document classification algorithms are: organization and filtering of news articles, document retrieval, opinion mining, email classification and spam filtering [7].

Traditional machine learning-based approaches for document classification, such as Naive Bayes, Support Vector Machines (SVM) and Random Forests, require the manual extraction of features [7] [9] [28] [38] as they are incapable of performing feature learning [10]. These features are typically either hand-crafted and domain specific, requiring expert-level knowledge for the task domain [38] or are simple and general but prone to the loss of information (e.g. being unable to account for context) since these features, such as term frequency-inverse document frequency (TF-IDF) scores or bag of words (BoW) models, are unable to represent and account for the sequential nature of text.

2.2.1 Representing Text as Features

Due to the unstructured nature of natural text, it is common practice for natural language processing (NLP) tasks to transform the text into a structured representation that minimises the loss of information stored within the original text whilst allowing the model to learn and analyse the features associated with the classification task. However, despite traditional representation methods such as count-based approaches (e.g. TF-IDF) or probabilistic-based approaches (e.g. N-Grams) showing respectable performances in the classification of texts [54] [55], these representations are typically sparsely distributed and have a high dimensionality that scales with the vocabulary size, has difficulty in generalising over unfamiliar information or writing styles [43] as the texts are represented using simple mechanisms that are unable to represent high-level information such as context or the semantic relationship of words. Due to the high dimensionality and sparseness of these features, they are computationally intensive to calculate and can require a relatively larger amount of space for storage.

Word embeddings are a more sophisticated language model that overcome these limitations as they are created by learning word vectors through the optimisation for a different task e.g. the prediction of a word based on its context. Through this, the learned embeddings are able to store information such as the similarity of different contexts, syntactic and semantic information within the text and analogies. These embeddings are relatively more efficient and require less time to compute due to their smaller dimensionality. Whilst these embeddings are able to be created using shallow neural networks, deep learning-based models for NLP tasks utilise these embeddings which can also be at the character, phrase or sentence levels [52]. Word embeddings, such as Word2Vec or GloVe, have produced state-of-the-art results in not only classification tasks [24] but also for a wide range of NLP tasks such as image annotation [51] and sentiment analysis [16], which used the popular word2vec embedding proposed by Mikolov et al. [33]

2.2.2 Machine Learning Methods for Document Classification

Naive Bayes Classifier

Naive Bayes is a probabilistic classifier that functions on an assumption in how the data (i.e. the words in a document) is generated. The assumption that these Bayesian classifiers are based on is that the distribution of different words within a corpus are independent from each other. Despite this assumption clearly being wrong when considering the distribution of words within a document (due to the sequential nature of text), Naive Bayes classifiers are still able to perform well in document classification tasks such as the filtering of spam emails [44].

Naive Bayes classifiers utilize Bayes' theorem, which attempts to find the probability of an event B occurring given some prior condition A i.e. $P(B|A)$. In the context of document classification, Naive Bayes classifiers classifies a document d by calculating the probabilities that the document belongs for all labels $l_i \in L$ and then selecting the highest probability [9]:

$$P(L = l_i|d) = \frac{P(l_i)P(d|l_i)}{P(d)} \tag{2.2}$$

According to Aggarwal et al. [7], there are two main types of Naive Bayes classifiers that are commonly used, they are the multivariate Bernoulli model and the multinomial model. The main discriminating factor between these two models is that the Bernoulli model does not take into account the frequency of words as it represents a document using a vector of binary features which signify the presence or absence of words, based on some vocabulary, for a given document. This is in contrast with the multinomial model, which accounts for the frequency of words as the document is represented using a BoW model. Deciding on which model to use for document classification largely depends on the size of the vocabulary as, according to McCallum et al. [31], the multinomial models almost always outperforms the Bernoulli models if the size of the vocabulary is large (> 1000) or even if the size has been optimally chosen for both models.

Support Vector Machines

SVMs are a type of supervised machine learning-based binary classifiers that are extensively used for document classification due to their capability of handling the high dimensional and sparse nature of the common techniques used to represent text documents [25]. SVMs are able to classify a document by using a hyperplane to separate the different classes into separate regions. The hyperplane used is determined by choosing the one that maximises the margin of separation (i.e. The Euclidean distance between the hyperplane and all data points in the representation space) between the two classes [7].

Consider the illustration shown in Figure 2.1 that presents three two-dimensional hyperplanes, A, B and C, which separates the classes 'x' and 'o'. Visually, we can determine that A is the hyperplane that maximises the margin of separation, thus, A will be used as the decision rule to classify any new article based on its location in the representation space with respect to A. Mathematically, determining which hyperplane to use as the decision rule is an optimisation problem that attempts to maximise the margin represented as shown in Equation 2.3 but is often re-framed to the minimisation of Equation 2.4:

$$\text{maximise: } \frac{1}{2}||\frac{1}{w}||^2 \tag{2.3}$$

$$\text{minimise: } \frac{1}{2}||w||^2 \tag{2.4}$$

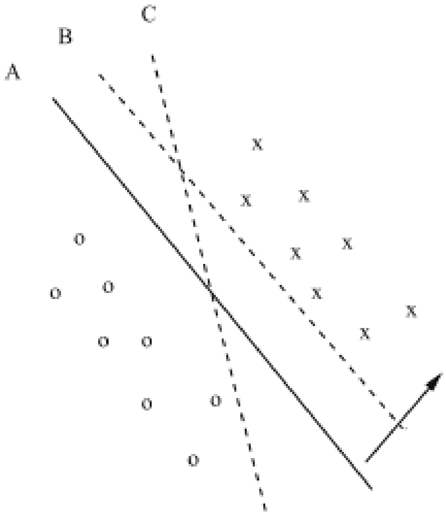


Figure 2.1: Determining the optimal hyperplane [7]

This technique of determining the hyperplane requires that the two classes are linearly separable. In situations where this isn't true, the kernel trick [8] is applied which is essentially a function that maps data in a particular representation space to another representation space of a different dimension. This change in dimensionality can allow the classes to become linearly separable and thus a hyperplane can then be constructed that separates each class as illustrated in Figure 2.2.

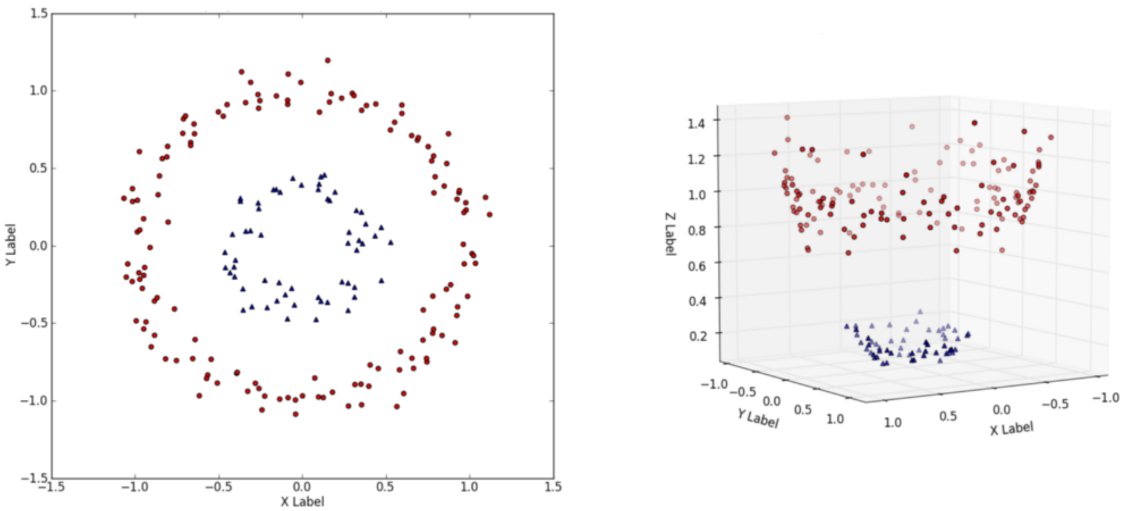


Figure 2.2: (Left) Non-linearly distributed data of two classes in a 2D representation space.
(Right) Linearly separable data of the same classes in a 3D representation space after the application of a kernel function [6].

2.2.3 Deep Learning Methods for Document Classification

Deep learning models are a class of machine learning models that have the capability of automatically learning a hierarchical representation of data [10]. These hierarchical representations are constructed through the use of artificial neural networks, the main underlying mechanism of deep learning models. The most commonly used models for document classification are Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs) - a variant of RNNs and Convolutional Neural Networks (CNNs).

For the task of text classification, there are commonly used, open datasets for specific classification tasks that aim to provide a benchmark in which the performance of different implementations can be compared and to facilitate the data needs of deep learning models.

