Challenges in Opinion Mining: an Analysis using Vaccine Hesitancy

Oleg Zhilin, School of Computer Science

McGill University, Montreal

August, 2021

A thesis submitted to McGill University in partial fulfillment of the

requirements of the degree of Master of Science

Abstract

Opinion mining as a field aims to measure public opinion on a given topic in order to facilitate decision-making. We argue that more emphasis should be placed on opinion detection to generate better datasets for downstream tasks such as sentiment analysis. We demonstrate the complexity of this task by applying it to the problem of detecting opinions on Twitter regarding the COVID-19 vaccine. We find that a high proportion of tweets contain an opinion, but comparatively few mention concrete effects of the vaccine.

Abrégé

Le domaine du «Opinion Mining» cherche à quantifier l'opinion publique sur un sujet particulier afin d'améliorer la prise de décisions. Nous affirmons qu'il faudrait mettre davantage l'accent sur la détection des opinions pour créer de meilleurs banques de données pour des tâches en aval comme l'analyse de sentiments. Nous démontrons la complexité de cette tâche en l'appliquant au problème de la détection d'opinions reliés au vaccin contre la COVID-19 sur Twitter. Nous trouvons qu'une proportion élevée de gazouillis contiennent une opinion, mais peu mentionnent un effet concret du vaccin.

Acknowledgements

I would first like to thank my supervisor, Professor Derek Ruths, for all the assistance and mentorship he has provided. Without your support I would not have been able call myself a researcher today. Your guidance has been instrumental in shaping how I carry out my work and your patience is the main reason this thesis is legible.

I would like to thank the NDL lab members for their encouragement. You were always generous with your time and made my graduate experience immeasurably better than what I thought it could be.

I would also like to thank my parents and Moyen Monsieur Mark for their love, advice, dogsitting and a million other things that would be longer than the thesis if enumerated.

Finally, I would like to thank my partner Kelly for her support and for showing me what real work ethic looks like and my dog Chaz for sleeping through the entire thesis-writing process.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction

Social media platforms have become an important form of communication, news consumption [47] and play a heavy role in shaping our worldview [4]. This is highlighted by the fact that Twitter contains 199 million daily active users as of 2020 [79]. The tweets on the platform range from casual conversation to debating and reporting on current events. For many highly active users, it is primarily used as a "gateway to other news" [15] and during active crises, Twitter is an important tool to "redistribute official information and to provide eyewitness reports from people close to the crisis events" [50].

The open and immediate nature of social media platforms like Twitter generates interest in running automated natural language processing (NLP) algorithms to gain insight into the what the platform's users are discussing. One particularly popular task, opinion mining, involves extracting opinions from tweets about a subject. Such systems aim to

enable better decision making for policymakers, government officials, and anyone else who communicates with the general public.

In the literature, opinion mining is defined as "the automatic processing of documents to detect opinion expressed therein" [68]. This is an intuitive definition, but it leaves a lot of room for distinct operationalizations. In particular, the three most common tasks used to address all or part of the high level definition are opinion mining, sentiment analysis, and subjectivity analysis (also known as subjectivity identification or opinion identification). At a high level, it is often considered appropriate to treat opinion mining and sentiment analysis as a unified body of work [53].

In practice, this causes some confusion in the literature surrounding the term *opinion*. Specifically, it is sometimes used as a synonym for *sentiment* [45]. In order to avoid such ambiguity, in this thesis we will use the definition provided by Kim and Hovy [33] which states that an opinion is: "a quadruple [Topic, Holder, Claim, Sentiment] in which the Holder believes a Claim about the Topic, and in many cases associates a Sentiment, such as good or bad, with the belief". This is not the only available operationalization in the literature, but for our purposes the key point is that a sentiment is not a "personal interpretation of information" [45] but rather a social construct that is prompted by emotions [45].

It follows that a successful system should extract all four components of the definition from a document. Specifically, we argue that "opinion mining" systems using the task labels "positive", "negative", and "neutral" to model opinion are oversimplifying the problem by exclusively extracting a measure of sentiment and (sometimes) topic. This

is insufficient in an applied setting because it rarely provides enough new information about the underlying phenomenon to assist in decision making.

The second key ability of an applied opinion mining system is to predict the presence of an opinion. Typically [31, 37], the problem of opinion identification is carried out implicitly as part of the data curation step of a sentiment analysis task. In many cases, documents without opinions are filtered out manually [82]. In a production setting, this would require human intervention before each document is supplied to the algorithm, which is unrealistic. In other cases [17], a heuristic such as keyword filters is used to isolate documents containing opinions. This may be an appropriate choice, but it is rarely part of the model selection steps because it filters out valuable training data. As a result, it is difficult to measure how representative are the predicted opinions of the overall discourse on the subject of interest.

We argue that both of the implicit and heuristic approaches introduce errors because they understate the difference between opinion identification and the task of interest such as sentiment analysis, which should focus solely on a measure of polarity. A better way to approach the task would be to have one component to identify the presence of an opinion followed by another to predict the necessary components of the opinion quadruple. The potential benefit of reinforcing this separation has been previously noted in the literature: "the problem of distinguishing subjective versus objective instances has often proved to be more difficult than subsequent polarity classification, so improvements in subjectivity classification promise to positively impact sentiment classification" [41].

In this thesis, we carry out an applied opinion identification exercise using COVID-19 vaccine data on Twitter in order to highlight the phenomenological insights it can

reveal as well as its benefits to a downstream task. The problem is motivated by the fact that the vaccination campaign during the COVID-19 pandemic has been met with resistance due to concerns regarding the vaccine's safety [14]. It is therefore in the interest of public health departments to understand and promptly address societal concerns in order to attain a vaccination rate that will be sufficient to achieve herd immunity. Given that public concerns related to a vaccine are likely to spread quickly on Twitter, having a system that characterizes their presence and prevalence would be of immense value. As the downstream task, we predict whether a tweet mentions a concrete effect of a COVID-19 vaccine.

In order to achieve this, we assemble a dataset of Canadian tweets related to the COVID-19 vaccine by searching for specific keywords within a nationally representative set of English tweets from 2020. The source dataset has previously been collected in real time during the pandemic based on a list of approximately 1.6 million confirmed-Canadian accounts [8].

Next, we annotate a portion of this dataset for the presence of an opinion and for mentions of concrete effects tied to the vaccine. We crowdsource the annotation using Amazon Mechanical Turk [1], then consider the impact of the obtained agreement measures on the downstream task and weigh the choice of building a training dataset by using majority labels versus only using cases with full annotation agreement.

Finally, we carry out the modeling exercise to predict whether tweets mentioning a COVID-19 vaccine contain an opinion. We use traditional machine learning algorithms such as Support Vector Machines (SVM) and Logistic Regression and experiment with

[1] https://www.mturk.com

a number of features used in the opinion mining literature. We also experiment with deep learning architectures as well as a number of different word embeddings as input features. We evaluate the algorithms by executing a grid search over models and their hyperparameters using different standard evaluation metrics applied over k-fold cross validation. We discuss the value and limitations of the resulting models and comment on how this translates to the real-world applicability of other applied opinion mining work.

1.2 Summary of Contributions

We summarize the contributions of this thesis as follows:

- We highlight some limitations of current applied opinion mining work and argue in favor of extracting the opinion detection task into a separate step. Most applied work has to implicitly solve the opinion detection problem along with the problem of interest. This is a challenge because, to the best of our knowledge, there is little recent literature providing benchmarks for various opinion detection models.

- We demonstrate the hardness of the opinion detection task by using a case study on vaccine hesitancy on Twitter. We curate a dataset and annotate a subset for the presence of opinions as well as the mention of effects. We find that obtaining annotator agreement for opinion detection is challenging because humans define the term differently.

- We experiment with various traditional and deep machine learning models that correspond to the state of the art of NLP techniques. We find that deep learning models outperform traditional algorithms under all metrics except weighted F1.

- We discuss model selection for opinion detection when separated from some downstream task and argue that weighted F1 is the best measure to use to produce a clean dataset. Based on our performance results, we find that using a simple logistic regression creates a better downstream dataset than deep learning models.

- Based on our annotation, we find that 67% of tweets contain an opinion while only 39% mention an effect. It seems that most conversations are using association to attach a positive or negative connotation to the vaccine.

1.3 Outline

The remainder of this thesis is organized as follows:

- Chapter 2 discusses prior work in opinion mining, stance detection, sentiment analysis, and the literature on differentiating the terminology used by the sub-problems. It is followed by background knowledge on dataset curation, annotation methodology and evaluation, traditional and deep machine learning text classification, as well as evaluation metrics used in this thesis.

- Chapter 3 argues for separating the tasks of opinion detection and mining. It also describes techniques and datasets that are related to the opinion detection problem.

- Chapter 4 describes the data collection process for the COVID-19 vaccination case study, the design of the crowdsourced annotation tasks, and reports agreement measures on the annotations.

- Chapter 5 carries out the modeling exercise using traditional and deep learning approaches and reports on the performance obtained under a number of evaluation metrics.

- Chapter 6 discusses the takeaways from the reported annotation agreements and their impact on the modeling exercise. It also compares prediction scores with similar applied modeling exercises and considers how the resulting models would be selected to create a quality downstream dataset.

- Chapter 7 concludes the thesis by summarizing the contributions and suggests future work along this line of research.

Chapter 2

BACKGROUND

2.1 Opinion Mining and Related Problems

Opinion mining uses NLP to extract the beliefs expressed in the documents under inspection. The results are invaluable for many applications ranging from marketing campaigns to government policy. Using the immense volume of data available on the Internet, the NLP community has produced a rich applied opinion mining and sentiment analysis literature across many social media platforms: Twitter [31, 40, 51, 60, 66, 77, 83], YouTube [48, 58, 69], Facebook [76], Reddit [1], and even traditional news media [59]. In addition, there are studies breaking down the difference between opinion and sentiment [45] as well as the various tasks that fall under the opinion mining umbrella [22, 53, 81]. In this section, we begin with a review of definitions and the literature surrounding the distinction between opinion mining, sentiment analysis, stance detection, and subjectivity identification. We follow with a review of the common concepts to each task as well as

the methods used to solve them. Finally, we review some of the general applied work in opinion mining as well as vaccine and COVID-19 specific studies.

