Summer 8-9-2022

When Silver Is As Good As Gold: Using Weak Supervision to Train Machine Learning Models on Social Media Data

Venkata Rukmini Ramya Tekumalla

When Silver Is As Good As Gold:

Using Weak Supervision to Train Machine Learning Models on Social Media Data

by

Venkata Rukmini Ramya Tekumalla

Under the Direction of Juan M. Banda, Ph.D.

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2022

ABSTRACT

Over the last decade, advances in machine learning have led to an exponential growth in artificial intelligence i.e., machine learning models capable of learning from vast amounts of data to perform several tasks such as text classification, regression, machine translation, speech recognition, and many others. While massive volumes of data are available, due to the manual curation process involved in the generation of training datasets, only a percentage of the data is used to train machine learning models. The process of labeling data with a ground-truth value is extremely tedious, expensive, and is the major bottleneck of supervised learning. To curtail this, the theory of noisy learning can be employed where data labeled through heuristics, knowledge bases and weak classifiers can be utilized for training, instead of data obtained through manual annotation. The assumption here is that a large volume of training data, which contains noise and acquired through an automated process, can compensate for the lack of manual labels. In this study, we utilize heuristic based approaches to create noisy silver standard datasets. We extensively tested the theory of noisy learning on four different applications by training several machine learning models using the silver standard dataset with several sample sizes and class imbalances and tested the performance using a gold standard dataset. Our evaluations on the four applications indicate the success of silver standard datasets in identifying a gold standard dataset. We conclude the study with evidence that noisy social media data can be utilized for weak supervision

INDEX WORDS: Weak supervision, Noisy learning, Large scale data analysis, Social media data mining

When Silver Is As Good As Gold:

Using Weak Supervision to Train Machine Learning Models on Social Media Data

by

Venkata Rukmini Ramya Tekumalla

Committee Chair: Juan M. Banda

Committee: Rajshekhar Sunderraman

Rafal Angryk

Xiaojun Cao

Gerardo Chowell

Electronic Version Approved:

Office of Graduate Services

College of Arts and Sciences

Georgia State University

August 2022

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Dr. Juan M. Banda for his continued guidance and opening doors to a plethora of seemingly endless fascinating projects. His unassuming approach to research and science is a source of inspiration. He expanded my ability to think and enhanced many traits which I never thought I was capable of. His approach to problem solving is something I will never forget in my life. I am grateful and forever indebted to Dr. Rajshekhar Sunderraman, Associate chair for the Department of Computer Science at Georgia State University who has been a constant pillar of support since 2013 and for being a part of my committee. I would like to thank my committee members Dr. Rafal Angryk, Dr. Xiaojun Cao and Dr. Gerardo Chowell for investing their time and providing detailed feedback on my work. I acknowledge the generous financial support from the Department of Computer Science at Georgia State University and immensely value the teaching opportunities granted to me.

I am extremely grateful to have a family that unconditionally supported me in this journey. I would like to acknowledge my husband, Mr. Harsha Mallapragada, who has been a solid support system. His constant encouragement and in-depth knowledge in computer science helped me reach my goal. I will forever be grateful to my mom, Dr. Rama Devi Gade, who pushed me to pursue a doctoral degree and who will be the proudest when I confer the degree. All through my life, especially during the Ph.D. journey she has been my strength and inspiration. I thank my sister Dr. Sravya Tekumalla and my brother in law Mr. Chandrashekar Viswanathan for constant support and teaching me the ways of academic life. Finally I would like to acknowledge my grandparents, in-laws and extended family for their constant encouragement. My accomplishments and success are because they believed in me.

It is important to strike a balance with life outside the depths of the lab. I thank my friends Divya, Monica, Sindhu, Bharath, Mukesh, Sushmitha, Manoj who motivated and cheered me when I was in need and gave me a pass when I was least sociable.

I thank the Almighty who has blessed me with great health and spirit to pursue this difficult journey and the little butterfly who has not made my journey rough in the end.

Finally, I thank the makers of Friends, Big Bang Theory and music which kept me sane over the last four years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 PURPOSE OF STUDY

The primary purpose of this study is to explore the viability of using large-scale noisy social media data for weak supervision. This study aims to reduce labeling costs associated with supervised learning and move towards scalable approaches to generate training datasets. This study intends to compute the theoretical bounds of noisy learning and evaluate the accuracy of the bounds in an actual application.

2 CONTRIBUTION TO SCIENCE

The primary contribution of this study is to demonstrate the usage of noisy social media data for weak supervision through extensive evaluation. Furthermore, we contribute a feasible methodology that decreases labeling costs and generates large scale training datasets which can be adapted to a variety of applications.