Recurrent Neural Networks

RNNs and LSTMs are commonly used for document classification, or NLP tasks in general, due to their ability to create language models that are able to capture the context and relationships of words within documents over long distances and represent this information at a much more sophisticated level when compared to traditional language models such as TF-IDF or n-grams [52].

RNNs have had widespread use in solving NLP-related tasks due to their ability in capturing the sequential nature of text at the character, word or sentence level. Consequently, RNNs are capable of creating language models that account for the semantic meaning of words based on the previously occurring words in the sentence, allowing models to be capable of understanding the difference between similar words or phrases (e.g. that the word 'dog' is likely referring to the animal whereas 'hot dog' would be more likely to refer to food). Accounting for RNNs' capability in handling variable length inputs (e.g. long sentences, paragraphs or documents), they have shown to produce state-of-the-art results in classification tasks such as sentiment, question and topic classification [24].

The simplest form of an RNN, as illustrated in Figure 2.3, consists of three layers: the input layer x_t, hidden state s_t and the output layer o_t, where t represents the current timestep. The input layer is typically represented as a one-hot encoding or embedding, the output layer is the resulting output which can take many forms, most commonly, it is the output of the softmax function and the hidden state is essentially the memory of the network as it captures and incorporates the information from previous timesteps into the current one. The hidden state is calculated by evaluating Equation 2.5:

$$s_t = f(Ux_t + Ws_{t-1}) \tag{2.5}$$

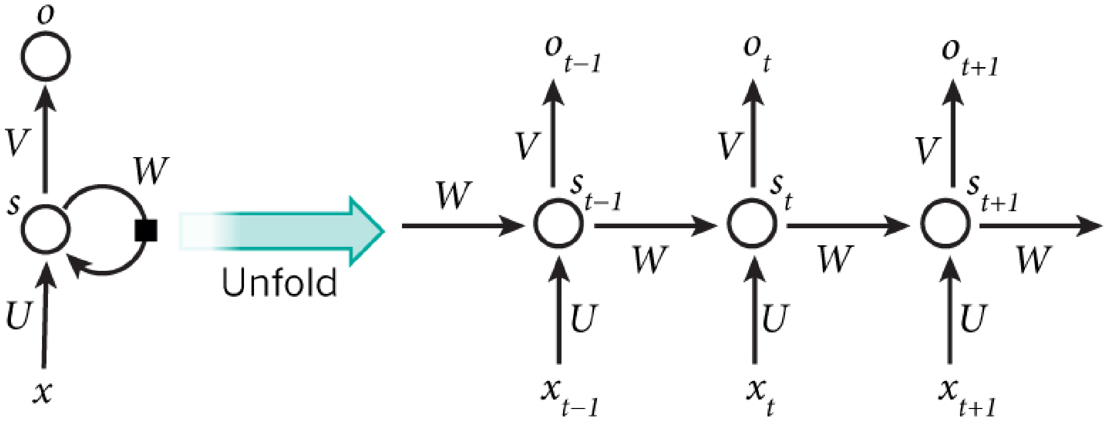


Figure 2.3: Layout of a simple RNN [5]

Where f is a nonlinear function e.g. Rectifier Linear Unit (ReLU), U, V and W are weight matrices that are shared across timesteps [52].

LSTMs [23] are a variant of RNNs that improves on it by introducing a 'forget' gate which regulates a cell's state within the network as it can allow information from different cells to be removed or added. This addition allows LSTMs to overcome the vanishing and exploding gradient problem [45] that feedforward networks i.e. RNNs are prone to.

Convolutional Neural Networks

Despite CNNs being initially developed for object recognition tasks [29], they are commonly employed for text classification tasks such as sentence classification and sentiment analysis as they have shown to produce competitive results when compared to RNNs and LSTMs [19] [27] [35].

In the context of text classification, a typical CNN is composed of the following main elements [52]:

- Kernels - Convolutional filters that acts as a sliding window function through an embedding matrix. CNNs typically employ hundreds of kernels each of which learns to extract for specific n-gram pattern.
- Pooling layer - Commonly either a max or average pooling layer, which maps the input to a fixed dimension in order to reduce the dimensionality of the output and ensuring that the most salient n-gram features of a sentence is kept.
- Nonlinear activation function - Such as a ReLU function applied to the results to output a prediction.

By stacking the kernels and pooling layers, deep CNNs can be constructed which can automatically capture an abstract and rich representation of the information [52]. These representations are considered to be efficient when compared to traditional representation methods (e.g. n-grams) as they don't require the storage of the entire vocabulary and is not as computationally intensive. CNNs are also more computationally efficient when compared to RNNs and LSTMs, CNNs typically require larger amounts of data in order to produce competitive results against its RNN and LSTM counterparts due to the higher number of trainable parameters that CNNs have. Another limitation of CNNs is their inability to model long-distance relationships and preserving the sequential nature of text within its representations.

2.2.4 Classifying Documents with Limited Labelled Examples

Typically, large amounts of training data is required to train a deep learning model in learning the language model for state of the art results, in the task of document classification for instance, the size of commonly used non-domain specific datasets range from hundreds of thousands of training examples to millions [20] [54]. For situations where you are required to procure a completely new dataset, such as training a deep learning model for a non-standard text classification task, it can be unfeasible or not worthwhile to invest the time and resources in creating the dataset.

Transfer Learning

Transfer learning involves the repurposing of an already existing deep learning model that has been trained to perform a different, but similar, task using an already existing and available dataset. It functions on the idea that the features automatically learned by a model for some similar task is general enough such that they can then be utilised on the completely new task.

Howard et al. [24] has proposed a transfer learning methodology for text classification which was designed to be able to develop models with state-of-the-art results for tasks where there is a limited amount of training data available.

The proposed method involves the use of an inductive learning technique [36]:

1. Create a language model to capture general features of the language by training on a general-domain corpus (e.g. wikitext103 [32]).
2. Learn task specific features by fine-tuning the language model using the target task data.
3. Adapt the high-level representations of the classifier that uses the language model, while preserving the low-level representations, using gradual unfreezing [24]

Whilst transfer learning is useful to train models for non-standard tasks where access to data is limited, it is also an optimisation technique, using previously acquired knowledge, to save time or improve model performance [36]. This potential performance improvement is illustrated in Figure 2.4 and is also demonstrated by Howard et al. [24] for common text classification tasks such as question, topic classification and sentiment analysis.

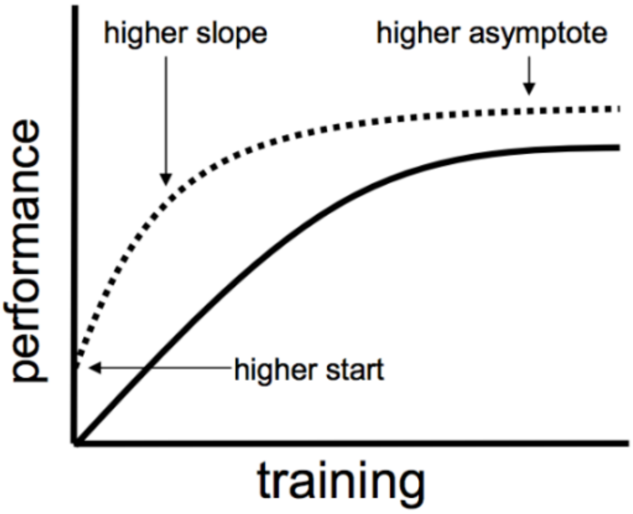


Figure 2.4: Potential improvements in performance using transfer learning [14]

2.3 Scaling the Assessment of Quality and Credibility of Health Information

The limited amount of articles that were evaluated was a common theme among the studies that aim to assess the credibility and quality of information [11] [17] [18] [26] [46]. This makes evaluating the credibility of online articles impractical when using the existing assessment tools as described in Section 2.1.1, due to the persistence, pace and volume of the production and propagation of online content. Due to the capabilities of machine learning models, they can be leveraged to accomplish the task of automatically assessing the credibility of online articles on a large scale as models, such as deep learning models, have shown the ability to develop high-level representations of language which can then be leveraged to be able to differentiate between credible and misinformed articles.

Chapter 3

Methods

3.1 Approach

Through a review of the relevant literature, I found that there is value in being able to appraise the credibility of online health information for use in tools that could help limit the spread of misinformation. However, it is a challenge to apply credibility appraisal tools at scale because the process is time-consuming and often requires a certain level of expertise to use. Tools for automating this task are therefore likely to be of value but to date most research in the area simply aims to label fake news or uses heuristics to identify misinformation available online. In this chapter, I describe the approach I used to address this challenge using machine learning to train classifiers capable of predicting whether an online article meets a set of credibility criteria.