2.1.1 Terminology and Task Definition

We begin by defining what is an opinion in order to understand the differences between various subtasks of opinion mining. While there are many options in the literature, the most complete separate the notion of sentiment from opinion. Munezero et al. [45] focus on five of the most common subjectivity terms found in the NLP literature: affect, feeling, emotion, sentiment and opinion. They compile many definitions for each term and we provide one for each term (with emphasis) in a way that highlights how they build on one another to form the concept of opinion:

- Affects: "*positive and negative evaluations* of an object, behavior, or idea with intensity and activity dimension" [78]
- Feelings: "affective phenomena to which we have *direct conscious access*" [39]
- Emotions: "affective manifestations to which we do not have direct conscious access, but which can be *inferred from behavioural clues*" [39]
- Sentiments: "social constructs of emotions that develop over time and are enduring" [45]. Critically, they emphasize that sentiments are *necessarily* targeted, unlike emotions.

- Opinions: "personal interpretations of information that may or may not be emotionally charged" [45]. The main difference is that sentiments are *possessed* while opinions are *expressed* [72].

Using these definitions, the authors point out that a positive or negative label corresponds to the optional affective component of an opinion representing an "emotional or sentimental expression" [45]. Such a taxonomy highlights a level of inconsistency within the NLP community, which blends various definitions together. As we proceed to review the various tasks related to opinion mining, we will provide a framing that is consistent with this clarified terminology.

We will follow the task hierarchy provided by Hemmatian and Sohrabi [22]. Alternate hierarchies do exist in the literature. For example: Yadollahi and Sohrabi [81] define opinion mining as a subset of sentiment analysis, but that does not make sense under our terminology because we define sentiment as an (optional) component of an opinion.

Opinion Mining

The first stated goal of opinion mining was to "process a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good)" [11]. This definition is similar to sentiment analysis and therefore merging the two has been encouraged [53]. As the two fields progressed together, definitions have evolved and opinion mining has often become a parent task to many others, including sentiment analysis.

Sentiment Analysis

The usage of sentiment within the NLP community first appeared in 2001 by Das and Chen [10], who predicted a positive or negative market sentiment using five different classifiers and combining their predictions with a voting scheme. Since then, the number of studies has grown rapidly, but the general goal has not changed. Definitions generally state that sentiment analysis involves predicting whether the "text expresses positive or negative (or sometimes neutral) opinion" [81]. With refinements in terminology, some researchers have argued that such a classification is a better proxy for emotion or affect than sentiment [45].

Stance Detection

A reasonable definition of stance detection states that it "focuses on identifying a person's standpoint or view toward an object of evaluation, either to be in favor of (supporting) or against (opposing) the topic" [3]. The main difference with sentiment analysis is that a target is available and the labels are typically "for", "against", and "neutral" instead of "positive", "negative", and "neutral".

In fact, many studies have shown that sentiment is not sufficient to model the polarity component of an opinion [2,73]. It is possible to be in favor of something without expressing any emotion that could be inferred. For example: a tweet saying "Life is sacred on all levels. Abortion does not compute with my philosophy" argues against the legalization of abortion without expressing sentiment [3].

Opinion Identification

Also referred to as subjectivity identification, this task is a necessary precursor to the others in a complete opinion mining system, though many models algorithms solve both at the same time by adding a label that signals that no opinion is present. This task simply involves assigning a "yes" or "no" label depending on whether or not a document contains at least one instance of an opinion [67].

Khatua et al. [32] identify 2 million subjective tweets in a discussion about nuclear reactors using three deep learning models. They trained their models using a gold standard dataset of 2308 tweets that obtained unanimous annotator agreement. They reached an accuracy of 80.7% using a Convolutional Neural Network (CNN).

2.1.2 Concepts and Methods

Common to most opinion mining tasks is the concept of *levels*. In descending order of granularity, we can classify documents, sentences, and aspects (or entities) [25]. Document-level opinion mining is the simplest task, but it assumes that the entire document contains a single opinion. Sentence-level eliminates that assumption but can be more challenging because the system will typically aim to handle context arising from surrounding sentences [46]. Tasks operating at the aspect-level, the most common being aspect-based sentiment analysis (ABSA), are the most challenging and require immense domain-specific datasets. Algorithms operating at all levels can use three methods to make their predictions: lexicon-based approaches, machine learning techniques, and hybrid methods (combining the first two) [22].

Lexicon-based algorithms use dictionaries of sentiments which assign a polarity to a set of words. For example, Taboada et al. [75] classify an Italian YouTube dataset using a dictionary capturing semantic orientation in the form of a polarity and a strength. They also incorporate the concepts of intensification ("this video is *very* good") and negation ("this video is "not" good"). The advantage of this method is that the dictionary can be inspected and criticized. The authors create their dictionary manually, but semi-automated methods that expand on a set of seed words have existed since 1997 [21].

Machine learning algorithms will typically use one of three learning approaches: supervised, unsupervised and semi-supervised. The most common option is a supervised architecture where the dataset has access to labels provided by human annotators. Severyn et al. [69] build such a system using Support Vector Machines (SVM) to predict the polarity of YouTube comments. Unlike lexicon-based methods, the algorithm does not get any input regarding what words or features of the text are relevant for prediction

Hajmohammadi et al. [20] use a graph-based semi-supervised approach to classify polarity of unlabelled documents in one language based on labelled documents in another. Such a method is promising because most languages do not have as many available corpora as English. In their case, the challenge is to try to match the performance of a model that is both trained and tested in the target language and they find that incorporating the intrinsic structure of the document increases classification accuracy in most cases.

Finally, Li and Liu [36] leverage the K-Means clustering algorithm to split a dataset of movie reviews encoded using a Term Frequency-Inverse document Frequency (TF-IDF) weighting method to encode documents. They find that their model obtains an accuracy of 77.17%-82%, which falls between that of a lexicon-based method (65.83%)

and a supervised model (77%-82%). In terms of operation, the model is only slightly slower than the lexicon-based approach. Moreover, it is significantly cheaper than the supervised method because it does not require human annotation.

2.2 Dataset Curation

For any NLP task we need a corpus of data that was at some point assembled, cleaned, and, if a supervised learning algorithm is used, annotated. Therefore, a proper data collection procedure should be able to answer the following questions before any NLP model is fit:

- Where did the data come from and what is it representative of?
- What kind of noise was identified and what steps were carried out to remove it?
- If we are annotating:
  - Can we trust the annotators?
  - How difficult is the task for humans?
  - How do we handle disagreements by high quality annotators in hard cases.

2.2.1 Data Collection and Cleaning

In the case of social media research, the data is collected either using an Application Programming Interface (API) or scraped directly from the web. In both cases, the text will likely have to resort to numerous *preprocessing* and *normalization* steps.

For example, Pandarachalil et al. [51] classified sentiment on Twitter but first had to remove URLs, hashtags, mentions. They then removed all contractions, *tokenized* slangs, acronyms, and emoticons, and removed all elongations. Finally, they *lemmatized* the words in their document (replaced them with the root of the word) and replaced words preceded by a negation with an antonym. In another case, Meduru et al. [40] simply removed all stop words ("a", "the", etc.), slang, emoticons (which are usually preserved as they reflect sentiment), hashtags, urls, and special characters. To the best of our knowledge, there is no single best approach or set of "required" preprocessing steps in the literature. The goal is to produce a dataset that is easy for a machine learning model to work with and that does not lose any expression of opinion and sentiment in the document.

2.2.2 Annotation

In parallel, the corpus will be annotated for the target observations if the machine learning task is supervised. Part of the dataset will be provided to human annotators who will manually carry out the same task as the NLP model. The implication is that the trained model will not learn to predict an "true" answer to a question but instead will provide answers that agree with the selected group of annotators. As a result, it is important to get a sense of annotator quality as well as the bias introduced [19].

There are two main approaches to guarantee a level of annotator quality. The first is to hire and train expert annotators. In this case, each individual is expensive but is likely to carry out the task with a high level of effort. The other option is to crowdsource the annotation on a platform such as AMT and add steps in the process that test for

annotation quality. For example, we can reject annotators based on wrong answers to known trivial examples or based on the time taken to perform a task.

Once we have high quality workers, it is important to have multiple annotators work on each example to control for annotator bias and to understand the inherent task difficulty. We will then use a statistic such as Cohen's Kappa or Krippendorff's alpha to score annotator agreement while accounting for chance agreement. Typically, the quality of an annotated dataset will be assessed by comparison with previous work in the literature. In applied work it can be difficult to find an appropriate reference, because a lower agreement on a more challenging task with the same possible answers does not imply that the dataset should be discarded.

In order to finalize the labels, it is important to select a strategy for resolving disagreements. Methods range from restrictive (by only keeping cases with full agreement) to permissive (where the majority class label is assigned). The restrictive scenario will be easier for the NLP model to classify, but harder cases that would no longer be modeled tend to have significant phenomenological differences to the easy ones [30]. The other option is to convert the task to a regression problem where the label has a strength reflecting the annotator agreement, but this is less common in the literature.

2.3 Text Classification

In this thesis, we experiment with a number of traditional machine learning algorithms that are used in text classification. We first discuss some methods of transforming tweets into feature vectors appropriate for NLP models. We then provide a brief explanation of

the algorithms that resulted in a reasonable performance during the modeling task. We also employ a number of deep learning models, so we will summarize various neural network architectures and word embeddings that are often used with them.

2.3.1 Feature vectors

N-grams

The simplest feature vectors for an NLP task are vectors of n-grams. An n-gram is a sequence of n consecutive items that are treated as a unit. The items can be either words or letters and we can specify the minimum and maximum length of the sequence. As a result, the feature space contains one dimension for every possible n-gram in the corpus. For example, given a corpus with one document $D = \{$"Chaz is a dog"$\}$, the possible 2-grams (bigrams) would be "Chaz is", "is a", and "a dog". Typically, *stop words* are removed, which are part of a blacklist of words that tend to carry little meaning for an NLP task. In the given example, "is" and "a" would be removed and the only remaining valid bigram would be "Chaz dog", resulting in a one-dimensional feature set. Given a feature space, we can then resort to a number of encoding schemes such as bag of words, count and TF-IDF vectors.

Bag of Words

The simplest method of encoding a document into the feature-space is to count the occurrence of each N-gram and place the result in the appropriate dimension. The weakness

of this method is N-grams that are very common across the corpus would appear more important than rare ones.