3 INTRODUCTION

Weak supervision utilizes noisy, limited, or imprecise sources to provide supervision signals for labeling large amounts of training data in a supervised learning setting[1]. The following are a few ways in which training data can be obtained using weak supervision.

1. Obtaining cheaper, low quality labels from non-experts
2. Obtaining large noisy data through heuristics, distant supervision, constraints, expected distributions and invariances
3. Utilizing pre-trained models to provide supervision signal for data

Weak supervision enables these noisy labels to be combined programmatically to form the training data that can be used to train a model. Labels are considered "weak" because they are imperfect i.e., the labels are not accurate and might have a margin of error.

To decrease the labeling costs, researchers have been using weaker forms of supervision by heuristically generating training data with external knowledge bases, patterns/rules, or other classifiers[2]. In the early phases of applying weak supervision in research, authors induced noise (using a random probability) and flipped the labels of the gold standard training dataset and trained the classifiers[3]. However, in the past decade, researchers used noisy text and labeled the noisy text using weaker forms of supervision such as heuristics and constraints. Wang et al.[4] proposed a clinical text classification paradigm using weak supervision and deep representation to reduce manual annotation. Deriu et al.[5] utilized large amounts of weakly supervised data for multi-language sentiment classification. Agarwal[6] utilized a semi-automatic method to label training sets to create phenotype models in the field of medicine. Dehghani et al.[7] proposed a method to train neural networks with a large set of noisy data with weak labels and a small amount of data with true labels and applied the method on a sentiment classification task. Zamani and Croft utilized weak supervision for information retrieval[8]. Since 2017, weak supervision has been applied to several health applications like detection of intracranial hemorrhage (ICH)[9], classification of aortic valve malformations[10] and seizure detection in electroencephalography[11]. In computer vision, primitives like predicted bounding box or segmentation attributes from existing models have often been used to weakly supervise more complex image-based learning tasks[12–14]. In 2019, Weng et al. utilized weak supervision to infer complex objects and situations in autonomous driving data[15]. Khattar et al. applied weak supervision to time series data and programmatically labeled a dataset from wearable sensors. Their weakly supervised model matched performance with hand-labeled data[16].

3.1 Theory of Noise Learning

It is mathematically proven that addition of noise during the training of a neural network model has a regularization effect and, in turn, improves the robustness of the model[17]. However, the important question to envisage is how much noisy data is required to obtain a model with satisfactory performance. In the past, Vibhu et al.[6] computed the theoretical bounds to demonstrate an alternative to manual labeling for creating training sets for statistical models of phenotypes. Suchanek et al. utilized theoretical bounds to combine linguistic and statistical analysis to extract relations from web documents[18]. Kulkarni et al. discussed the use of theoretical bounds in pattern classification[19]. Several other researchers employed the sample bounds and created their own bounds for specific applications[20–22]. Simon[23] and later Aslam et al. [24] formulated the following theory as a sample complexity bound, given below:

D as the target data distribution consisting of observations and correct labels

D_n as the data distribution consisting of observations and noisy labels

τ as the random classification error for D_n

H as the class of learning algorithms to which our models belong

S as the set of m observations drawn from D_n

hˆ as a model in H and trained on S

h* as a model in H that best fits the target distribution D

ε(hˆ) as the generalization error of hˆ

ε(h*) as the generalization error of h*

Then for |ε(hˆ) - ε(h *)| ≤ γ, with probability 1 - δ, it suffices that

$\mathbf{m \geq O\ VC(H)}\quad {}^{\gamma(1-2\tau}\mathbf{)^{2} + \log(1/\delta)}\ {}^{\gamma(1-2\tau}\mathbf{)^{2}}$where $\gamma > 0$ and $0 \leq \delta \leq 1$

The case τ = 0 corresponds to observation data with clean labels, and the case τ = 0.5 represents the random flipping of labels that makes learning impossible. For a given error bound γ, probability 1 − δ, and classification error rate τ, a learning algorithm can learn equally well from approximately $\mathbf{m^{*}(1-2\tau)^{2}}$ observations of noisy data of what it can learn from **m** observations of clean data. The important aspect to note is, it is easier to obtain $\mathbf{m^{*}(1-2\tau)^{2}}$ noisy observations than to acquire m clean data. In this work, we calculated the theoretical bounds for each application where we determined the number of noisy samples required when **m** clean data is available and draw comparisons to results with noisy and clean samples.

To demonstrate a working example of calculating theoretical bounds, we considered the following hypothetical question. How many samples of noisy data do we require when a gold standard data of 1,000 samples is available?