This involved obtaining training and testing data consisting of online vaccine-related articles along with its associated credibility that were manually determined by a team of health professionals using a pre-existing framework (Section 3.2).

Once the dataset was created, a set of machine learning and deep learning models were implemented (Section 3.3). The machine learning models comprised of the implementation of widely used text classifiers, Naive Bayes (NB) and Support Vector Machines (SVMs), in combination with a variety of textual representation methods. The deep learning model involved the construction of a Quasi-Recurrent Neural Network (QRNN) in conjunction with a fine-tuned language model (LM).

The models are then analysed and evaluated on their feasibility for real-world use based on their ability to correctly predict the label for each criteria and other factors such as the training time of the models and their storage requirements (Section 3.4).

3.2 Study Data

3.2.1 Dataset

I obtained a set of vaccine-related online articles via the URLs included in a set of 6.59 million Twitter posts (tweets) from 1.86 million Twitter users collected between January 2017 and March 2018. There were 1.27 million unique URLs included in the set of tweets. A majority of the URLs led to web pages that could not be used as they were either broken or were links to other social media posts and YouTube videos. After removing web pages that could not be used in the analysis, I finally included a set of 3,348 vaccine-related online articles (the corpus). The content of the set of online articles included in the corpus ranged from descriptions of recent research to discussions (Figure 3.1).

The articles included in the corpus were collected by selectively accessing the extracted URLs embedded within tweets that contained specific keywords related to vaccination on Twitter. This was done to ensure that the articles used represented, to some degree, the articles that are most likely to contribute to the effects of the propagation of misinformation as the articles that have been collected and labelled are the ones being shared and discussed online.

3.2.2 Credibility Criteria and Appraisal

The credibility criteria were developed by a team of three researchers from the Centre for Health Informatics as part of a separate project funded by the National Health & Medical Research Council. The seven criteria describe a set of desirable characteristics for online health information (Table 3.1). The development of the criteria was based on a set of existing tools and checklists (see Section 2.1), which was adapted by the researchers to be applicable to online articles about vaccination.

Prior to creating the dataset for this project, a small pilot experiment was conducted by the team that developed the criteria to ensure that the simplicity and objectivity of each criteria allowed for consistent and reliable labels to be produced. In the pilot test of the credibility criteria, each of the three researchers manually and independently labelled the same set of 30 online articles. The team then measured how well their answers matched, resolving differences by discussion, and updating the definitions of the criteria to improve the consistency.

The final tool used by the three investigators included seven criteria, which were inspired and derived from existing checklist-based tools and designed specifically for use with vaccine-related online articles (Table 3.1). They then used the tool to manually assess an additional 470 articles; for each article, the investigators labelled whether the article satisfied each of the individual criteria. Once labelled, the article's credibility was

> Thousands of parents describe their children as ‘fine’ one day, then their children suddenly develop autism (a neurological regression) after vaccines.
>
> What does the science say? It agrees...
>
> But first let’s go through the numbers. The increase in autism cases in the last three decades is truly shocking. Before the 1980s, autism was so rare, it was not even tracked. Remember the eye-opening movie RainMan with Dustin Hoffman? Before that movie came out in the early 1980s, many had no idea what autism was.
>
> At that time, statistics put the rate of autism at 1 in every 8,000 children. By the mid-1990s, it was around 1 in 1000 children, but by the mid-2000s, it had risen to 1 in 250. The epidemic has continued and in 2017, the US National Center for Health Statistics just released the latest rate: 1 in every 36 children now have autism (link to report here). These skyrocketing rates clearly prove there is a true epidemic of autism in the United States.
>
> So what is autism? Autism is brain damage caused by brain inflammation, which can be triggered by the heavy metals used as adjuvants (aka ingredients) in vaccines. Just as thousands of parents saw firsthand with their own children. Read some of these stories here.
>
> Doctors usually try to deny any responsibility and connection to the vaccines they gave by saying “Autism is genetic.” But autism is even listed as a possible reaction (or side effect) of vaccines on some of the vaccine inserts. Unfortunately, doctors rarely see these inserts, favoring the pharmaceutical marketing sheets over real science.
>
> So let’s get to that real science — the type that is NOT funded by the pharmaceutical industry that wants to sell vaccines and then the prescription drugs that these children are on to control the symptoms of autism, an issue possibly caused by the vaccines.

Figure 3.1: Excerpt from a low credibility vaccine-related online article [2].

then aggregated to a credibility score which is defined by the number of criteria that were satisfied. Due to the infeasibility of developing the expert-level knowledge and skill set required to manually label the articles given the time constraints of this project, I did not participate in this labelling process.

When sampling articles, we expected to find relatively few high credibility articles, so the team over-sampled from the subset of articles that were directly linked to research literature. To do this, we used Altmetric, an organisation that provides access to information about how often research articles are mentioned in online media. As a consequence, the proportion of online articles that met the first criterion (identifying the source of evidence) was higher than might be expected from a random sample (Figure 3.2). Relatively few articles satisfied criterion 3 or criterion 7, while most articles satisfied criterion 4 and criterion 6. Overall, the resulting data were relatively unbalanced, which has implications in relation to the training of classifiers, as discussed in Chapter 4.

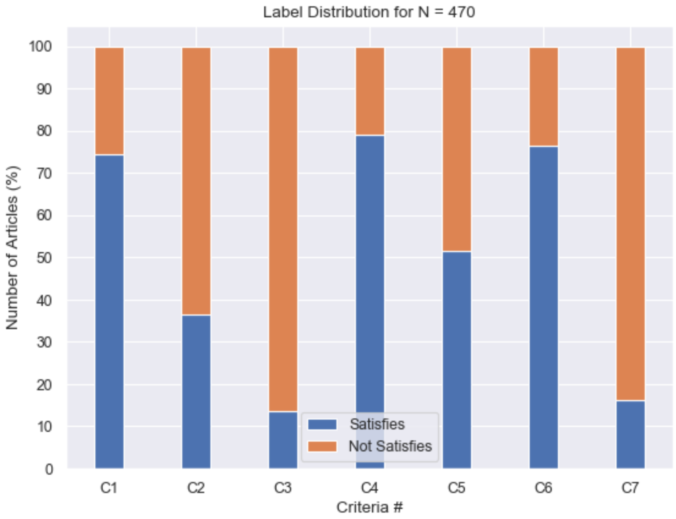


Figure 3.2: Distribution of labels amongst the expert-labelled articles.

I then aggregated the total number of criteria met by an aritcle to produce a credibility score for each of the 470 documents, which produces a score that may take any integer value between 0 and 7. Partly because of the way the online articles were sampled for expert labelling, the credibility scores for the expert-labelled set were unevenly distributed. The largest proportion of the expert-labelled articles received a credibility score of 4; only 14 and 4 articles received a credibility score of 0 and 7, respectively (Figure 3.3).

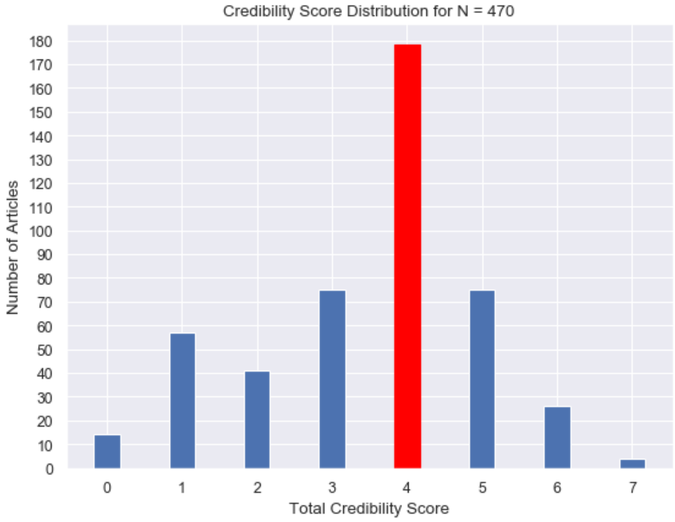


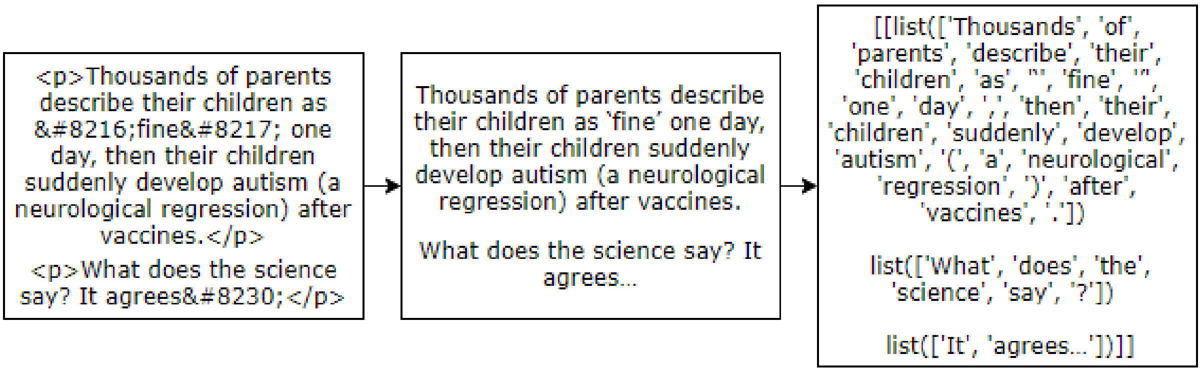
Figure 3.3: Distribution of article scores amongst the expert-labelled articles.