TF-IDF Vectors

Term Frequency-Inverse Document Frequency is an operation that seeks to fix the issue with count vectors by penalizing N-grams that occur many times across different documents. The theory is that the most significant features of a document are the N-grams that it uses often which are also uncommon in the rest of the corpus. The formula for this operation is:

$$\text{TF-IDF}(t, d, C) = \text{TF}(t, d) * \text{idf}(t, C) \tag{2.1}$$

t stands for the term (in our case, N-gram) of interest, d is the working document, and C is the entire corpus. The $\text{TF}(t, d)$ function is the term frequency, defined as the number of times the N-gram occurred in the document divided by the total number of N-grams. Finally, the inverse document frequency $\text{idf}(t, C)$ is log of the inverse of the fraction of the documents containing the N-gram:

$$\text{idf}(t, C) = \log \frac{|C|}{|d \in C : t \in d|} \tag{2.2}$$

where $|C|$ is the size of the corpus and $|d \in C : t \in d|$ stands for the number of documents d in the corpus C containing the N-gram t.

GloVe

An improvement introduced by the NLP community over count and TF-IDF vectors is the concept of *word embeddings*. The idea is to embed words in a vector space where we can reason mathematically about their semantics. These embeddings are usually learned from massive corpora assembled for a given language.

GloVe [56] is an unsupervised learning algorithm that aims to encode semantic similarity through vector distance. That is, vectors whose cosine similarity is smaller are expected to be closer in meaning or "relatedness". It relies on the assumption that "certain aspects of meaning can be extracted directly from co-occurrence probabilities" [56].

2.3.2 Traditional Machine Learning Algorithms

Logistic Regression

Logistic Regression is a popular choice when predicting the probability of a binary outcome variable in statistics and machine learning. It is equivalent to a linear regression where the outcome variable is measured on the logit scale. More specifically, given a Bernoulli distributed outcome variable Y with outcomes 0 or 1, a vector of independent predictors (in our case a feature vector) $\vec{x}$, and a vector of weights $\vec{\theta}$, we are interested in modeling the probability $p = P(Y = 1|\vec{x}, \vec{\theta})$. In order to do so, we fit a linear regression that corresponds to the relationship in the following equation:

$$l = \text{logit}(p) = \ln \frac{p}{1-p} = \vec{\theta}^T \vec{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \tag{2.3}$$

Given that we are trying to classify documents, we are interested in p itself so we invert the logit operator using a sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

In our case, the probability p is defined as:

$$p = P(Y = 1|\vec{x}, \vec{\theta}) = \sigma(\vec{\theta}^T \vec{x}) \tag{2.5}$$

$$1 - p = P(Y = 0|\vec{x}, \vec{\theta}) \tag{2.6}$$

In order to convert the probability obtained in Equation 2.5 to a classification label, we can select a decision boundary by selecting the most likely class based on the probability. In mathematical terms, we say $Y = 1$ if $p > 0.5$, otherwise $Y = 0$.

To fit this model, consider a dataset X, where each row is a feature vector $\vec{x}$. We want to maximize a likelihood function $L(\vec{\theta}|X, \vec{y})$.

$$L(\vec{\theta}|X, \vec{y}) = \prod_i p_i = \prod_i p_i^{y_i}(1 - p_i^{1-y_i}) \tag{2.7}$$

Typically, the negative log likelihood is used, giving the final formulation:

$$-\log L(\vec{\theta}|X, \vec{y}) = -\sum_i \Big(y_i \log(\sigma(\vec{\theta}^T \vec{x}_i)) + (1 - y_i) \log(1 - \sigma(\vec{\theta}^T \vec{x}_i)) \Big) \tag{2.8}$$

In order to minimize the negative log likelihood, we need to take the derivative of this function and set it to 0. Unfortunately, there is no closed form solution for this, so we have to use gradient descent.

Finally, it is common to use regularization to keep the parameters $\vec{\theta}$ reasonably bounded. There are two ways to do this: Lasso and Ridge Regression. Both methods add a penalty term to equation 2.8: the form uses the L1 norm $\lambda \sum_i |\theta_i|$ while the latter uses the L2 norm $\lambda \sum_i \theta_i^2$. In both cases λ is a hyperparameter that controls the strength of regularization that is tuned using grid search.

Random Forests

In order to understand random forests, we first need to discuss decision trees [7], which are also called classification trees in the appropriate context. The algorithm is based on growing a tree by iteratively dividing the prediction space into boxes, also called nodes. Specifically, the splitting is done in a top-down and greedy fashion. That is, at each iteration, we split every box in our prediction space into two and then repeat this step again on the smaller boxes until a stopping criteria is met. The split that we choose is the *best split* according to a selected measure of node purity, which is typically either entropy or the Gini index.

Both entropy and the gini index are a function of $p_{i,k}$, the proportion of examples from the training data that fit in the ith region and belong to the kth class. A low value in either metric indicates that a node contains values that mostly belong to a single class. The functions for entropy and the GINI index are given in the two equations below:

$$\text{Entropy} = -\sum_{k} p_{i,k} \log p_{i,k} \tag{2.9}$$

$$\text{GINI} = \sum_{k} p_{i,k}(1 - p_{i,k}) \tag{2.10}$$

The benefit of a decision tree over a linear model such as logistic regression is that it makes less assumptions about the structure of the data. They are also more interpretable as it is possible to see the selection rules at each step during classification

A random forest is a model that builds upon decision trees by introducing the concept of bagging [6]. We first *bootstrap* the dataset by building K new datasets by sampling N predictors from M training examples belonging to the original dataset. We then train a separate decision tree on each sampled dataset and select the most common prediction to produce a final classification result.

Support Vector Machines

Support vector machines [5] are a generalized version of a maximal margin classifier. The idea is to draw a line (or hyperplane) in the feature space that separates all the points belonging to one class from those belonging to another. While there are many such lines, we are interested in the one that also maximizes the distance to the nearest points (called support vectors).

The issue, however, is that in most cases the data is not linearly separable. The easiest way to rectify the issue is to introduce a slack variable that will create a soft decision boundary, where cases that are close but on the incorrect side will be tolerated. This

variable tolerance is a regularization parameter and can be tuned as a hyperparameter to the SVM.

The second option is use a function to project the data into a higher dimension where it is linearly separable. This is called the kernel trick and there are a number of possible *kernels* such as a linear, polynomial, sigmoid, and more [24].

2.3.3 Deep Learning Algorithms

With the increase in available computing power, neural networks have become a viable solution to many complex NLP tasks. Since then, architectures such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) have achieved state-of-the-art results in most tasks, including various subproblems of opinion mining. In this thesis, we make use of feed-forward neural networks, CNNs, RNN architectures, and more advanced models such as ELMo. We also discuss GloVe, which is a model for representing words as vectors.

Feed-forward Neural Networks

Feed-forward neural networks are built using *perceptrons* [63]. A perceptron consists of four components: an input vector with elements $x_1, \cdots, x_N$, randomly initialized weights $w_1, \cdots, w_N$ and a bias term w_0, a sum, and an activation function f. As a result, the output y is defined as:

$$y = f(w_0 + \sum_{i=1}^{N} w_i x_i) \tag{2.11}$$

It is interesting to note that if f is the identity function $f(x) = x$, then a perceptron model is exactly a linear regression. If we use a sigmoid activation function, then the model becomes very similar to a logistic regression.

In order to train the perceptron, we will first make a (forward) prediction y^t using the weights we currently have w_i^t and equation 2.11. Then we will compute an error term and use it to update weights. This will produce new weights w_i^{t+1} using the following formula for each element in the input vector x_i:

$$w_i^{t+1} = w_i^t + r(d - y^t)x_i \tag{2.12}$$

Here, $r \in [0, 1]$ is a learning rate that controls the strength of the weight change and d is the desired output. It follows that a larger error $d - y$ forces a bigger weight adjustment.

Moving on from a single perceptron, we can build a feed-forward neural network by noticing that if we use M perceptrons (nodes) and give them the same inputs, then we will obtain M different outputs in the early stages of training. Furthermore, we can send these outputs into another set of N nodes, and the outputs of those into a final node. The first two sets of perceptrons would then be called *hidden layers* and are responsible for projecting the data from a previous layer into a different vector space.

Updating the weights works in a similar way to the individual perceptron, but the hidden layers will not use the predicted output as their d. Instead, there is a backpropagation [65] process that transfers the output error into input errors that can be used by the previous layer.

Convolutional Neural Networks

Convolutional Neural Networks [35] were first introduced in the context of image classification, but have shown good performance in NLP tasks such as sentiment analysis [28]. Their new additions over a feed-forward achitecture are the concepts of *convolution* and *pooling layers*. Convolution layers are regularizing as they require fewer parameters than fully connected layers because of the convolution operation which essentially uses a *filter* to *scan* the input space and produce an *activation map*. This activation map is responsible for detecting the presence of some features based on the combination many adjacent inputs from the previous layer.

Recurrent Neural Networks

Recurrent Neural Networks [64] build on the feed forward architecture by sending some connections backwards, effectively forming loops. This adds a temporal component to its output which is very valuable for modeling sequential data such as text. When reasoning about their operation, we often use an *unrolled* version, which is an equivalent feed-forward neural network whose behaviour is identical over a finite period of time [42]. In this case, we reason about the model as a a network of *cells* whose weights are shared and that are responsible for storing the *memory* of the network.

Bidirectional RNNs are another useful architecture. In NLP, this is done by combining the outputs of an RNN that reads the text in the correct direction with one that does the opposite. The result is that our model can encode the context for each step by also looking ahead.

Learning RNN models is more complicated than previous architectures and it requires a modified backpropagation algorithm called Backpropagation Through Time [44]. The other challenge that has a bigger impact in RNNs is that of *vanishing or exploding gradients*. The issue is that the derivative term of the activation function during backpropagation becomes so small (or large) that the weight update effectively stops (or increases uncontrollably). This issue can be mitigated through specific choices of activation functions or by using special architectures such as Long Short Term Memory Networks (LSTM).

Long Short Term Memory Networks

LSTMs [23] were developed specifically to deal with the vanishing gradient problem in RNNs. They introduce the concepts of *input, output, and forget gates*. Broadly speaking, each cell has a forget gate that decides what information from the input and previous hidden state to keep and ignore. It also has an input gate that will update its internal state. Finally, the output gate sets the value of the hidden state provided to the next cell.