To answer the hypothetical question, we require the following details: a) error bound (γ), b) probability (1 − δ), c) classification error rate (τ) and d) learning algorithm with accuracy score (A) to calculate classification rate (τ = 1-A). Since a machine learning algorithm can perform either exceptionally well or fail drastically, we calculated the minimum number of samples required for both a high performing and a low performing model.

3.1.1 Calculating theoretical bounds for a high performing model

In this computation, we consider “BERT” to be a model with high performance with an accuracy score of 95%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score(0.95), and clean samples ($m = 1,000$), the minimum number of noisy samples are calculated in the following way

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 1,000/(1-2*(1-0.95)))**2

noisy samples = 1,235

We would require 1,235 noisy samples to achieve the performance similar to the performance of models trained on 1,000 clean samples for a high performing model.

3.1.2 Calculating theoretical bounds for a low performing model

In this computation, we consider “Naive Bayes” to be a model with low performance with an accuracy score of 65%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score (0.65), and clean samples ($m = 1,000$), the minimum number of noisy samples are calculated in the following way

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 1,000/(1-2*(1-0.65)))**2

noisy samples = 11,112

We would require **11,112** noisy samples to achieve the performance similar to the performance of models trained on 1,000 clean samples for a low performing model. It is important to note that it is relatively easier to obtain 11,112 noisy samples than 1,000 clean samples.

3.2 Literature Review

There has been a shift towards relying on methods that use less to no data (E.g: Zero, One shot learning) or methods that can replace labeled data (Eg: Weak Supervision) due to cost associated with labeling the data. The trend of shifting towards relying on weak supervision has also been fueled by the recent empirical success of automated feature generation approaches. Notably, deep learning methods such as long short-term memory (LSTM) networks[25] ameliorate the burden of feature engineering when large labeled training sets are given. To help reduce the cost of training set creation, several frameworks have been built, designed and re-engineered to automate the process of labeling. Data Programming[26] was the first paradigm to be built in 2016 to create large training sets quickly. The same team designed Snorkel[2] in 2018, a first-of-its-kind system that enables users to train state-of-the-art models without hand labeling any training data. To deploy weak supervision at industrial scale, Snorkel DryBell was created[27]. The key to these paradigms was the design of heuristics. Large training sets could be created automatically by adopting heuristics and developing labeling functions that use the heuristics to label the dataset. In the beginning, the heuristics were manually developed by Subject Matter Experts (SMEs). As weak supervision gained popularity and was used on an industrial scale, there was a need to automate the process of generating heuristics that assign training labels to unlabeled data. Snuba[28] automatically generates heuristics using a small labeled dataset to assign training labels to a large, unlabeled dataset in the weak supervision setting. Bringer et al. designed Osprey[29], a weak-supervision system suited for highly imbalanced data, built on top of the Snorkel framework to support non-coders. With the increase of utilizing Weak Supervision for research in recent years, Zhang et al.[30] compiled “WRENCH”, which is a comprehensive benchmark for weak supervision. WRENCH includes a set of 22 real-world datasets which can be utilized for weak supervision. The datasets can be applied for several domains like chemical, biomedical, news for several tasks like classification, sequence tagging and question & answering. However, not even one social media dataset was included in this study, out of 22 datasets. This demonstrates the absence of benchmark social media datasets in weak supervision and there is an immense scope for expansion to include datasets which can be utilized for several different applications. While there has been a growth in the use of weak supervision over the years, there has also been an increase in study on label generation for training data. Makar et al.[31] suggested an approach to discourage shortcut learning by using auxiliary labels, and specify a set of distribution shifts across a robust model which is risk-invariant. Chen et al.[32] proposed a targeted relabeling methodology where the budget is split between labeling and building the label set using machine learning. Wang et al.[33] proposed a weighted feature agent and an updating mechanism to do contrastive learning by using the pseudo labels to bridge the gap between supervised and unsupervised learning for fine-grained classification.

While weak supervision is attaining popularity in several applications, research on its application using social media data is limited. There have been several studies[3,34,35] in the past which utilized noisy learning in conjunction with approximate learning or incomplete samples. The studies cited in Introduction and Literature review sections were either from the labs that created the frameworks for weak supervision or standalone works with no extensions. Hence, there is an immense scope for expansion where weak supervision can be utilized.