#	Criteria	Description
1	Identifies the information sources supporting a position	Score 1 if the article clearly indicates what sources of information are used to support its main claims or statements.
2	Uses information sources based on objective, scientific research.	Score 1 if the article's main claims are based on objective, scientific research. Score 0 if the article uses anecdotal evidence exclusively to support its main claims.
3	Communicates the strength or weakness of the evidence used to support a position.	Score 1 if the article includes adequate details about the level of evidence offered by the research.
4	Does not exaggerate or overstate a position.	Score 0 if the article: - Uses unjustified sensational language - Presents information in a sensational, emotive or alarmist way - Selectively or incorrectly presents evidence
5	Presents information in a balanced manner.	Score 1 if the article: - Identifies/acknowledges uncertainties and limitations in the research - Acknowledges where an issue is controversial, and includes all reasonable sides in a fair way - Uses a range of information sources - Provides a balanced description of the strengths and weaknesses of the study
6	Uses clear, non-technical language that is easy to understand.	Score 1 if the article: - Is professionally written, with proper grammar, spelling, and composition - Defines any technical jargon, or uses everyday examples to explain the technical terms or concepts
7	Is transparent about sponsorship and funding.	Score 1 if the article: - Clearly distinguishes content intended to promote or sell a product from service from educational and scientific content - Discloses sources of funding for the organisation/website

3.3 Model Implementation

3.3.1 Preprocessing

Preprocessing of the unlabelled and expert-labelled articles of the dataset to obtain its raw article text was performed by removing the superfluous content from the web-scraped pages such as the HTML, CSS and JavaScript elements. They were removed as they were deemed to have no contribution to the credibility of the article's content with regards to the credibility criteria. Once the unstructured article text was obtained, it was transformed into a more functional representation via tokenising each article into sentences that consisted of unigrams (Figure 3.4). The steps involved in the preprocessing were designed to produce a standard set of documents that could then be used consistently across each of the approaches to constructing features (detailed in Section 3.3.2).



```
<p>Thousands of parents
describe their children as
&#8216;fine&#8217; one
day, then their children
suddenly develop autism (a
neurological regression) after
vaccines.</p>
<p>What does the science
say? It agrees&#8230;</p>
```

Thousands of parents describe their children as ‘fine’ one day, then their children suddenly develop autism (a neurological regression) after vaccines.

What does the science say? It agrees…

```
[[list(['Thousands', 'of',
'parents', 'describe', 'their',
'children', 'as', "‘", 'fine', "’",
'one', 'day', ',', 'then', 'their',
'children', 'suddenly', 'develop',
'autism', '(', 'a', 'neurological',
'regression', ')', 'after',
'vaccines', '.'])

list(['What', 'does', 'the',
'science', 'say', '?'])

list(['It', 'agrees…'])]]
```

Figure 3.4: Example process of preprocessing phase showing the original HTML format (left), removal of superfluous content (middle) and the restructuring into a more standardised representation for feature construction (right).

3.3.2 Feature Construction

In document classification tasks, the way document text is transformed into a set of features can have a substantial impact on the overall performance. Traditional approaches to feature representation include simple methods that take the words from a corpus and assign values to them; for example, based on the frequency with which they occur in the text. More sophisticated methods are able to transform document text into multi-dimensional vector representations that captures information such as the similarity between words or the context in which a word is used in.

Bag of Words

Two unigram BoW model variants were constructed using the expert-labelled articles; both of which were limited to a maximum of 20,000 n-grams. Identical BoW models were also constructed for different maximum unigram amounts, specifically, BoW models for

5,000 and 10,000 unigrams were also tested. The first BoW model included all of the words that appeared within the articles. The second BoW model was constructed by removing common English stopwords such as ’the’, ’is’ and ’are’ listed in NLTK’s English stopwords corpus [12] were removed.

Term Frequency - Inverse Document Frequency

TF-IDF scores of the unigrams extracted from the expert-labelled articles which were limited to the top 20,000 scores were calculated and used for evaluation. Two sets of scores were used for evaluation, the first set of scores was calculated with the inclusion of all words in the expert-labelled articles and the second set of scores was calculated with the removal of stopwords.

GloVe

A pre-trained global vector (GloVe) [41] was also used for the evaluation of the ML-based models. This was accomplished by using the mean value of each word’s vector representation. The GloVe vector used was trained on 6 billion tokens, with a vocabulary size of 400,000 and a has dimensionality of 300.

Language Models

A general, non-domain specific language model (LM) was constructed for evaluating the DL-based model. This was achieved by training a Quasi-Recurrent Neural Network (QRNN) using the same hyperparameter settings and training scheme described by Bradbury et al. [13] using the Wikitext-103 corpus. Wikitext-103 is a corpus consisting of approximately 28,000 Wikipedia articles that meet either the ‘Good’ or ‘Featured’ article criteria [32].

The LM was then fine-tuned using the techniques proposed by Howard et al. [24] by using the 2,878 unlabelled articles from the corpus in order to learn the task-specific features for classification. The Wikipedia articles used to create the non-domain specific LM also underwent the same preprocessing process described in Section 3.3.1, except for the removal of webpage elements, which was unnecessary.

3.3.3 Model Construction

In the context of this section, a model refers to the combination of a feature and classifier. The following sections detail the implementation process of the classifiers and the features used in conjunction to create the various models. For the software used to implement the models, the NB and SVM models were implemented using scikit-learn, an open source machine learning library in Python [40]. The QRNN model which utilised the LM and QRNN classifier was implemented using PyTorch [39] and the open-source software developed by Bradbury et al. [13] and Howard et al. [24].

Naive Bayes

A multinomial Naive Bayes (NB) classifier was used for evaluation, based on the reasoning discussed in Section 2.2.2. Along with the 3 types of features (BoW, TF-IDF and GloVe) and their variants (inclusion/exclusion of stopwords) used in conjunction with the NB classifier, a total of 35 NB models (7 criteria × 5 features evaluated) were developed (Figure 3.5). The hyperparameter values were optimised via grid search and evaluated using 10-fold cross-validation.

Support Vector Machine

A linear support vector classifier (linear SVC), an SVM with a linear kernel, was used for evaluation. Similar to the implementation of the NB models, a classifier was trained for each criteria with the same set of features totalling 35 SVM models. The hyperparameter values were also optimised and evaluated using grid search and 10-fold cross-validation.

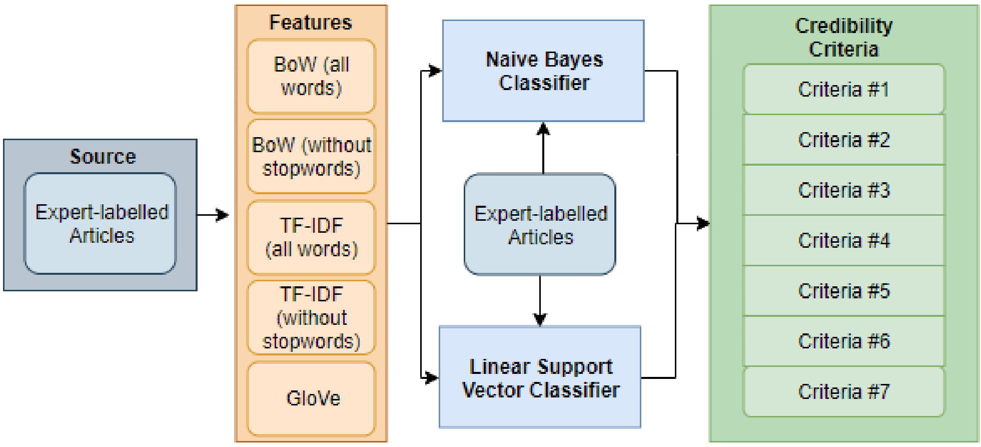


Figure 3.5: Overview of the NB and SVM models implemented.