2.3.4 Deep Learning and Word Embeddings

There are a few strategies to feed textual data to a neural network. In the case of feed-forward neural networks, we are free to use a similar feature vector as in the traditional models. However, once we resort to RNNs, each timestep typically receives a new word as a complete vector. A naive solution would be to use a one-hot encoding, where we would provide a vector of 0s in all columns except the one corresponding to the current word. This is not particularly useful and instead we will usually use some form of word embedding like GloVe, or alternatives that rely on deep learning such as ELMo.

ELMo

One limitation with GloVe is that it does not account for synonymy. In particular, a word that has multiple meanings will have the same vector representation irrespective of which definition is implied by the rest of the sentence. ELMo [57] attempts to account for this by learning the word embeddings using a stacked bidirectional LSTM trained on a massive corpus with over a billion words of data [9]. The result has proven to have excellent performance in many NLP tasks including sentiment analysis.

2.3.5 Model Selection

When carrying out a supervised classification exercise, we are using a labeled dataset to teach a model how to predict future examples that it will encounter in the real world. With a perfect model, dataset, and with a subject matter that does not evolve over time, we would be able to learn every single relevant case and classify every single data point correctly. Unfortunately none of the three assumptions are true, but we can at least get a sense of how well we think the model can do under the assumptions of the dataset that we have. We have to choose an evaluation metric to score our model that most accurately reflects the relative cost of different types of incorrect classifications. In order to estimate the score our model will have on unseen data, we will use cross validation to train on a partial dataset and test on the rest. Finally, in order to be able to find the best model, we will execute a grid search over all models and their hyperparameters to find the one that gives the best expected test score.

Evaluation Metrics

In a binary classification problem, there are four types of results: true positives, true negatives, false positives, false negatives. In the COVID-19 vaccination opinion identification task, their interpretation is as follows:

- **True positive (TP)**: The model correctly predicts that the tweet contains an opinion
- **True negative (TN)**: The model correctly predicts that the tweet does not contain an opinion
- **False positive (FP)**: The model predicts that the tweet contains an opinion, when it does not
- **False negative (FN)**: The model predicts that the tweet does not contain an opinion, when it does

By choosing a different scoring metric, we can specify which of these cases is more important to us. Unfortunately, there is a trade-off. If we say that we we are interested in true positives, then the way to maximize that number is by saying every tweet contains an opinion, which would also maximize the number of false positives. If we decide that this is unacceptable and we cannot have any false positives because that will pollute the downstream opinion mining task, then the model can just say that no tweet contains an opinion.

Typically, the priorities are then refined to the following evaluation metrics:

Accuracy

Accuracy is the proportion of correct classifications divided by the total number of predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.13}$$

This is an intuitive concept but it is sensitive to class imbalance. If only 10% of all the tweets in the dataset contain an opinion, a classifier that always returns "no opinion" will easily achieve 90% accuracy despite being useless.

Precision

Precision looks at how many of the predicted positive labels are actually positive:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.14}$$

In the example with the model returning "no opinion" for all tweets and having an accuracy of 90%. It will have a precision of 0%, surfacing an issue with the model that was missed by accuracy.

Recall

Recall is interested in measuring what is the percentage of the positive cases that were labeled as positive:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.15}$$

In the running example, we would also have a recall of 0%. The choice between precision and recall lies in whether a false positive is more costly than a false negative. If it is absolutely unacceptable to flag a tweet that has no opinion as a positive case, then we would favor precision. If we cannot afford to miss a single opinionated tweet but are happy to sort through false positives, then recall is the better option.

F1 Score

If we are equally interested in controlling false positives and false negatives, we will use the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{2.16}$$

Balanced Accuracy

This metric is an improvement on accuracy that accounts for class imbalance. It corresponds to the average per-class accuracy:

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2} \tag{2.17}$$

The main reason to optimize balanced accuracy over an F1 Score is that the F1 score is only interested in the positive class. For example, an extra true positive will be more significant than an infinite amount of true negatives.

Cross validation

When we train a model, the result is the closest fit to the data that is possible. This data is a sample of what the model will encounter in the real world, and the performance

score against previously unseen data will be lower. In the extreme case, if our model perfectly classifies a noisy training dataset, then it will "overfit" and will suffer a drastic performance reduction in the real world. We can simulate this situation by splitting our training dataset into a training and validation set and using the latter as a simulation of the real world.

Cross validation is a generalization of this concept where the dataset is split into k folds and the train-test exercise is run k times. In each iteration, one fold is the validation set and all the rest are combined to form a training set. We will use *stratified* cross validation, which preserves the class imbalance across all folds.

When paired with a grid search over a model's hyperparameters, we can control overfitting by selecting the model with the best validation performance rather than training performance.

Grid Search

It is impossible to know ahead of time which algorithm will have the best performance, or what its regularization settings should be. It is also difficult to predict which preprocessing steps will best cooperate with each model. We carry this out by brute force by specifying all the possible parameters for each component of our pipeline that we are willing to try and trying a cross validation procedure on every possible permutation.

Chapter 3

OPINION DETECTION

In this chapter we posit that there should be a separation between opinion detection and downstream opinion mining tasks. We then discuss the techniques leveraged by current applied research and the datasets available for advancing the state of the art of opinion detection.

3.1 Opinion Mining or Detection

An important use case for opinion mining is to guide decision-making where it needs to be informed by some population's conversations. As a result, a step involving stance detection or sentiment analysis is necessary to produce a classification that can inform decisions. The workflow consists of a typical NLP pipeline beginning with data collection, followed by preprocessing and classification. While it is clear that design choices made during preprocessing and data collection have an impact on the classification, the effects have not always been clear [74]. In particular, researchers have to choose whether

the final step will receive documents that all contain opinions or whether they will add a label to signal that no opinion was detected.

This inherently requires that the final step be able to do both the work of detecting and classifying opinions. We therefore need to use a more powerful model that is able to solve two tasks at the same time, with the detection component often being more complex than the classification task [41].

Jointly learning two tasks is challenging even for the most complicated models available because it requires an understanding of distinct concepts. It is clear, for example, that opinion detection and sentiment analysis share indicators in the form of the presence of terms such as "like" and "hate". It is less obvious for an NLP model that the two are not always correlated, with example phrases being "I think he loves vaccines" (opinion, no sentiment) and "I am annoyed, but I am getting my vaccine later" (sentiment, no opinion). A sentiment analysis model solving both tasks jointly would have to learn to not only understand when the term "love" carries sentiment, but also when to ignore a sentiment-laden phrase because it is not of value to a decision-maker.

3.2 Techniques

Another effect of the trend of combining the detection and classification of opinions is that progress in detecting opinions is fragmented across various opinion mining tasks. As a result, typical solutions for detecting opinions in applied work are not based on benchmarked research. Most of the solutions can be grouped into three categories: manual, heuristic, and lexicon-based.

An example of a manual solution is the work of Yoon et al. [82], who classified the polarity of tweets surrounding the 2014 Seoul mayoral election. After annotation of their training set, the authors discarded the tweets that did not have a polarity. This is a reasonable choice in a research context but could not scale to a production system because it would be important to continue to manually filter out tweets without polarities especially as the topics drift away with time from the training dataset.

Fan and Wu [17] use a heuristic method to summarize opinions of comments on various e-commerce products. They identify opinions by checking for the presence of adjectives from an expanding list based on a seed set. Such a method might be reasonable, but it would have to be compared to other methods using the same dataset before its performance can be determined objectively. To the best of our knowledge, neither this nor any other heuristic method has been established as a state of the art tool for identifying opinions.

Sindhu et al. [70] use the Opinion Finder lexicon [80] to show how the performance of a polarity classifier increases along with a threshold for subjectivity. While, Opinion Finder is no longer competitive in sentiment analysis compared to more recent tools [62], the benefit it provides to a downstream model remains valuable. Moreover, it is not clear that a two-step process using more modern NLP tools could not achieve or surpass the current state of the art.

3.3 Datasets

Researchers interested in improving the state of the art for a task use benchmark datasets to form a basis for comparison to prior work. There are few established datasets built for opinion detection compared to popular tasks such as sentiment analysis (e.g. [29, 38]) or stance detection (e.g. [12,43]). The ones that do exist tend to be older (e.g. [52]) and recent literature often resorts to adapting datasets such as SemEval-2016 Task 6 [43] instead.

The dataset provided by Pang and Lee [52] requires a system to specify whether or not a sentence is subjective. It was curated by combining mostly "objective" sentences from plot summaries on the International Movie Database (IMDb) with "subjective" sentences from reviews on the Rotten Tomatoes website. It is not clear, however, whether training on the syntax of movie reviews is representative of how opinions are communicated in a broader online conversation.

The SemEval-2016 Task 6 dataset [43] is sometimes used in the literature [13] whenever opinion identification is carried out explicitly. It was created by collecting tweets for six topics: "Atheism", "Climate Change is a Real Concern", "Feminist Movement", "Hillary Clinton", "Legalization of Abortion", "Donald Trump". The tweets were then annotated for their stance towards the given topic with the following categories: "favor", "against", "neutral", and "no stance". The last two categories were merged because less than 0.1 % of the data was assigned the "neutral" label. A second question asked annotators to specify which of the following options is true for a given tweet:

1. The tweet explicitly expresses opinion/sentiment about the target

2. The tweet expresses opinion/sentiment about something/someone other than the target

3. The tweet is not expressing opinion/sentiment about anything

In order to train a model to identify opinions based on this dataset, the first two categories of the second question can be combined. However, such modified datasets are rarely used as a standard benchmark for opinion detection performance. As a result, applied research that seeks to separate opinion identification from downstream tasks does not have a clear reference for which models are likely to have the best performance.

Chapter 4

DATA

The COVID-19 case study that we carry out in this thesis aims to help downstream NLP systems characterize various opinions related to the vaccine held by Canadians on Twitter. In this Chapter, we describe the collection and annotation processes to create our labeled dataset.

4.1 Collection

In a true applied setting, it would be important to know how common is a conversation on the vaccine compared to any other subject. Without such a measure, it would be difficult to describe the likelihood of an average Canadian Twitter user expressing an opinion on the COVID-19 vaccine. In order to prevent this issue, we use a Twitter dataset collected by the Media Ecosystem Observatory (MEO) that focuses on capturing a representative sample of the Canadian online Twitter population instead of searching for specific tweets related to a narrow topic [8]. This was done by maintaining an expanding list of likely-

Canadian users and collecting their complete posting histories. The list was initialized during the 2019 Canadian elections and expanded in 2020 using the COVID-19 pandemic as one of the topics for user discovery. Specifically, the procedure is as follows:

1. Initial seed user set creation: During the 2019 Canadian elections, a list of accounts belonging to political parties was manually assembled and defined as a seed user set.