4 SOCIAL MEDIA

Social media is producing massive amounts of data at an unprecedented scale[36]. 4.62 billion individuals use social media globally, and 424 million more people have signed up since the beginning of the year[37]. According to a survey on social media usage worldwide, 2h 27m is the average daily amount of time spent on social media[37]. Several key interactions like, day to day communications, personal and professional relationships, expression of opinions, are presented via online interactions such as posts, comments, favorites, tags, likes, and links on social media. Interactions on social media leave traces in the form of data, which can be utilized for research[38]. The data on social media possesses unique qualities such, as

a. The data stream is close to **real time,** which benefits research on current issues

b. Large data on a global scale is **available**, which can be utilized to understand different perspectives on a similar topic

c. Since the data is available, it can be **reused** to reproduce or enhance research

d. The data is **noisy** and **unstructured** with misspellings, grammatical errors and poorly constructed sentences due to limitation of text

In this work, we used Twitter data for the experiments, since the data acquisition is relatively easier when compared to other social media platforms.

4.1 Advantages of using Twitter

Facebook and Twitter are the most popular social media platforms where most user interactions take place[39]. However, it is illegal to scrape data from Facebook due to terms and conditions[40]. While Reddit permits users to scrape all the available data from subreddits the exact subreddit must be known to extract data, which might not offer extensive coverage. On the contrary, Twitter allows for easier and efficient data extraction. As of 2021, Twitter contains 322.4 million users and generates 500 million tweets every day on an average[41], making it an attractive choice of social media platform to obtain data for research. Twitter data can be acquired in the several ways as listed below:

a. Obtain the data from Internet Archive[42] that contains the json objects of tweets. This data is a 1% sample of the tweets that Twitter releases for the users

b. Obtain 1% sample of tweets from Twitter directly using the Twitter API. This requires a Twitter developer account and must obtain keys from Twitter

c. Obtain only the tweets that are relevant to the research using keywords filter on Twitter API

d. Hydrate tweet IDs from publicly available datasets

To obtain tweets from Twitter streams or to hydrate tweets from publicly available datasets, a Twitter developer account is required. This developer account lets users access the Twitter API through which data can be collected. We used version 1 of the Twitter API for the data collection as our work started in 2019. Twitter released a new stable version (v2) in November 2021, which contains new features such as “ability to request specific objects and fields”, “new tweet create features” and “new and more detailed data objects”. Additionally, “academic research” access can be requested from Twitter, which would obtain access to even more data and advanced search endpoints. The newest Twitter API for Academic Research allows access to Twitter's real-time and historical public data with additional features and functionality that support collecting more precise, complete, and unbiased datasets. Pfeffer et al. demonstrated that Twitter's data endpoint v2 delivers better samples than the previously used endpoint v1.1[43]. While the application process is fairly easy, several restrictions are placed on the developer account. Failure to adhere to the restrictions will result in freezing or canceling the account, which will impact the data collection process. However, a Twitter developer account is not required to download tweets from the Internet Archive.

4.2 Twitter's Role in Academic Research

There is no accurate method to identify the total number of articles that utilize Twitter data for research. Since 2006, there have been a total of 631,600 articles on arxiv.org articles that have either used or analyzed Twitter data[44]. In the field of computer science and machine learning, Twitter, in particular has been used as a data source for several applications such as hate speech detection[45,46], sentiment analysis[47,48], identifying adverse pregnancy outcomes[49,50], symptoms associated with Covid-19[51,52], many-to-many crisis communications during disasters[53–56], usage of opioids[57], detecting depression symptoms[58], disease surveillance[59], chemotherapy analysis[60], quantifying mental health signals[61–63] and many other countless applications. Additionally, Twitter is also utilized at organizational level to communicate with users. For instance, public health organizations use Twitter to promote smoking prevention[64,65], oncologists use Twitter to share research findings and discuss treatment options[66]. In the artificial intelligence front, several machine learning approaches like volume analysis, time series analysis, classification, regression, clustering utilized Twitter data in applications.

5 METHODS

5.1 Data Acquisition from Twitter

In the last 10 years, there has been a shift towards relying on Twitter data for research. To ease the data acquisition process and to access the Twitter API, several libraries and frameworks were created. In python, there are several libraries like Tweepy[67], which can easily access the Twitter API, Twarc[68], which is famously used for hydrating twitter data and retrieving historic data. Several third party scripts are available on Github[69–72] to acquire data. Several researchers built their own in-house toolkits to acquire data which are application specific and would not work well with other applications. To address this issue, we created a Social Media Mining Toolkit (SMMT)[73], containing utilities for data acquisition, preprocessing, annotation and standardization. The data acquisition utility contains utilities to hydrate data, obtain data from the Twitter stream. The preprocessing utility contains utilities to preprocess the tweet text by removing hyperlinks, extra spaces, emojis and emoticons. The data annotation and standardization contains utilities to make automatic NER annotations on preprocessed tweets, plugins to use popular annotation tools and NER systems. Researchers will be able to obtain, use, and disseminate data in a uniform and transparent manner by using a standard toolkit, hence easing reproducibility and accessibility in the social media domain. We have employed several utilities of SMMT in this work to acquire, preprocess and label data. In this work, we collected Twitter data from three different sources, a) Internet Archive, b) Twitter Stream and c) Publicly available datasets.