Quasi-Recurrent Neural Network

In addition to using a QRNN for constructing the fine-tuned and general LM, another variant was implemented as a classifier. The architecture for the classifier was adapted from the experiments conducted by Bradbury et al. [13] for sentiment classification. From this model, the number of units within each layer was increased to 2500 and the embedding size was changed to 150. The model was trained for up to 20 epochs with a batch size of 100, however the model began to converge after only 5 epochs. In contrast to the NB and SVM models, 21 QRNN models (7 classifier × 3 LM variants) were created (Figure 3.6).

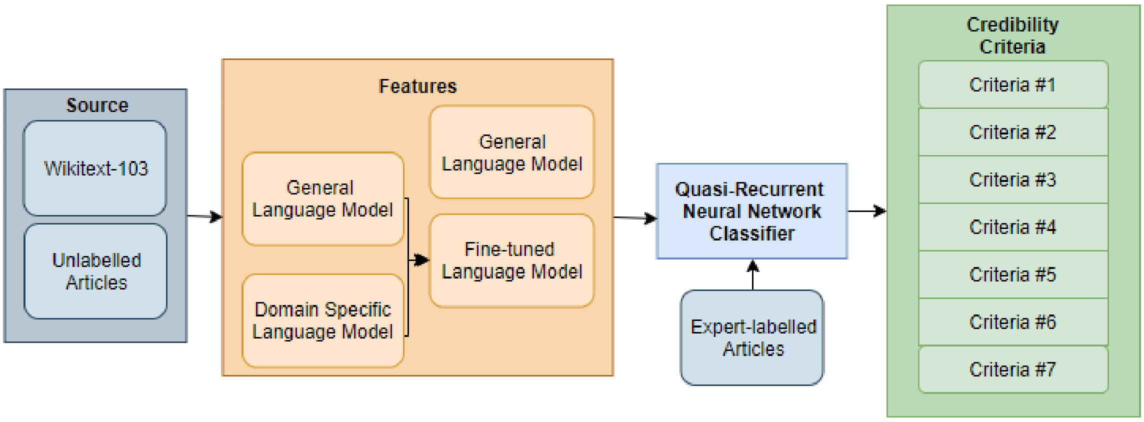


Figure 3.6: Overview of the QRNN models implemented.

3.4 Experiments

Considering the context of this project, other factors such as the size of the model and its training speed must be considered in addition to its capability on correctly producing the correct labels to determine the feasibility of deploying the model for a real-world application.

3.4.1 Performance

Two experiments have been conducted to evaluate the performance of a model. The credibility classification experiment is designed to determine the model's ability to produce the correct label on a criteria-level allowing for a more detailed insight on the reasoning for a particular article's credibility score. Whereas the low credibility identification experiment is designed to determine the model's ability to correctly determine whether an article is considered to have low credibility or not for situations where the detailed insight is not required.

Credibility Classification Experiment

The credibility classification experiment is a set of binary classification tasks which aims to determine a model's ability in correctly identifying whether an article either satisfies or doesn't satisfy a certain criteria. For each criteria, the respective NB and SVM model is employed along with the QRNN model to classify whether the article satisfies the criteria or not. The correctness of the model was then evaluated by measuring the micro averaged f1-scores via 10-fold cross-validation, which is the harmonic mean of the precision and recall of a model's predictions. This experiment attempts to assess the feasibility in providing insight on the factors that attributed to the credibility of an article in addition to identifying the model(s) that best classify a specific criteria.

Low Credibility Identification Experiment

The Low Credibility Identification Experiment was designed to evaluate the capability of a model to identify low credibility articles which have been defined to be articles with a credibility score lower than 3. The expert-labelled articles were then separated into two classes based on this condition and the best performing model(s), determined from the results of the credibility classification experiment, was then applied and evaluated using the calculated micro averaged f1-scores.

The classification of low credibility articles was then carried out following two methodologies. The first methodology involves the calculation of the article's credibility score via the classification of each criteria, resulting in a total of seven binary classification tasks. The second methodology utilises the separated expert-labelled articles to construct a new binary label consisting of the labels 'Credible' for articles that had a credibility score greater than or equal to 3 and a 'Not Credible' label for the remaining articles. This method results in only a single binary classification where the model is trained to classify an unseen article as one of the aforementioned labels rather than carrying out the classification of each credibility criteria.

3.4.2 Training

The training time for a model is another factor that must be considered when determining the feasibility of deploying and using a model on a real-world application. This is because the model will require to be constantly re-trained when additional article samples are incorporated as the model encounters new and never before seen information. To ensure consistent results, Amazon Web Service's EC2 instances [1] have been used to act as an isolated environment for timing the training of the NB, SVM, QRNN classifiers and also the creation of the LM. The reported training times in Section 4.2 are for the best performing NB and SVM-based models, which were trained within a c5.2xlarge instance utilising all CPU cores. The reported training times for the QRNN classifier and LM were obtained using the P3 instances and utilising only a single K80 GPU.

3.4.3 Storage

To minimise the frequency in which a model will have to be re-trained or re-created, storing the model for later use is required. Since the entire implementation was completed using Python, the Python module *pickle* was used to store the models externally. Pickle serialises any Python object into a byte stream allowing a model to be persistently stored. The NB and SVM-based models were stored using pickle and the size of the resulting byte stream was then measured. The QRNN classifier and LM were stored using PyTorch's inbuilt save function, that acts as a wrapper over the *pickle* module which provides storage optimisations for PyTorch objects. The resulting byte stream of the model is then recorded and reported in Section 4.3.

Chapter 4

Results

In this chapter I detail the results of the experiments described in the previous chapter. These are the results that come from testing the performance, training time, and storage requirements of the models in the classification task of predicting whether a document extracted from a web page satisfies each of the seven credibility criteria. The classifiers include the proposed QRNN-based deep learning approach as well as classifiers trained using NB and SVM methods.

4.1 Performance

4.1.1 Credibility Classification

I trained and tested the performance of NB and SVM classifiers in combination with two feature representations, BoW and TF-IDF, implementations of which as described in the prior chapter. For each criteria, all combinations of classifiers and feature representations was then trained to perform a binary classification task that aimed to predict whether a given article satisfied that criteria. This produced a combined total of 70 NB-based and SVM-based models.

The performance of the models varied considerably across all criteria (Table 4.1). When considering both the NB and SVM based approaches, the models had the most difficulty in classifying criteria 6 producing the lowest micro averaged f1-score of 0.71, with the best performing model for this criteria being an SVM classifier paired with TF-IDF (stopwords removed) who achieved a score of 0.77. The models were most successful in labelling criteria 3 which had a micro averaged f1-score of 0.85 with the top performing model being a tie between the NB and SVM classifier paired with BoW (stopwords) with a score of 0.90.

Across all criteria, the SVM with TF-IDF (stopwords removed) was the best-performing of the models, followed by the NB classifier using BoW with stopwords removed. This shows that the removal of stopwords is beneficial in improving model performance which is consistent with other research done for text classification tasks [*cite*].

The performance of the models may have highly been influenced by the unbalanced nature of the training data as the measured scores are proportionate to the imbalance of the label. This is most evident in the model performance for criteria 3, the most imbalanced label as shown in Figure 3.2, where the models had the highest success in classifying.

Criteria		**BoW** **(all words)**	**BoW** **(stopwords removed)**	**TF-IDF** **(all words)**	**TF-IDF** **(stopwords removed)**	**GloVe**
Criteria 1: **Use of evidence**	NB	0.79	0.85	0.79	0.79	0.82
	SVM	0.82	0.78	0.80	**0.86**	0.60
Criteria 2: **Based on objective research**	NB	0.70	0.79	0.74	0.80	0.78
	SVM	0.70	0.80	0.70	**0.85**	0.77
Criteria 3: **Explains limitations and uncertainties**	NB	0.86	**0.90**	0.87	0.89	0.89
	SVM	0.86	**0.90**	0.84	0.89	0.63
Criteria 4: **No sensationalist or exaggerated language**	NB	0.84	0.84	0.80	0.81	0.82
	SVM	0.80	**0.87**	0.84	0.86	0.61
Criteria 5: **Balanced information**	NB	0.79	**0.80**	0.79	0.77	0.78
	SVM	0.68	0.78	**0.80**	**0.80**	0.59
Criteria 6: **Simple Language**	NB	0.75	0.74	0.62	0.69	0.68
	SVM	0.75	0.72	0.68	**0.77**	0.67
Criteria 7: **Statement of funding or conflicts of interest**	NB	0.82	0.80	0.80	0.81	0.83
	SVM	**0.86**	0.81	0.82	0.82	0.81
Average Performance	NB	0.792	0.820	0.773	0.794	0.800
	SVM	0.799	0.809	0.783	**0.836**	0.669

Table 4.1: Micro-averaged f1-Scores of the NB and SVM based models. The best performing model(s) for a given criteria and overall best performing model is highlighted in bold.