2. New user discovery: A list of popular hashtags was also maintained during the elections and a sample of tweets for each popular hashtag was collected.

3. Location retrieval: The self-reported locations (and mentions of locations in their descriptions) for each user in Step 2 not already present in the seed set were collected.

4. Location parsing: These locations were mapped to a country using Google's Geocoding API [1]. Manual validation ensured that text with city names existing in other countries such as Kingston (Canada, Jamaica, or many others) resolved to the correct location.

5. Seed user set expansion: Users with a Canadian self-reported location were then added to the seed set.

6. Scraping and reexpansion: In 2020, the timelines of the seed users were scraped and the previously unknown users that appeared in the mentions, retweets, and replies of the seed set underwent steps 3, 4, and 5.

[1] https://developers.google.com/maps/documentation/geocoding/overview

The result was a dataset of approximately 452 million tweets in 2020 from likely-Canadian users. One limitation of this collection procedure is that the political conversation from which it originated could contain bots or fake accounts that have an interest in incorrectly self-reporting a Canadian location. It would be challenging to reliably filter out such users given the difficulty of detecting inauthentic accounts [16].

We proceed with our case study by departing from the MEO dataset by filtering tweets for the presence of keywords related to the vaccine. The keywords were selected by looking at a co-occurrence matrix that included the terms "covid-19" and "vaccine" to find new terms that would significantly increase the dataset size. The resulting keyword list is:

- "vaccine" and at least one of: "covid19", "covid-19", "coronavirus", "microchip", "oxford", "mrna"
- "astrazeneca"
- "pfizer"
- "moderna"

The resulting dataset contains 8,161,582 tweets that mostly discuss a vaccine related to COVID-19. The list was built over multiple iterations by manually inspecting for problematic cases. For example, the word "vaccine" on its own returned conversations unrelated to the COVID-19 vaccine. Meanwhile, "moderna" did not need to be tied to "vaccine" because manual inspection of a sample did not surface unrelated tweets. This manual process is analogous to sentiment analysis research that filters for opinions heuristically. In our case, the upstream task that would be required to reduce the manual intervention

step is topic modeling, where we train a model to predict whether a tweet is related to a COVID-19 vaccine. Such an exercise is beyond the scope of this thesis and we therefore simplify the task by filtering on topics as is standard practice in applied NLP literature (e.g. [49,77]).

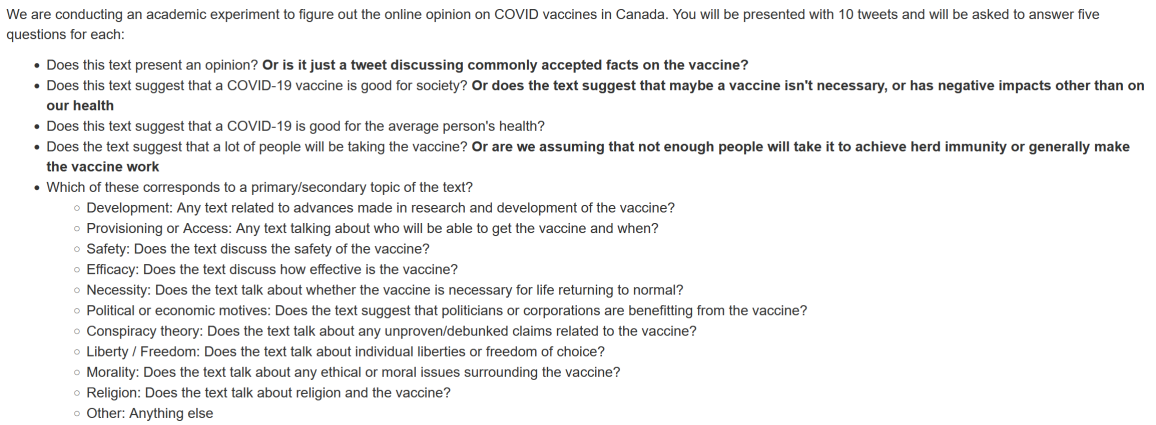
Using the filtered COVID-19 vaccine dataset, we then clean up a tweet's text by removing all URLs and mentions from each tweet. This is important because labeling a tweet as having an opinion based on the presence of either of the two elements would be incorrect. The two key special features of tweets that we do ensure stay present are emoticons and hashtags, because they often form a key part of the message the author is trying to communicate.

4.2 Annotation

We carry out two annotation tasks on AMT for our case study with three annotators per annotated example. In the opinion identification task (see instructions in Figure 4.1), we randomly sample 5000 tweets and create 500 annotation tasks with 10 tweets each. We require annotators to answer a number of questions including whether or not each tweet contains an opinion. This question is difficult because we avoid providing a narrow definition of opinion in order to obtain labels that reflect an intuitive human judgment. More specifically, the casual definition of opinions is subjective and some annotators will search for the presence of sentiment, while others focus on the veracity of the statement. The consequence of allowing annotators to define opinion using their intuition is that there is more room for disagreement and bias (see Table 4.1).

Tweet	Opinion Justification	No Opinion Justification
So Pfizer announce they've discovered a Covid vaccine just days after the election?	Author implies vaccine is merely a political tool	The opinion presented is unrelated to the vaccine itself, rather the politics surrounding it
@thomaskaine5 My arm was a little sore the night of, mild muscle aches the next day. I've had mosquito bites more painful than the covid vaccine.	Author provides a subjective description of vaccine's side-effects with an implication that the vaccine is safe	The tweet is a strict recollection of what happened to the author

Table 4.1: Examples of tweets where the presence of opinion was subject to disagreement

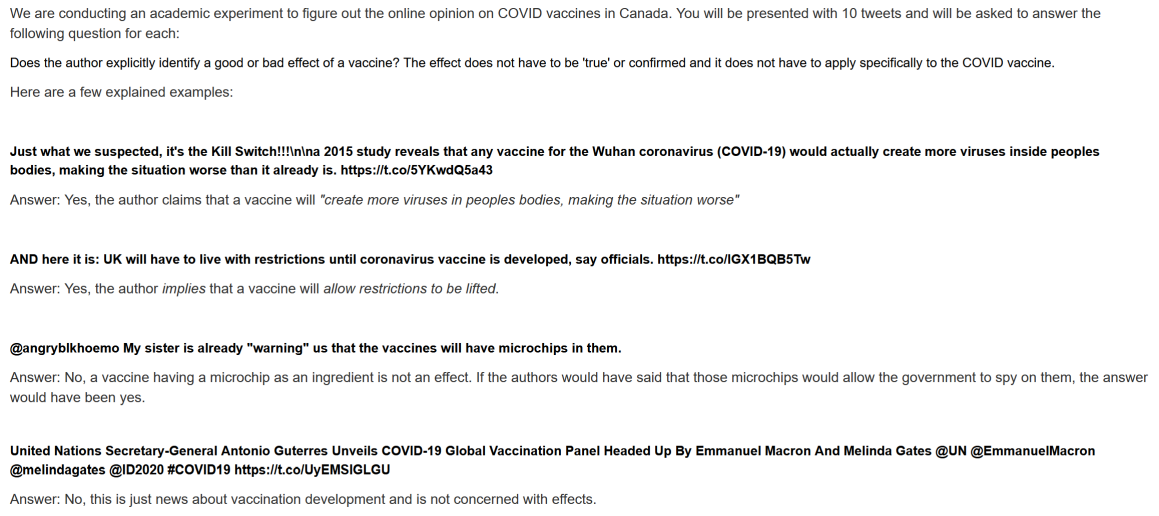

We are conducting an academic experiment to figure out the online opinion on COVID vaccines in Canada. You will be presented with 10 tweets and will be asked to answer five questions for each:

- Does this text present an opinion? **Or is it just a tweet discussing commonly accepted facts on the vaccine?**
- Does this text suggest that a COVID-19 vaccine is good for society? **Or does the text suggest that maybe a vaccine isn't necessary, or has negative impacts other than on our health**
- Does this text suggest that a COVID-19 is good for the average person's health?
- Does the text suggest that a lot of people will be taking the vaccine? **Or are we assuming that not enough people will take it to achieve herd immunity or generally make the vaccine work**
- Which of these corresponds to a primary/secondary topic of the text?
  - Development: Any text related to advances made in research and development of the vaccine?
  - Provisioning or Access: Any text talking about who will be able to get the vaccine and when?
  - Safety: Does the text discuss the safety of the vaccine?
  - Efficacy: Does the text discuss how effective is the vaccine?
  - Necessity: Does the text talk about whether the vaccine is necessary for life returning to normal?
  - Political or economic motives: Does the text suggest that politicians or corporations are benefitting from the vaccine?
  - Conspiracy theory: Does the text talk about any unproven/debunked claims related to the vaccine?
  - Liberty / Freedom: Does the text talk about individual liberties or freedom of choice?
  - Morality: Does the text talk about any ethical or moral issues surrounding the vaccine?
  - Religion: Does the text talk about religion and the vaccine?
  - Other: Anything else

Figure 4.1: Instructions for annotation task with the opinion detection question

An additional problem that we handled is annotator effort. Many crowdsourced annotators rushed through the task and after manual inspection, we observed that annotators who finished in less than 2 minutes and 30 seconds (15 seconds per tweet) were answering randomly.

In the effect mention task (see instructions in Figure 4.2), we ask the annotators whether a tweet's author explicitly identifies a good or bad effect of a vaccine. We also insist that the effect does not have to be "true" or confirmed and that it does not have to apply specifically to the COVID vaccine. We provide in the task instructions numerous examples of


We are conducting an academic experiment to figure out the online opinion on COVID vaccines in Canada. You will be presented with 10 tweets and will be asked to answer the following question for each:

Does the author explicitly identify a good or bad effect of a vaccine? The effect does not have to be 'true' or confirmed and it does not have to apply specifically to the COVID vaccine.