5.2 Internet Archive

The Internet Archive (IA)[42] is a non-profit organization that builds digital libraries of Internet sites and other cultural artifacts in digital form and provides free access to researchers, historians, and scholars. The archive contains Twitter data collected using Twitter stream API, which yields a 1%

sample of daily tweets. This is the "Spritzer" version, the most light and shallow of Twitter grabs. This is the largest publicly available Twitter repository containing several json files of tweets in tar files sorted by date for each month of the year. To download and process the tweets disk space is essential since each month can take up to 700 GB of space. In order to download, a bash script was created which downloads tweets from all the days of a month. We downloaded tweets for each month and preprocessed the json file and created “tab separated value” (tsv) files for each day with relevant fields. Table 1 lists the details of the data available for each year from the Internet Archive. IA contains data from 2011 to 2020, however, we collected data from 2012 to 2018 for our applications since we started a longitudinal stream collection in 2018. The primary advantage of using IA over other sources is that the tweet json objects in IA are available and are never deleted, unless the repository has been removed. Since data is not lost, reproducible research is possible when using Internet Archive. The IA is also a very valuable source to obtain historic data since we cannot obtain large historic data using the Twitter end points and additionally the data is available for free. Since the files are stored on the web, a user can process only the required files and can delete the files from their local machine after a study reducing the need for large storage access. However, this method is time consuming since we have to re-process each file for each study if there are no storage options. It took us 190 days to process all the files from the Internet Archive from 2012 to 2018. The only disadvantage with IA is obtaining the latest tweets as the IA is only updated once in a few months.

Table 1. Internet Archive Data Collection Details

Year	**Total Tweets Available**
2012	1,245,785,016
2013	1,871,457,526

2014	1,086,859,898
2015	1,224,040,556
2016	1,427,468,805
2017	1,448,114,354
2018	1,102,507,263
Total	**9,406,233,418**

5.3 Twitter Stream Collection

While the Internet Archive contains historical data, it does not contain the latest tweets. So in order to obtain the current tweets, we started collecting 1% sample of the tweets from Twitter, yielding around 4 million tweets a day. We utilized the data acquisition tool from SMMT[73] to collect the data. We set up a python script that listens to the Twitter end point and collects tweets every day. A json file is created each day with 1 json tweet object per line. We process the json files weekly and create a "tab separated value" (tsv) file with several relevant fields like "tweet id", "tweet text", "date", "time", "language", "user id", "user name", "retweet status". Depending on the retweet status, we filter the files and store the clean files and retweet files separately. We used a bash script to multi process the files. We created a total of 1,139 clean tsv files for the data collected between 2018 and 2021 used in this study. It takes 35 minutes to process all the clean files when using 8 threads on the server. Table 2 lists the details of data collected between 2018 and 2021. We use only clean tweets in this study, however, we also include the total number of tweets available for each year.

Table 2. Twitter Stream Data Collection Details

Year	Total Tweets	Clean Tweets
2018	936,487,968	455,783,507

2019	1,180,731,480	570,157,502
2020	151,260,4381	777,863,405
2021	621,615,285	325,579,195
Total	**4,251,439,114**	**2,129,383,609**

There are a few limitations and disadvantages when using this type of data collection. Firstly, data collection must be continuous and any problems like server downtime or storage issues would terminate the collection process which increases gaps in data collection. Secondly, data loss cannot be recovered unless we have a copy of the tweet ids to hydrate the tweets. While hydration is a good data recovery strategy, 100% data can never be recovered as tweets cannot be hydrated when a tweet is removed or deleted by the user. Finally, it is difficult to collect data every day unless there is access to a server with massive storage. 3.5 years of data collection (01/2018-05/2021) required 5.3 TB storage.