A similar approach was also taken for the QRNN classifiers, which were trained and tested in combination with two LM variants, a general LM and a fine-tuned LM, producing a total of 14 QRNN-based models.

The QRNN-based models performed poorer across all criteria compared to the other models (Table 4.2), relative to the NB and SVM based models. The use of the fine-tuned LM with the QRNN classifier consistently reported an improved performance over the models that used the general language model for each criteria.

Using the best classifiers for each criteria, I found that it was possible to predict whether or not an online article satisfied each of the 7 credibility criteria. The results suggest that while it may not be feasible to robustly predict all of the individual criteria, the classifiers would be useful in predicting the credibility score, defined by the number of criteria satisfied. This idea is expanded on in the following experiment.

Criteria		**General LM**	**Fine-tuned LM**
Criteria 1: **Use of evidence**	QRNN	0.38	**0.42**
Criteria 2: **Based on objective research**	QRNN	0.36	**0.39**
Criteria 3: **Explains limitations and uncertainties**	QRNN	0.41	**0.44**
Criteria 4: **No sensationalist or exaggerated language**	QRNN	0.37	**0.40**
Criteria 5: **Balanced information**	QRNN	0.36	**0.41**
Criteria 6: **Simple Language**	QRNN	0.38	**0.41**
Criteria 7: **Statement of funding or conflicts of interest**	QRNN	0.39	**0.43**
Average Performance	QRNN	0.379	**0.414**

Table 4.2: Micro-averaged f1-Scores of the QRNN-based models. The best performing model(s) for a given criteria and overall best performing model is highlighted in bold.

4.1.2 Low Credibility Identification

I tested and employed three different approaches for constructing a model on the identification of low credibility articles. As described in Section 3.4.1 articles with a credibility score that is less than 3 are deemed to be untrustworthy. The three approaches are referred to as the single model approach, ensemble approach and binary classification approach. Where the single model and ensemble approach are variants of the first methodology.

The low credibility experiment was utilised to test three different approaches inspired from the results in the previous section. As described in Section 3.4.1 articles with a credibility score less than 3 are deemed to be untrustworthy. Based on this definition, there are three main methods that can be used to determine whether an article is untrustworthy.

The first method, the single model approach, utilises the overall best performing model. The best performing model is defined to be the model with the highest average performance across Table 4.1 and Table 4.2. Thus, the model used to evaluate the single model approach will be SVM with TF-IDF scores (stopwords removed).

Model	**Micro-averaged Precision**	**Micro-averaged Recall**	**Micro-averaged f1-Score**	**Average Predicted Score Differential (Correctly Labelled)**	**Average Predicted Score Differential (Incorrectly Labelled)**
SVM + TF-IDF (stopwords removed)	0.89	0.89	0.89	0.52	2.14

Table 4.3: Performance of low credibility identification task via an ensemble approach.

In contrast to the average performance of the model reported in Table 4.1 having a micro-averaged f1-score of 0.836, it is more able to successfully classify low credibility articles, with an improvement in performance of 8.85%. This is due to the increased leeway with regards to the condition for producing a correct classification as the model is only required to predict a credibility score between 0 and 2 for untrustworthy articles and 3 to 7 for trustworthy articles.

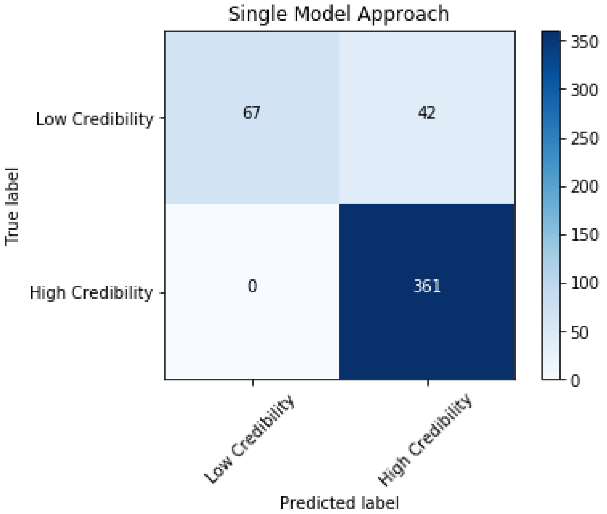


Figure 4.1: Performance of single model approach after 10-fold cross validation.

The second method, referred to as the ensemble approach, which leverages the best performing models for each criteria (Table 4.4). The ensemble classifier produced the highest performance results of the three classifiers (Table 4.5) with a micro-averaged f1-score of 0.91, an improvement of 7.82% from its average performance.

Criteria	**Model**
Criteria 1	SVM + TF-IDF (stopwords removed)
Criteria 2	SVM + TF-IDF (stopwords removed)
Criteria 3	SVM + BoW (stopwords removed)
Criteria 4	SVM + BoW (stopwords removed)
Criteria 5	SVM + TF-IDF (stopwords removed)
Criteria 6	SVM + TF-IDF (stopwords removed)
Criteria 7	SVM + BoW (all words)

Table 4.4: Architecture of the ensemble model.

Model	**Micro-averaged Precision**	**Micro-averaged Recall**	**Micro Averaged f1-Score**	**Average Predicted Score Differential (Correctly Labelled)**	**Average Predicted Score Differential (Incorrectly Labelled)**
Ensemble	0.91	0.91	0.91	0.55	2.0

Table 4.5: Performance of low credibility identification task via an ensemble approach.

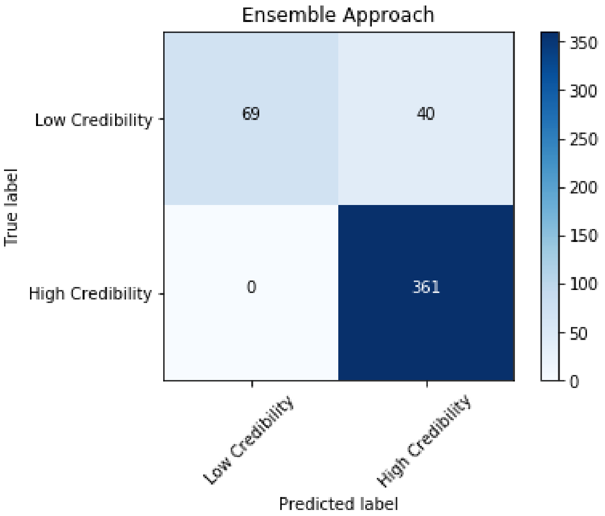


Figure 4.2: Performance of ensemble classifier after 10-fold cross validation.

Binary Classification Approach

Model	Micro-averaged Precision	Micro-averaged Recall	Micro Averaged f1-Score
SVM + TF-IDF (stopwords removed)	0.61	0.61	0.61

Table 4.6: Performance of low credibility identification task via a binary classification approach.

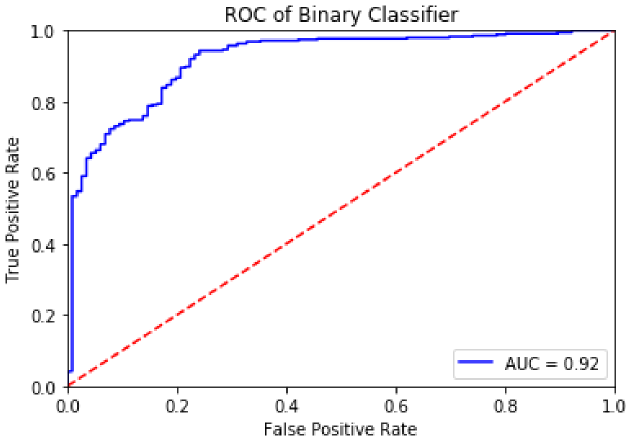


Figure 4.3: Performance of binary classification approach after 10-fold cross validation.

4.2 Training Time

The training times reported for each model includes the time required to perform the feature-specific preprocessing requirements in addition to the training time of the classifier itself. Information about training times can provide additional support for selecting one approach over others, especially in environments where ongoing training may be needed to update tools used in practice. Since the size and complexity in constructing count-based features such as BoW and TF-IDF (Table 4.7), especially when limited to an upper bound, is relatively lower when compared to the construction of statistical-based features such as a language model (Table 4.8) and relatively smaller than a pre-trained feature such as GloVe (Table 4.9), it is expected that the training times for models that utilise the count-based features are lower.