Here are a few explained examples:

Just what we suspected, it's the Kill Switch!!!\n\na 2015 study reveals that any vaccine for the Wuhan coronavirus (COVID-19) would actually create more viruses inside peoples bodies, making the situation worse than it already is. https://t.co/5YKwdQ5a43

Answer: Yes, the author claims that a vaccine will *"create more viruses in peoples bodies, making the situation worse"*

AND here it is: UK will have to live with restrictions until coronavirus vaccine is developed, say officials. https://t.co/IGX1BQB5Tw

Answer: Yes, the author *implies* that a vaccine will *allow restrictions to be lifted.*

@angryblkhoemo My sister is already "warning" us that the vaccines will have microchips in them.

Answer: No, a vaccine having a microchip as an ingredient is not an effect. If the authors would have said that those microchips would allow the government to spy on them, the answer would have been yes.

United Nations Secretary-General Antonio Guterres Unveils COVID-19 Global Vaccination Panel Headed Up By Emmanuel Macron And Melinda Gates @UN @EmmanuelMacron @melindagates @ID2020 #COVID19 https://t.co/UyEMSIGLGU

Answer: No, this is just news about vaccination development and is not concerned with effects.

Figure 4.2: Effect mention task instructions

our annotation expectations and reject annotators who misunderstand the exercise. This is tested by injecting the following two easy questions into their task:

- "SHOCK VIDEO: Bill Gates Admits COVID-19 Vaccine Will Kill And Maim 700,000!": **Effect mentioned**
- "The Ontario government is making a $20 million investment towards provincially-based research to help find a vaccine for COVID-19": **No effect**

In order to compute agreement on these two tasks, we generate sparse matrices with columns assigned to annotators and rows assigned to each tweet. We obtain a Krippendorff's alpha of 0.27 for the opinion detection task and 0.45 for the effect mention task. In terms of consensus, 50% tweets for opinion detection and 63% for effect mention have agreement between all three annotators.

To some extent, the reliability measures obtained for the opinion identification annotation are worrying and indicative of issues in the dataset. Even accounting for the fact

that Krippendorff's alpha is affected by class imbalance [27], we expect a high number of misclassifications. This highlights the fact that intuitive definitions of opinion do not form a sharp boundary. The impact is that the trained model will be more permissive in what it identifies as an opinion. In fact, it will learn to predict whether a tweet is more likely than not to be opinionated, which is sufficient for a downstream task. To train such a model, we form a dataset by selecting tweets where a majority could be obtained after discarding low effort annotators. The result contains 3676 tweets, 2441 of which are opinionated tweets and 1235 tweets that are not.

For the effect mention task, we have a higher agreement and consensus percentage because we enforce a strict definition rather than relying exclusively on annotator judgment. Given that the annotator is given precise acceptance criteria for the task, we can filter workers more aggressively based on incorrect answers to easy questions. Nevertheless, some disagreements do occur and we account for them by assigning the majority answer to the label. Having annotated 376 tweets from the 5000 belonging to the opinion task, we form a dataset containing 112 tweets that mention an effect of the vaccine and 264 that do not.

Chapter 5

MODELING

We carry out an opinion identification task on our COVID-19 dataset and also classify whether or not a tweet mentions an effect of the COVID-19 vaccine. Both tasks are a binary classification problem with the two possible answers being "yes" and "no". Despite the outcome variable being binary, we need to remember that the models predict whether a *majority* of annotators would think that a tweet is opinionated or mentions an effect. In the case of opinion identification, the downstream model will have access to a smaller set of tweets where identification was easier. On the other hand, the diffuse definition should ensure that a varied set of opinionated tweets are preserved.

We present in this chapter the pipelines for the traditional and deep learning models.

Pipeline Component	Hyperparameter	Values
Count & TF-IDF Vectorizer	N-gram sizes (min,max)	(1,1), (2,2) (3,3), (1,3)
Count & TF-IDF Vectorizer	minimum n-gram corpus frequency percentage	0.05, 0.11, 0.23, 0.5, 1.08, 2.32, 5.0, 10.77, 23.20, 50.0
SVM	$C \in [0, 1]$ (inverse of regularization strength)	0, 0.25, 0.5, 0.75, 1
SVM	Kernel	linear, polynomial rbf, sigmoid
Logistic Regression	Regularization Type	None, L1, L2
Random Forest	Number of Estimators	5, 10, 15
Random Forest	Class Weight (to handle class imbalance)	balanced subsample

Table 5.1: Hyperparameter Grid

5.1 Traditional Machine Learning Methods

We will use logistic regression as a benchmark algorithm and try to improve on it using an SVM and a random forest. For all three algorithms, we experiment with both count and TF-IDF vectors based on word n-grams.

We carry out a grid search with a 5-fold stratified cross validation using each of the following metrics for evaluating the validation set: precision, recall, f1, weighted f1, and balanced accuracy. Table 5.1 summarizes the parameter ranges for each component of the pipeline. The tuning process is focused on exploring various scales for the numerical hyperparameters and the minimums and maximums were defined after manual experimentation. The model that will be selected could likely be further improved on by carrying out an exhaustive search around its hyperparameter values. All traditional models were implemented using Python's scikit-learn library [55].

5.2 Deep Learning Methods

With the rise of deep learning, a diverse set of neural network architectures has been used to solve opinion mining tasks. These models aim to reduce the feature engineering burden on the researcher by dealing with word tokens at the input level instead of feature vectors. In applied research, the core of the work consists of designing an architecture that is able to model the patterns and domain-related concepts that can be extracted from the data.

In this thesis, we use an architecture that has three key steps: an embedding layer that is followed by an encoder and which is finally fed into a feed-forward classifier (with a softmax output). The embedding layer is responsible for converting each word in a tweet into a vector representation. We experiment with GloVe, ELMo, and no embeddings. The encoding step uses a deep neural network architecture to convert the sequence of vectorizer words into a single vector. We try LSTMs, BiLSTMs, CNNs, and a *Bag of Embeddings* (BOE) model, which sums up all the word vectors together. Finally, the feed-forward classifier uses a standard fully-connected neural network that outputs one value per possible classification label and the softmax function at the end converts this output to a probability.

As discussed throughout Chapter 4, both the opinion detection and effect-mention datasets are highly skewed, so we must use a weighted cross entropy loss function [61]. In order to learn the network weights inside the system, we use the Adam optimizer [34] for 30 epochs, but we stop training early if the validation loss and F1 score do not improve

for 5 epochs. All models were built using AllenNLP [18], which in turn relies on PyTorch [54].

Chapter 6

EVALUATION & DISCUSSION

In this chapter, we first discuss the performance of deep learning and traditional NLP models in the case study and reflect on how to use this model in an applied pipeline. We also discuss the phenomenological findings related to the COVID-19 vaccine conversations on Twitter.

6.1 Model Performance

6.1.1 Opinion Identification

Deep learning models had worse opinion detection performance under the weighted F1 score, but were stronger on all other metrics. The test scores of traditional models and deep learning models are reported in Tables 6.1 and 6.2, respectively. This is the case despite the fact that the model selection process for traditional algorithms specifically used each scoring metric during cross validation whereas the deep learning models solely

relied on the loss function. Traditional learning models trained to maximize recall failed to learn the task correctly and obtained a perfect score by simply predicting that every tweet contains an opinion.

We hypothesize that an important reason for this improvement is that the deep learning architectures we used are a form of *transfer learning*. This dataset is very small and therefore extracting an understanding of English semantics using count or TF-IDF vectors is challenging. Therefore, the extra prior knowledge injected into the model by the contextual embeddings make it easier to train the opinion identification task on just a few thousand tweets.

In our case study, a logistic regression model is a sufficient approach for detecting opinions in tweets for a downstream task. During model selection for our case study, we need to minimize the number of examples that are incorrectly labeled as containing an opinion (false positives), which implies maximizing precision. However, we want to generate as large and diverse a dataset as possible, meaning we want to find as many of the opinionated tweets as possible, maximizing recall. This, coupled with the class imbalance, means we should favor the weighted F1 metric over balanced accuracy (and precision or recall). Balanced accuracy is less desirable because we are less concerned with false negatives than false positives. Specifically, given that a majority of tweets do contain an opinion, we can be strict with identifying opinions based on a diffuse definition implicitly defined by annotators and focus on correctly classifying the true positives.

We therefore have to concede that the deep learning models we have used do not seem to have created a significantly better downstream dataset than a simple logistic regression using N-grams. This is a similar result to Igarashi et al. [26], who carried out a stance

Algorithm	F1	F1 Weighted	Balanced Accuracy	Precision	Recall
LogisticRegression	**0.874**	**0.870**	**0.792**	**0.848**	1.0
RandomForestClassifier	0.856	0.845	0.757	0.824	1.0
SVC	0.871	0.868	0.770	0.846	1.0

Table 6.1: Opinion identification performance metrics for traditional models

detection exercise using a CNN and traditional models and found that the traditional approach resulted in a better performance on a test dataset, despite having higher scores during cross validation.

Further, these results suggest one of two possibilities: either there are no further clues about opinion in a tweet's syntax or the deep learning models are not learning syntax. The first option would imply that models based on N-gram frequencies are close to the limit of task performance for this dataset and the errors are caused by annotation noise. The second option implies that there is still room for improvement in task performance, but deep learning models that are insensitive to syntax perturbations such as word order [71] are not reasoning about language to an extent that is sufficient for detecting opinions. More specifically, such models seem to reason about language in a way that is not significantly different from a bag of words approach, despite the additional complexity they introduce.

6.1.2 Effect Mention

Based on the annotation results, only 40% of tweets containing an opinion mention an effect of the vaccine. The dataset for the task was too small for models to learn to classify whether a tweet contains an opinion. We see in Tables 6.3 and 6.4 that the expected advan-

Embedding	Encoding	F1	F1 Weighted	Balanced Accuracy	Precision	Recall
GloVe	LSTM	0.840	0.802	0.803	0.903	0.785
	BiLSTM	0.743	0.718	0.759	0.929	0.619
	CNN	0.886	0.841	0.809	0.871	0.902
	BOE	0.815	0.774	0.774	0.887	0.754
None	LSTM	0.860	0.819	0.807	0.890	0.832
	BiLSTM	0.893	0.845	0.805	0.861	0.928
	CNN	0.867	0.799	0.745	0.821	0.918
	BOE	0.870	0.831	0.818	0.896	0.846
ELMo	LSTM	0.864	0.830	**0.831**	**0.919**	0.816
	BiLSTM	0.887	0.840	0.805	0.867	0.908
	CNN	**0.905**	**0.863**	0.829	0.878	**0.932**
	BOE	0.845	0.810	0.817	0.918	0.783

Table 6.2: Opinion identification performance metrics for Deep Learning models

Algorithm	F1	F1 Weighted	Balanced Accuracy	Precision	Recall
LogisticRegression	0.280	0.280	0.576	0.500	0.143
RandomForestClassifier	**0.491**	**0.444**	0.525	**0.667**	**0.833**
SVC	0.373	0.414	**0.589**	0.600	0.262

Table 6.3: Effect mention performance metrics for Deep Learning models

tage from using pretrained deep learning models did not translate to a useful prediction performance in a low data setting.