5.4 Publicly Available Datasets

Twitter is heavily used as a data source in many studies to analyze and identify patterns. We identified 35 studies which not only utilized Twitter as their primary source of data, but also made their data publicly available, enabling reproducible research. These datasets are valuable since they have historic data, which are very difficult to obtain. We cannot obtain “Nepal Earthquakes” or “H1N1 pandemic data” from Twitter end points (version 1) since it is historic data. The publicly available datasets, while collected for a different purpose, still have the data signals relevant to events that happened during the collection period. We intend to build a longitudinal dataset for each application which contains tweets from the past. A huge advantage in using the past data is the ability to identify the shift or trends in data. For example, during natural disasters, hurricane Harvey tweets could be analyzed to identify commodities that are required during a crisis and can be easily adapted for future hurricanes of the same magnitude. The primary intent to use publicly available datasets is to re-use existing work and demonstrate an approach on how existing work can be utilized to build a superior dataset. Further, we observed that data augmentation improves the performance of machine learning models when we apply a heuristic to obtain more relevant tweets[74]. All the datasets from the 35 studies are collected based on keyword based search and contain significant noise. Since tweet texts cannot be shared publicly, all the studies released the tweet IDs corresponding to the tweets they have utilized in their study. The get_metadata utility of the SMMT[73] toolkit was employed to hydrate the tweet IDs. The number of tweets to hydrate per day depends on the type of Twitter developer account. Using an academic research developer account, we could hydrate 8,640,000 tweets per day. However, a tweet cannot be hydrated if the tweet is deleted either by the user or Twitter. The following table summarizes the details of tweets we hydrated using publicly available datasets. A total of 2,905,714,184 tweets were hydrated out of which we could hydrate only 1,357,409,820 tweets. 46.71% of tweets were lost since the Twitter user or Twitter deleted the tweet. A total of 336 days was required to hydrate the 2,905,714,184 (~2 billion) tweets. We pre-processed each dataset and extracted only relevant fields ("tweet id", "text", "language", "date" and "time") and stored all the extracted fields in a "tab separated value"(tsv). We used the processed tsv files for this study. It takes 55 minutes to process all the extracted files when using 8 threads on the server.

Table 3. Publicly Available Dataset Details

Dataset Name	Total Tweets	Total Hydrated Ids	Clean Tweets	Time taken (in days)
2016 presidential election[75]	283,244,653	122,799,810	50,788,341	33
Solar Eclipse[76]	13,816,206	8,345,117	1,537,247	2
Election 2012[77]	38,393,134	22,703,483	21,751,070	4
Datarelease[78]	106,116,957	38,912,028	30,799,490	12
Beyond the Hashtag[79]	40,815,855	23,137,993	7,307,037	5

Climate Change[80]	40,000,000	25,728,395	8,029,516	5
Trump Tweet Ids[81]	40,202,199	16,690,791	9,408,459	5
Health Care[82]	254,971,894	79,348,847	22,762,224	30
Women's March[83]	14,478,518	7,061,577	1,286,113	2
US Govt Ids[84]	9,673,959	9,085,817	6,933,491	1
End of Term[85]	5,655,632	5,288,040	4,116,967	1
Nipsey Tweets[86]	11,642,103	6,944,028	1,307,212	1
Winter Olympics[87]	13,816,206	8,336,254	1,530,613	2
Dallas Shooting[88]	7,146,993	3,683,170	1,224,715	1
News Outlets[89]	110,656,738	103,811,445	91,026,264	13
Charlottesville[90]	3,015,437	1,517,338	327,856	0
Twitter-Events-2012-2016[91]	147,055,035	80,675,871	35,454,578	17
Immigration Exec Order[92]	16,875,766	7,108,723	2,088,736	2
Irish news English tweets[93]	198,725,860	100,359,505	45,924,135	23
Black Lives Matter[94]	17,292,130	6,460,739	2,527,358	2
Tweets to Donald Trump[95]	583,890,932	227,909,402	175,277,501	68
HurricaneHarvey[96]	18,352,142	10,406,538	2,142,577	2
Hurricane Irma[96]	17,244,139	9,474,907	2,341,596	2
Hurricane Florence[97]	7,766,964	4,891,342	1,394,576	1
Hurricane Harvey[98]	7,041,866	4,433,003	883,466	1
115th U.S. Congress Tweet Ids[99]	2,041,399	1,919,544	1,528,001	0
2020 Presidential Election[100]	802,029,566	366,187,559	143,239,345	93
Hurricane Florence[101]	4,971,575	3,399,192	744,050	1
Hurricane Maria[102]	987,938	647,001	160,947	0
Hurricane Sandy[103]	14,915,897	8,101,431	5,144,820	2
Hurricane Dorian[104]	3,000,553	2,234,048	416,410	0
Hurricane Dorian[105]	9,186,117	6,549,744	1,723,639	1
2018 Congregational Election[106]	60,689,821	33,257,138	9,792,467	7
Health ATAM[107]	144,344,099	75,053,674	75,053,674	17
Epic Corpus[108]	30,651,626	27,903,463	27,903,463	3
Total	**3,080,709,909**	**1,460,366,957**	**793,877,954**	**359**

Storage is the primary disadvantage of this kind of data collection. We needed a total of 7 TB disk space to download and process the 35 studies used in this study. A second disadvantage is that since there is no procedure to identify the removed or deleted tweet ids from the list of tweet ids, we have to make an API call with all the tweet ids resulting in increased amount of hydration time.