Model	**Training Time (seconds)**
NB + BoW (all words)	5.20
NB + BoW (stopwords removed)	**4.86**
NB + TF-IDF (all words)	5.05
NB + TF-IDF (stopwords removed)	5.05
SVM + BoW (all words)	5.91
SVM + BoW (stopwords removed)	4.87
SVM + TF-IDF (all words)	5.35
SVM + TF-IDF (stopwords removed)	5.58

Table 4.7: Average training time of models using count-based features.

Model	**Epoch Completion (minutes)**
QRNN + General LM	**271**
QRNN + Fine-tuned LM	292

Table 4.8: Average training time of models using LMs.

Model	**Training Time (seconds)**
NB + GloVe	3,628.82
SVM + GloVe	**3,558.26**

Table 4.9: Average training time of models using pre-trained GloVe feature.

4.3 Model Storage Requirements

The reported sizes for each of the models are based on the model implmentations in Section 4.1.1. Due to the model sizes staying relatively unchanged when trained for all criteria (e.g. The size of the NB + BoW (all words) model stayed practically unchanged when trained on each criteria), the aggregated model size of a given model trained on each criteria has been reported.

Models that utilised count-based features (Table 4.10) unsurprisingly had the lowest storage requirements. This is due to its relatively simplistic representation of text in conjunction with the enforced limit on the number of words that are stored. On the other hand, models that utilised the pre-trained GloVe feature (Table 4.11) has a substantially higher storage requirement due to the feature having a substantially larger vocabulary and higher dimensionality. Whilst this is also explains the storage requirements of models that used LMs (Table 4.12), the LMs used were not trained as extensively as the pre-trained GloVe.

Model	**Size (MB)**
NB + BoW (all words)	11.10
NB + BoW (stopwords removed)	**11.00**
NB + TF-IDF (all words)	13.20
NB + TF-IDF (stopwords removed)	13.20
SVM + BoW (all words)	7.91
SVM + BoW (stopwords removed)	**7.87**
SVM + TF-IDF (all words)	10.00
SVM + TF-IDF (stopwords removed)	7.88

Table 4.10: Aggregated size of count-based models.

Model	**Size (MB)**
NB + GloVe	25,410
SVM + GloVe	25,340

Table 4.11: Aggregated size of GloVe-based models.

Model	**Size (MB)**
QRNN + General LM	**3,612.70**
QRNN + Fine-tuned LM	4,193.35

Table 4.12: Aggregated size of LM-based models.

Chapter 5

Discussion

This chapter examines the feasibility of the implemented and evaluated models in the context of assessing the credibility of vaccine-related articles. The structure of this chapter is as follows.

The reliability of the results reported in Chapter 4, is firstly evaluated by examining the consistency of the general trends that emerged from the experiments via the comparison of results from prior literature on text classification tasks (Section 5.2).

Feasibility of the models is then evaluated by determining the trade-offs associated between the outcome measures (as identified in Section 3.4) and how the priority or importance of each of these measures change depending on the type of application being created (Section 5.3).

Finally, the significance of the obtained results in the context of real-world applications is examined by investigating potential applications of these models based on the advantages and limitations of the top performing models (Section 5.4).

5.1 Related Work

5.1.1 title

5.2 Consistency of Results

Since the classification task performed by the models involve the classification of a novel set of labels (credibility criteria), it is unsuitable to directly compare the performance of the models with any prior work. However, by identifying trends from the reported results, it is possible to determine whether these trends are consistent with the trends found on similar classification tasks from existing literature and thus extrapolate the reliability of the results.

From the reported results, major trends that have been identified are:

1. Utilisation of BoW were more successful overall when used in conjunction with NB classifiers rather than SVM classifiers.
2. Inversely, TF-IDF scores were more effective when paired with SVM classifiers rather than with NB classifiers.
3. Classical ML models (NB and SVM based models) consistently outperformed DL models (QRNN-based models) on classification tasks with limited training and testing data.
4. The process of fine-tuning a LM in order to learn task specific features consistently improved the model's performance.

The sentiment classification task using the subjectivity dataset [37] evaluated by Wang et al. [50] and the product-review classification task performed by Pranckevičius et al. [42] attained results that aligned with the first major trend described above. Both works presented results that showed better performance when using BoW with a NB classifier over a SVM classifier. The work conducted by Wang et al. involved evaluating a linear SVM and multinomial NB classifier in combination with a unigram BoW model to perform a binary sentiment analysis task on movie reviews ('thumbs up', 'thumbs down') [37]. Similarly, Pranckevicius et al., performed a multi-class classification task, that aimed to predict the resulting product review score (ranging from 1 to 5) from reviews on Amazon [42]. For the sentiment analysis task, the NB-based model had an improved performance of 5.5% over the SVM-based model and the product-review classification task reported an improvement of 2.4% by the NB-based model over the SVM-based model. These results align with the measurements obtained from the classification experiment reported in Section 4.1.1 which showed the NB-based model having a higher average performance by a margin of 0.25%.

The diminishing difference in performance between the NB-based models and the SVM-based models relative to the aforementioned literature may have a relation to the increasing number of classes being classified in these tasks. This is somewhat supported by the results obtained by Hassan et al. [22] who evaluated NB and SVM based models using the 20 Newsgroups dataset [34] which consists of 20 classes where the baseline performance of the SVM-based models outperformed the NB-based models by 25%. This however does not conclusively explain the diminishing difference as Hassan et al. utilised domain-specific information extracted from as features for the construction of their models. The common factor in regards to the construction of the models resides within the preprocessing phase where only the stopwords of the text were removed.

The following paragraph discusses the consistency of TFIDF working better with SVMs rather than NBs

[——————]

The following paragraph discusses the consistenctу of ML models outperforming DL models for text classification tasks with limited data

[——————]

The following paragraph discusses the consistent results of fine-tuned LMs outperforming general-domain LMs

While the performance of the QRNN-based models was lower than what was initially expected, this can be theorised to be due to the relatively low number of vaccine-related training samples utilised during the training of the fine-tuned LM. When compared to the work conducted by Howard et al. [24] for the task of question classification, the number of training samples used to construct the discriminative fine-tuned LM was 5,500 which only improved the model's performance by 0.3%.

5.3 Model Feasibility

Notes about what to talk about:

- Discuss the trade-offs between optimising the performance, training time and storage requirements and the factors that affect these measures.

- What it means to be feasible e.g. it is application dependent and depending on the application some factors might be more important than others such as having a higher true negative rate can be alright for some tasks and that having a high false positive rate is usually always bad.

5.4 Real World Applications

This is where I'll be talking about the performance of constructing and testing the low credibility classifiers So once the credibility of the results have been verified and also what the feasibility of a model entails, I can then talk about:

- Possible applications of the models e.g. web browser extension, twitter bot, article filtering system and the differing priorities they have on the for outcome measures (accuracy, training time and storage requirements)

Chapter 6

Conclusions and Future Work

This section will be completed after the fulfilment of the proposed work detailed in Section ??

6.1 Future Work

6.2 Conclusions

Appendix A

Exmaple

A.1 Overview

This is an example entry in the appendix

Bibliography

[1] "Amazon EC2 Instance Types – Amazon Web Services (AWS)." [Online]. Available: https://aws.amazon.com/ec2/instance-types/

[2] "AUTISM & VACCINES: IS THERE A LINK?" [Online]. Available: https://www.learntherisk.org/autism/

[3] "DISCERN - The DISCERN Instrument." [Online]. Available: http://www.discern.org.uk/discern{_}instrument.php

[4] "Our Review Criteria - HealthNewsReview.org." [Online]. Available: https://www.healthnewsreview.org/about-us/review-criteria/

[5] "Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs – WildML." [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[6] "Understanding the kernel trick. – Towards Data Science." [Online]. Available: https://towardsdatascience.com/understanding-the-kernel-trick-e0bc6112ef78

[7] C. C. Aggarwal and C. Zhai, "A SURVEY OF TEXT CLASSIFICATION ALGORITHMS," 2012. [Online]. Available: http://alias-i.com/lingpipe/

[8] M. A. Aizerman, E. A. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, June 1964.

[9] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," Tech. Rep., 2017. [Online]. Available: http://en.wikipedia.org/wiki/Statistics

[10] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," Tech. Rep., 2000. [Online]. Available: www.elsevier.com/locate/jmicmeth

[11] J. M. Batchelor and Y. Ohya, "Use of the DISCERN Instrument by Patients and Health Professionals to Assess Information Resources on Treatments for Asthma and Atopic Dermatitis," Tech. Rep., 2009. [Online]. Available: www.jsaweb.jp!

[12] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python.* O'Reilly Media Inc., 2009. [Online]. Available: https://www.nltk.org/book/

[13] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-Recurrent Neural Networks," *International Conference on Learning Representations (ICLR 2017)*, 2017.