Embedding	Encoding	F1	F1 Weighted	Balanced Accuracy	Precision	Recall
GloVe	LSTM	**0.551**	0.210	**0.500**	**0.381**	**1.000**
	BiLSTM	0.430	0.466	0.475	0.359	0.535
	CNN	0.485	0.356	0.465	0.360	0.744
	BOE	0.229	**0.492**	0.457	0.296	0.186
None	LSTM	0.542	0.206	0.488	0.375	0.977
	BiLSTM	**0.551**	0.210	**0.500**	**0.381**	**1.000**
	CNN	**0.551**	0.210	**0.500**	**0.381**	**1.000**
	BOE	0.526	0.217	0.472	0.367	0.930
ELMo	LSTM	**0.551**	0.210	**0.500**	**0.381**	**1.000**
	BiLSTM	**0.551**	0.210	**0.500**	**0.381**	**1.000**
	CNN	0.513	0.195	0.453	0.358	0.907
	BOE	0.476	0.197	0.414	0.337	0.814

Table 6.4: Effect mention performance metrics for Deep Learning models

Stance	Tweet
Pro-vaccine	@NicolleDWallace @maddow @MSNBC @ChrisCuomo @CNN KEY OFFICIAL -A VIROLOGIST WORKING ON A #COVID19 VACCINE-TAKEN OFF THE JOB B/C HE HAD THE AUDACITY TO TELL #trump #Hydroxychloroquine is BULLSHIT. Once again, trumps thin skin comes b4 OUR LIVES. https://t.co/cpv8X2vUAg
Pro-vaccine	@angryblkhoemo My sister is already "warning" us that the vaccines will have microchips in them.
Pro-vaccine	Dr. Rick Bright is a career not political official who was leading the effort to develop a #coronavirus vaccine as the director of @BARDA. That level of expertise is not mediocre. https://t.co/AvTUSgIou3
Against	The whack-jobs telling me coronavirus kills or almost killed me..We need to keep the country closed till we get a vaccine.. GET REAL!! I was hit by a drunk police officer and almost died and had my last rights read to me twice..should we ban cars, alcohol, or Police..NOT!
Against	Thats a big hell no. Tom Hanks' blood will be used to develop coronavirus vaccine https://t.co/1k9i2Bw6nu
Against	@Mareq16 gates: "i want to microchip everyone when i vaccinate them" bezos: "i want people out of work so my sales increase" dems: "we want mail in voting so we can rig the election" china: "i have a deal for all of you. bat soup anyone?"

Table 6.5: Examples of opinionated tweets that do not mention a vaccine's effects

6.2 COVID Vaccine Takeaways

Many users express an opinion on the COVID-19 vaccine, but they rarely mention a concrete effect. Instead, users usually express an opinion by making an associative reference between the vaccine and something that has an implied positive or negative connotation. Table 6.5 gives examples of opinionated tweets that do not specify an effect caused by a vaccine and are typical of the online conversation.

We also find that amongst the tweets without an opinion, 20% mention an effect while 40% of opinionated tweets do so. This makes sense given that there is a significant volume of unopinionated tweets that report on progress of vaccine development and other related news.

Chapter 7

CONCLUSION

Debates and conversations on social media play an important part in shaping the world-view of some users and many NLP tasks are defined to characterize the opinions that circulate within. Systems solving these tasks could be valuable to policymakers and anyone else who has an interest in the public's opinion on a topic. The current trend in the literature is to merge opinion detection with the downstream task of interest and solve the two at the same time.

In this thesis, we first argued that this trend has created a difficult situation for applied research because there is little recent work clarifying what is the state of the art for opinion detection. Moreover, the trend in the literature seems to be one of overlooking the opinion identification task to the detriment of general opinion mining work. We then carried out a case study on COVID-19 vaccine hesitancy to demonstrate the hardness of the opinion identification task. In order to do so, we first assembled a dataset of Canadian tweets mentioning the COVID-19 vaccine. Next, we annotated 5000 tweets for the presence of opinions and 376 for the mention of effects. Finally, we trained a number of traditional

and deep learning NLP models and measured their ability to detect opinions under a variety of metrics.

We found that annotators' definitions of opinion vary significantly, which is a challenge for the creation of a clean annotated dataset because it reduces the likelihood of obtaining a consensus. In terms of weighted F1, which we have argued is the metric of interest for creating a downstream dataset, we found that logistic regression outperforms other models, including a number of deep learning architectures. Finally, we reported that the COVID-19 conversation by Canadians on Twitter has a high prevalence of opinions but comparatively few mentions of the effects of the vaccine.

An interesting avenue for future work would be to reproduce the state of the art results for stance detection and sentiment analysis, but using an opinion detection step before the final prediction task. Moreover, there is an opportunity for refining the use of modern NLP architectures on the opinion detection problem specifically, but it would first require a standard benchmark dataset to be provided to the community.

In conclusion, we hope to have argued for the benefit to applied research provided by isolating the opinion detection problem. Improving on the state of the art for this task would directly benefit many popular challenges such as sentiment analysis and stance detection. As a result, applied models that aim to directly support decision making will be able to focus on solving the phenomenological problem instead of working on two challenging tasks at the same time.

Bibliography

[1] AGGARWAL, A., GOLA, B., AND SANKLA, T. Data mining and analysis of reddit user data. In *Cybernetics, Cognition and Machine Learning Applications*. Springer, 2021, pp. 211–219.

[2] ALDAYEL, A., AND MAGDY, W. Assessing sentiment of the expressed stance on social media. In *International Conference on Social Informatics* (2019), Springer, pp. 277–286.

[3] ALDAYEL, A., AND MAGDY, W. Stance detection on social media: State of the art and trends. *Information Processing & Management 58*, 4 (2021), 102597.

[4] BAIL, C. A., ARGYLE, L. P., BROWN, T. W., BUMPUS, J. P., CHEN, H., HUNZAKER, M. F., LEE, J., MANN, M., MERHOUT, F., AND VOLFOVSKY, A. Exposure to opposing views on social media can increase political polarization. *Proceedings of the National Academy of Sciences 115*, 37 (2018), 9216–9221.

[5] BOSER, B. E., GUYON, I. M., AND VAPNIK, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory* (1992), ACM Press, pp. 144–152.

[6] BREIMAN, L. Random forests. *Machine Learning 45*, 1 (Oct 2001), 5–32.

[7] BREIMAN, L., FRIEDMAN, J., STONE, C., AND OLSHEN, R. *Classification and Regression Trees*. Taylor & Francis, 1984.

[8] BRIDGMAN, A., MERKLEY, E., ZHILIN, O., LOEWEN, P. J., OWEN, T., AND RUTHS, D. Infodemic pathways: Evaluating the role that traditional and social media play in cross-national information transfer. *Frontiers in Political Science 3* (2021), 20.

[9] CHELBA, C., MIKOLOV, T., SCHUSTER, M., GE, Q., BRANTS, T., KOEHN, P., AND ROBINSON, T. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005* (2013).

[10] DAS, S., AND CHEN, M. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)* (2001), vol. 35, Bangkok, Thailand, p. 43.

[11] DAVE, K., LAWRENCE, S., AND PENNOCK, D. M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web* (New York, NY, USA, 2003), WWW '03, Association for Computing Machinery, p. 519–528.

[12] DERCZYNSKI, L., BONTCHEVA, K., LIAKATA, M., PROCTER, R., HOI, G. W. S., AND ZUBIAGA, A. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972* (2017).

[13] DEY, K., SHRIVASTAVA, R., AND KAUSHIK, S. Twitter stance detection — a subjectivity and sentiment polarity inspired two-phase approach. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (2017), pp. 365–372.

[14] DROR, A. A., EISENBACH, N., TAIBER, S., MOROZOV, N. G., MIZRACHI, M., ZIGRON, A., SROUJI, S., AND SELA, E. Vaccine hesitancy: the next challenge in the fight against covid-19. *European journal of epidemiology 35*, 8 (2020), 775–779.

[15] ERIKSSON, M., AND OLSSON, E.-K. Facebook and twitter in crisis communication: A comparative study of crisis communication professionals and citizens. *Journal of Contingencies and Crisis Management 24*, 4 (2016), 198–208.

[16] ESLAHI, M., SALLEH, R., AND ANUAR, N. B. Bots and botnets: An overview of characteristics, detection and challenges. In *2012 IEEE International Conference on Control System, Computing and Engineering* (2012), pp. 349–354.

[17] FAN, M., AND WU, G. Opinion summarization of customer comments. *Physics Procedia 24* (2012), 2220–2226. International Conference on Applied Physics and Industrial Engineering 2012.

[18] GARDNER, M., GRUS, J., NEUMANN, M., TAFJORD, O., DASIGI, P., LIU, N. F., PETERS, M., SCHMITZ, M., AND ZETTLEMOYER, L. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 1–6.

[19] GEVA, M., GOLDBERG, Y., AND BERANT, J. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *arXiv preprint arXiv:1908.07898* (2019).

[20] HAJMOHAMMADI, M. S., IBRAHIM, R., AND SELAMAT, A. Graph-based semi-supervised learning for cross-lingual sentiment classification. In *Intelligent Information and Database Systems* (Cham, 2015), N. T. Nguyen, B. Trawiński, and R. Kosala, Eds., Springer International Publishing, pp. 97–106.

[21] HATZIVASSILOGLOU, V., AND MCKEOWN, K. Predicting the semantic orientation of adjectives. In *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics* (1997), pp. 174–181.

[22] HEMMATIAN, F., AND SOHRABI, M. K. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review 52*, 3 (Oct 2019), 1495–1545.

[23] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[24] HOFMANN, T., SCHÖLKOPF, B., AND SMOLA, A. J. Kernel methods in machine learning. *Annals of Statistics 36*, 3 (2008), 1171–1220.

[25] HU, M., AND LIU, B. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining (New York, NY, USA, 2004), KDD '04, Association for Computing Machinery, p. 168–177.