While this data source obtains tweets, reproducing results is difficult since the majority of the data is lost.

Table 4 depicts the summary of the data collection. A total of 16,738,382,441 (16.7 billion) tweets were collected as part of this study. We used several subsets of the dataset in each of our applications, since the data was collected in a span of 3 years.

Table 4. Data Collection Summary

Dataset	Total tweets	Clean Tweets	Data collection duration
Internet Archive	9,406,233,418	4,003,116,709	2011-2018
Regular Stream	4,251,439,114	2,129,383,609	2018-2021
Publicly available datasets	3,080,709,909	793,877,954	2012-2020
Total	**16,738,382,441**	**6,926,378,272**	

5.5 Technical Details

For data collection, processing and running the experiments we used our lab server with the following configuration details. Our server is built with 2x Intel Xeon-Gold 6148 which contains 20 cores or 40 threads. 768 GB RAM was available to run files in parallel. 14.4TB Hard Disk Drive (HDD) was available which was primarily used to store files and 7.68 TB Solid State Drive (SSD) was used for data collection. Our server is also equipped with 7 NVIDIA Tesla V100 GPUs with 32 GB GDDR5 that has 640 Tensor cores and 5,120 CUDA cores. We used a bash script to run python scripts that either collect data or standardize data. We used the GPUs to run deep learning models in parallel and used the server CPU cores to run classical models. 20-60 days were required to complete each application based on the number of experiments.

6 MACHINE LEARNING

Machine learning is the study of computer algorithms that can improve automatically through experience and by the use of data[109]. Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system. The three categories are Supervised Learning, Unsupervised Learning and Reinforcement Learning. Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. The task of the machine is usually to group unsorted information according to similarities, patterns, and differences without any prior training of data. Reinforcement Learning is a type of machine learning technique that enables an algorithm to learn in an interactive environment by trial and error using feedback from its own actions and experiences. Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs[110]. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. In this work, we utilized supervised learning algorithms in a weak supervision setting, i.e. the training data is noisy and is a silver standard instead of a gold standard. Both classical and deep learning models are utilized for the experiments, and the details of the models used are expanded below.

6.1 Conventional or Classical Models

Conventional or classical machine learning algorithms are based on learning of systems by training set to develop a trained model. This pre-trained model is used to classify or recognize the test dataset111. To implement the classical models, the scikit-learn112 python library was used. For all the models, scikit-learn's TF-IDF vectorizer was used to convert raw tweet text to TF-IDF features and return the document-term matrix which is sent to the model. The classical models used in this work are detailed below. For the classical models, we utilized the “compute_class_weight” utility in scikit learn, which estimates class weights for unbalanced datasets.

6.1.1 Support Vector Machines

A Support Vector Machine (SVM)[113] is a discriminative classifier formally defined by a separating hyperplane. Given labeled training data, the algorithm outputs an optimal hyperplane that categorizes new examples. For the implementation of the SVM model, we used a LinearSVC, similar to SVC, but implemented using liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and scales better to large numbers of samples.

6.1.2 Logistic Regression

Logistic regression[114] (LR) is a statistical model that uses a logistic function to model a binary dependent variable. Logistic regression becomes a classification technique only when a decision threshold is available. For the implementation of this model, we used regularized logistic regression using the ‘lbfgs’ solvers.

6.1.3 Naive Bayes

Naive Bayesian[115] (NB) classifiers are Bayesian networks that make use of directed acyclic graphs containing only one unobserved (parent) node and several observed (children) nodes having an assumption of independence among them that is given by Naive Bayes independency model[116]. In this work, the multinomial Naive Bayes model was utilized which implements the naive Bayes algorithm for multinomial distributed data and is one of the two classic naive Bayes variants used in text classification.

6.1.4 Decision Trees

A decision tree[117] (DT) is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). Decision trees are used for classification by sorting the classes based on parameters values. ID3, C4.5, CHAID and CART are some algorithms belonging to the decision tree. A major advantage of this approach is that it is able to handle numerical, as well as categorical attributes. This method holds good for small datasets, but causes lagging for large datasets. In this work, the decision tree classifier uses a CART algorithm (Classification And Regression Tree) from scikit-learn. CART is a non-parametric decision tree learning technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively. However, the scikit-learn library uses an optimized version of the CART, which does not support categorical values.

6.1.5 Random Forest

A Random Forest[118] (RF) is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output, rather than relying on individual decision trees. Random forest has multiple decision trees as base learning models. It is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

6.2 Deep Learning Models

Deep learning[119] is a set of algorithms in machine learning that attempt to learn at multiple levels, corresponding to different levels of abstraction. It typically uses artificial neural networks. The levels in these learned statistical models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. Several deep learning models were implemented in the applications using Pytorch and Keras python libraries. Keras implementation of CNN model by Text Classification Algorithms: A survey[120] was utilized for implementing the keras models.