[14] J. Browniee, "A Gentle Introduction to Transfer Learning for Deep Learning," 2017. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[15] D. C. Burgess, M. A. Burgess, and J. Leask, "The MMR vaccination and autism controversy in United Kingdom 1998-2005: Inevitable community outrage or a failure of risk communication?" *Vaccine*, vol. 24, pp. 3921–3928, 2006. [Online]. Available: https://ac.els-cdn.com/S0264410X06002076/1-s2.0-S0264410X06002076-main.pdf?{_}tid=46d1dda6-f576-4f5e-ad53-d550f1cd9990{&}acdnat=1534726962{_}1b237371d8bb916694f34f0f951c84bc

[16] E. Cambria, S. Poria, A. Gelbukh, I. P. Nacional, and M. Thelwall, "AFFECTIVE COMPUTING AND SENTIMENT ANALYSIS Sentiment Analysis Is a Big Suitcase," Tech. Rep., 2017. [Online]. Available: www.computer.org/intelligent

[17] N. Cantey Banasiak and M. Meadows-Oliver, "Journal of Asthma and Allergy Dovepress Evaluating asthma websites using the Brief DISCERN instrument," *Journal of Asthma and Allergy*, pp. 10–191, 2017. [Online]. Available: http://dx.doi.org/10.2147/JAA.S133536

[18] C. Cipriani, J. Pepe, S. Minisola, and ·. E. M. Lewiecki, "Adverse effects of media reports on the treatment of osteoporosis," *Journal of Endocrinological Investigation.* [Online]. Available: https://doi.org/10.1007/s40618-018-0898-9

[19] R. Collobert, J. Weston, J. Com, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011. [Online]. Available: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf

[20] A. Conneau, H. Schwenk, Y. Le Cun, and L. Loıc Barrault, "Very Deep Convolutional Networks for Text Classification," 2017. [Online]. Available: https://arxiv.org/pdf/1606.01781.pdf

[21] V. Hall, E. Banerjee, C. Kenyon, A. Strain, J. Griffith, K. Como-Sabetti, J. Heath, L. Bahta, K. Martin, M. McMahon, D. Johnson, M. Roddy, D. Dunn, and K. Ehresmann, "Measles Outbreak — Minnesota April–May 2017," *MMWR. Morbidity and Mortality Weekly Report*, vol. 66, no. 27, pp. 713–717, jul 2017. [Online]. Available: http://www.cdc.gov/mmwr/volumes/66/wr/mm6627a1.htm

[22] S. Hassan, M. Rafi, and M. Shahid Shaikh, "Comparing SVM and Naïve Bayes Classifiers for Text Categorization with Wikitology as knowledge enrichment," Tech. Rep. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1202/1202.4063.pdf

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[24] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," Tech. Rep., 2018. [Online]. Available: http://nlp.fast.ai/ulmfit.

[25] F. Informatik, "UNIVERSIT AT D ORTMUND Text Categorization with Support Vector Machines: Learning with Many Relevant F eatures Thorsten Joachims," Tech. Rep., 1997. [Online]. Available: https://www.cs.cornell.edu/people/tj/publications/joachims{_}97b.pdf

[26] J. Kaicker, V. Borg Debono, W. Dang, N. Buckley, and L. Thabane, "Assessment of the quality and variability of health information on chronic pain websites using the DISCERN instrument," Tech. Rep., 2010. [Online]. Available: http://www.biomedcentral.com/1741-7015/8/59

[27] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Tech. Rep. [Online]. Available: http://nlp.stanford.edu/sentiment/

[28] V. Korde and C. N. Mahender, "TEXT CLASSIFICATION AND CLASSIFIERS: A SURVEY," *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 3, no. 2, 2012. [Online]. Available: http://aircconline.com/ijaia/V3N2/3212ijaia08.pdf

[29] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning." Springer, Berlin, Heidelberg, 1999, pp. 319–345. [Online]. Available: http://link.springer.com/10.1007/3-540-46805-6{_}19

[30] K. Matsoukas, S. Hyun, L. Currie, M. P. Joyce, J. Oliver, S. Patel, O. Velez, P.-Y. Yen, and S. Bakken, "Expanding DISCERN to create a tool for assessing the quality of Web-based health information resources," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, p. 1048, 2008.

[31] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," Tech. Rep., 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.9324{&}rep=rep1{&}type=pdf

[32] S. Merity, "The wikitext long term dependency language modeling dataset," 2016. [Online]. Available: https://einstein.ai/research/the-wikitext-long-term-dependency-language-modeling-dataset

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," Tech. Rep. [Online]. Available: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[34] T. Mitchell, "Twenty Newsgroups Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

[35] C. Nogueira, D. Santos, and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts," Tech. Rep. [Online]. Available: http://www.aclweb.org/anthology/C14-1008

[36] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," 2009. [Online]. Available: http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95

[37] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," Tech. Rep., 2004. [Online]. Available: www.cs.cornell.edu/people/pabo/movie-

[38] K. Pasupa, "A Comparison between Shallow and Deep Architecture Classifiers on Small Dataset," 2016.

[39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[41] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[42] T. PRANCKEVIČIUS and V. Marcinkevicius, "Comparison of Naïve Bayes , Random Forest , Decision Tree , Support Vector Machines , and Logistic Regression Classifiers for Text Reviews Classification," 2017. [Online]. Available: https://www.semanticscholar.org/paper/Comparison-of-Na{\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{\OT1\i\global\mathchardef\accent@spacefactor\spacefactor}\accent127\OT1\i\egroup\spacefactor\accent@spacefactor}ve-Bayes-{%}2C-Random-Forest-{%}2C-Tree-{%}2C-PRANCKEVI{\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{C\global\mathchardef\accent@spacefactor\spacefactor}

\accent20C\egroup\spacefactor\accent@spacefactor}IUS-Marcinkevicius/e853f8827a2b9c596296de296b482e1175f9713f?navId=similar-papers

[43] R. Rosenfeld, "TWO DECADES OF STATISTICAL LANGUAGE MODELING: WHERE DO WE GO FROM HERE?" Tech. Rep., 2000. [Online]. Available: https://www.cs.cmu.edu/{~}roni/papers/survey-slm-IEEE-PROC-0004.pdf

[44] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," Tech. Rep. Cohen, 1998. [Online]. Available: http://research.microsoft.com/en-us/um/people/horvitz/junkfilter.htm

[45] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," Tech. Rep. [Online]. Available: http://nlp.stanford.edu/

[46] R. Som and N. P. Gunawardana, "Internet chemotherapy information is of good quality: assessment with the DISCERN tool," *British Journal of Cancer*, vol. 107, no. 2, pp. 403–403, jul 2012. [Online]. Available: http://www.nature.com/articles/bjc2012223

[47] S. Sommariva, C. Vamos, L. U.-L. Mantzarlis, Alexios Dao, and D. Martinez Tyson, "Spreading the (Fake) News: Exploring Health Messages on Social Media and the Implications for Health Professionals Using a Case Study," *American Journal of Health Education*, vol. 49, no. 4, pp. 246–255, jul 2018. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/19325037.2018.1473178

[48] "Germany investigating unprecedented spread of fake news online — World news — The Guardian," 2017. [Online]. Available: https://www.theguardian.com/world/2017/jan/09/germany-investigating-spread-fake-news-online-russia-election

[49] S. Vosoughi, D. Roy, and S. Aral, "SI:The spread of true and false news online," Tech. Rep. 6380, 2018. [Online]. Available: http://science.sciencemag.org/content/sci/359/6380/1146.full.pdf{%}0Ahttp://www.sciencemag.org/lookup/doi/10.1126/science.aap9559

[50] S. Wang and C. D. Manning, "Baselines and bigrams: simple, good sentiment and topic classification," pp. 90–94, 2012. [Online]. Available: https://dl.acm.org/citation.cfm?id=2390688

[51] J. Weston, S. Bengio, and N. Usunier, "WSABIE: Scaling Up To Large Vocabulary Image Annotation," Tech. Rep. [Online]. Available: http://torch5.sourceforge.net/

[52] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," Tech. Rep. [Online]. Available: http://veredshwartz.blogspot.sg.

[53] D. Zeraatkar, M. Obeda, J. S. Ginsberg, and J. Hirsh, “The development and validation of an instrument to measure the quality of health research reports in the lay media,” *BMC Public Health*, vol. 17, no. 1, p. 343, dec 2017. [Online]. Available: http://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-017-4259-y

[54] X. Zhang, J. Zhao, and Y. Lecun, “Character-level Convolutional Networks for Text Classification,” 2015. [Online]. Available: https://arxiv.org/pdf/1509.01626.pdf

[55] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, “A C-LSTM Neural Network for Text Classification,” 2015. [Online]. Available: https://arxiv.org/pdf/1511.08630.pdf