[26] IGARASHI, Y., KOMATSU, H., KOBAYASHI, S., OKAZAKI, N., AND INUI, K. Tohoku at semeval-2016 task 6: Feature-based model versus convolutional neural network for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (2016), pp. 401–407.

[27] JENI, L. A., COHN, J. F., AND DE LA TORRE, F. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction* (2013), IEEE, pp. 245–251.

[28] KALCHBRENNER, N., GREFENSTETTE, E., AND BLUNSOM, P. A convolutional neural network for modelling sentences. *CoRR abs/1404.2188* (2014).

[29] KEITH, B., FUENTES, E., AND MENESES, C. A hybrid approach for sentiment analysis applied to paper reviews.

[30] KENYON-DEAN, K., AHMED, E., FUJIMOTO, S., GEORGES-FILTEAU, J., GLASZ, C., KAUR, B., LALANDE, A., BHANDERI, S., BELFER, R., KANAGASABAI, N., ET AL. Sentiment analysis: It's complicated! In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), pp. 1886–1895.

[31] KHAN, F. H., BASHIR, S., AND QAMAR, U. Tom: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems 57* (2014), 245–257.

[32] KHATUA, A., CAMBRIA, E., HO, S. S., AND NA, J. C. Deciphering public opinion of nuclear energy on twitter. In *2020 International Joint Conference on Neural Networks (IJCNN)* (2020), pp. 1–8.

[33] KIM, S.-M., AND HOVY, E. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics* (USA, 2004), COLING '04, Association for Computational Linguistics, p. 1367–es.

[34] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[35] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[36] LI, G., AND LIU, F. Sentiment analysis based on clustering: a framework in improving accuracy and recognizing neutral opinions. *Applied intelligence 40*, 3 (2014), 441–452.

[37] LOCHTER, J. V., ZANETTI, R. F., RELLER, D., AND ALMEIDA, T. A. Short text opinion detection using ensemble of classifiers and semantic indexing. *Expert Systems with Applications 62* (2016), 243–249.

[38] MAAS, A. L., DALY, R. E., PHAM, P. T., HUANG, D., NG, A. Y., AND POTTS, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 2011), Association for Computational Linguistics, pp. 142–150.

[39] MATTHIS, I. Sketch for a metapsychology of affect. *Int J Psychoanal 81 (Pt 2)* (Apr 2000), 215–227.

[40] MEDURU, M., MAHIMKAR, A., SUBRAMANIAN, K., PADIYA, P. Y., AND GUNJGUR, P. N. Opinion mining using twitter feeds for political analysis. *Int. J. Comput.(IJC) 25*, 1 (2017), 116–123.

[41] MIHALCEA, R., BANEA, C., AND WIEBE, J. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, Czech Republic, June 2007), Association for Computational Linguistics, pp. 976–983.

[42] MINSKY, M. L., AND PAPERT, S. A. Perceptrons: expanded edition, 1988.

[43] MOHAMMAD, S., KIRITCHENKO, S., SOBHANI, P., ZHU, X., AND CHERRY, C. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (2016), pp. 3945–3952.

[44] MOZER, M. C. A focused backpropagation algorithm for temporal. *Backpropagation: Theory, architectures, and applications 137* (1995).

[45] MUNEZERO, M., MONTERO, C. S., SUTINEN, E., AND PAJUNEN, J. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE Transactions on Affective Computing 5*, 2 (2014), 101–111.

[46] NEVIAROUSKAYA, A., PRENDINGER, H., AND ISHIZUKA, M. Textual affect sensing for sociable and expressive online communication. In *International Conference on Affective Computing and Intelligent Interaction* (2007), Springer, pp. 218–229.

[47] NEWMAN, N., FLETCHER, R., KALOGEROPOULOS, A., LEVY, D., AND NIELSEN, R. K. Reuters institute digital news report 2017. *Available at SSRN 3026082* (2017).

[48] NGUYEN, H. T., AND LE NGUYEN, M. Multilingual opinion mining on youtube–a convolutional n-gram bilstm word embedding. *Information Processing & Management 54*, 3 (2018), 451–462.

[49] NIRMALA, M., AND BABU, M. R. Analytic-based product opinion detection algorithm for twitter microblogging network. *INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS 33*, 13 (2020).

[50] PALEN, L., STARBIRD, K., VIEWEG, S., AND HUGHES, A. Twitter-based information distribution during the 2009 red river valley flood threat. *Bulletin of the American Society for Information Science and Technology 36*, 5 (2010), 13–17.

[51] PANDARACHALIL, R., SENDHILKUMAR, S., AND MAHALAKSHMI, G. S. Twitter sentiment analysis for large-scale data: An unsupervised approach. *Cognitive Computation 7*, 2 (Apr 2015), 254–262.

[52] PANG, B., AND LEE, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058* (2004).

[53] PANG, B., AND LEE, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr. 2*, 1–2 (Jan. 2008), 1–135.

[54] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A.,

YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[55] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[56] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[57] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[58] PORRECA, A., SCOZZARI, F., AND DI NICOLA, M. Using text mining and sentiment analysis to analyse youtube italian videos concerning vaccination. *BMC Public Health 20*, 1 (Feb 2020), 259.

[59] RAMESHBHAI, C. J., AND PAULOSE, J. Opinion mining on newspaper headlines using svm and nlp. *International Journal of Electrical and Computer Engineering (IJECE) 9*, 3 (2019), 2152–2163.

[60] RATHAN, M., HULIPALLED, V. R., VENUGOPAL, K., AND PATNAIK, L. Consumer insight mining: aspect based twitter opinion mining of mobile phone reviews. *Applied Soft Computing 68* (2018), 765–773.

[61] REZAEI-DASTJERDEHEI, M. R., MIJANI, A., AND FATEMIZADEH, E. Addressing imbalance in multi-label classification using weighted cross entropy loss function. In *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)* (2020), pp. 333–338.

[62] RIBEIRO, F. N., ARAÃºJO, M., GONÃ§ALVES, P., ANDRÃ© GONÃ§ALVES, M., AND BENEVENUTO, F. Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science 5*, 1 (Jul 2016), 23.

[63] ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review 65*, 6 (1958), 386.

[64] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[65] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning representations by back-propagating errors. *nature 323*, 6088 (1986), 533–536.

[66] SAAD, A. I. Opinion mining on us airline twitter data using machine learning techniques. In *2020 16th International Computer Engineering Conference (ICENCO)* (2020), IEEE, pp. 59–63.

[67] SAIF, H., HE, Y., FERNANDEZ, M., AND ALANI, H. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management 52*, 1 (2016), 5–19. Emotion and Sentiment in Social and Expressive Media.

[68] SALEH, M. R., MARTÍN-VALDIVIA, M. T., MONTEJO-RÁEZ, A., AND UREÑA-LÓPEZ, L. Experiments with svm to classify opinions in different domains. *Expert Systems with Applications 38*, 12 (2011), 14799–14804.

[69] SEVERYN, A., MOSCHITTI, A., URYUPINA, O., PLANK, B., AND FILIPPOVA, K. Multi-lingual opinion mining on youtube. *Information Processing & Management 52*, 1 (2016), 46–60. Emotion and Sentiment in Social and Expressive Media.

[70] SINDHU, C., SASMAL, B., GUPTA, R., AND PRATHIPA, J. Subjectivity detection for sentiment analysis on twitter data. In *Artificial Intelligence Techniques for Advanced Computing Applications* (Singapore, 2021), D. J. Hemanth, G. Vadivu, M. Sangeetha, and V. E. Balas, Eds., Springer Singapore, pp. 467–476.

[71] SINHA, K., PARTHASARATHI, P., PINEAU, J., AND WILLIAMS, A. Unnatural language inference. *arXiv preprint arXiv:2101.00010* (2020).

[72] SOKOLOVA, M., AND LAPALME, G. Learning opinions in user-generated web content. *Natural Language Engineering 17*, 4 (2011), 541–567.

[73] SOMASUNDARAN, S., AND WIEBE, J. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (2009), pp. 226–234.

[74] SUN, X., LIU, X., HU, J., AND ZHU, J. Empirical studies on the nlp techniques for source code data preprocessing. In *Proceedings of the 2014 3rd International Workshop on Evidential Assessment of Software Technologies* (New York, NY, USA, 2014), EAST 2014, Association for Computing Machinery, p. 32–39.

[75] TABOADA, M., BROOKE, J., TOFILOSKI, M., VOLL, K., AND STEDE, M. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics 37*, 2 (06 2011), 267–307.

[76] TANWANI, N., KUMAR, S., JALBANI, A. H., SOOMRO, S., CHANNA, M. I., AND NIZAMANI, Z. Student opinion mining regarding educational system using facebook group. In *2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)* (2017), IEEE, pp. 1–5.

[77] TAVOSCHI, L., QUATTRONE, F., D'ANDREA, E., DUCANGE, P., VABANESI, M., MARCELLONI, F., AND LOPALCO, P. L. Twitter as a sentinel tool to monitor public opinion on vaccination: an opinion mining analysis from september 2016 to august 2017 in italy. *Human Vaccines & Immunotherapeutics 16*, 5 (2020), 1062–1069. PMID: 32118519.

[78] THOITS, P. A. The sociology of emotions. *Annual Review of Sociology 15*, 1 (1989), 317–342.

[79] TWITTER. Q1 2021 letter to shareholders, Apr. 2021.

[80] WILSON, T., HOFFMANN, P., SOMASUNDARAN, S., KESSLER, J., WIEBE, J., CHOI, Y., CARDIE, C., RILOFF, E., AND PATWARDHAN, S. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations* (Vancouver, British Columbia, Canada, Oct. 2005), Association for Computational Linguistics, pp. 34–35.

[81] YADOLLAHI, A., SHAHRAKI, A. G., AND ZAIANE, O. R. Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput. Surv. 50*, 2 (May 2017).

[82] YOON, H. G., KIM, H., KIM, C. O., AND SONG, M. Opinion polarity detection in twitter data combining shrinkage regression and topic modeling. *Journal of Informetrics 10*, 2 (2016), 634–644.

[83] ZERVOUDAKIS, S., MARAKAKIS, E., KONDYLAKIS, H., AND GOUMAS, S. Opinionmine: A bayesian-based framework for opinion mining using twitter data. *Machine Learning with Applications 3* (2021), 100018.