6.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were inspired by the visual system's structure[121], in which the architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the human brain. The algorithm[122] takes an input image, assigns importance (learnable weights and biases) to various aspects/objects in the image, and is able to differentiate one from the other. A CNN comprises three main types of neural layers, namely, (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers. CNNs have demonstrated exceptional results in image related tasks[12,123–125]. CNNs have been applied in text classification applications with remarkable results[126–128]. Adam Optimizer, Relu Activation function were used in the experiments.

6.2.2 Long Short Term Memory

Long Short-Term Memory (LSTM)[25] networks are a type of Recurrent Neural Network (RNN) capable of learning order dependence in sequence prediction problems. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data. Bidirectional LSTMs (BiLSTM) are an improvement on the LSTMs that present each training sequence forwards and backwards to two separate recurrent networks, both of which are connected to the same output layer. An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. Each block contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates, which provide continuous analogues of write, read and reset operations for the cells. BiLSTMs are used for the experiments with Adam Optimizer, max sequence length set to 280, dropout set to 0.2 and softmax activation function.

6.2.3 Word Embeddings

A word embedding model is representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word, such that the words that are closer in the vector space are expected to be similar in meaning[129]. To implement CNNs and LSTM models, several word embedding models were experimented with in each application. We utilized RedMed model[130], Glove embeddings[131] and Twitter Workd2Vec embeddings[132] for the applications. The model used and the details of the model are explicitly mentioned in each application.

6.3 Transformers

A transformer[133] is a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. With the evolution of transformer models, there has been a shift in using transformers in the deep learning models as transformers allow for significantly more parallelization and can reach a new state of the art in translation quality. To implement the transformer models, we utilized Simple Transformers[134], which seamlessly worked with the Natural Language Understanding (NLU) architectures made available by Hugging Face's Transformers models[135]. For the transformer models, early stopping techniques were employed. The models would cease the training process when there is no significant improvement in the performance. To select and optimize the hyperparameters, Optuna[136] framework was used. We assigned weight to the transformer models based on the proportion of negative samples in the training set. For example, for the 1:25 ratio, we assigned the weight for labels [1,0] as [1.0,0.04]. Several pre-trained transformer models were used and fine-tuned in our applications. The details of the models are presented in the following section.

6.3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT)[137] is a language representation model designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. The two important steps in BERT are *pre-training* and *fine-tuning*. The BERT is pre-trained on the BooksCorpus[138] (800M words) and English Wikipedia (2,500M words). The BERT-Large model, which has 24 layers, 1024 hidden size, 16 self-attention heads, and 340M total parameters was used in this study. Fine-tuning the BERT model is straightforward since the self-attention mechanism in the Transformer allows BERT to model many downstream tasks. In this work, we utilized silver standard datasets to fine-tune the BERT model.

6.3.2 BioBERT

Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT)[139] is the first domain-specific BERT based model pre-trained on biomedical corpora for 23 days on eight NVIDIA V100 GPUs. The BioBERT was trained on English Wikipedia (2.5B words), BooksCorpus[138] (800 M words), biomedical PubMed Abstracts (4.5B words), and biomedical PMC Full-text articles (13.5B words) using initial weights from the BERT. The BioBERT-base model, containing 12 layers, 768 hidden size, 16 self-attention heads, and 1 million parameters, was used in this study. In this work, we utilized silver standard datasets to fine tune the BioBERT model.

6.3.3 RoBERTa

A Robustly Optimized BERT Pretraining Approach (RoBERTa)[140] is a replication study of BERT pretraining, demonstrating the impact of hyperparameter selection during training. The modifications applied to the RoBERTa model over BERT are (i) training the model longer with bigger batches using more data (ii) removing the next sentence prediction objective (iii) training on longer sequences and (iv) dynamically changing the masking pattern applied to the training data. The architecture used in RoBERTa are 24 layers, 1024 hidden size, 16 self attention heads and 355M parameters. Five English-language corpora (BookCorpus[138], CCNews[141], OpenWebText[142] and Stories[143]) of varying sizes and domains, totaling over 160 GB of uncompressed text has been utilized for pre-training. In this study, we utilized silver standard datasets to fine tune the RoBERTa model.

6.3.4 DisasterBERT

Disaster-Tweet-Bert[144] is a pre-trained language representation model, which is trained on disaster tweets such as road accidents, emergencies during natural disasters and man-made disasters. This model is uploaded to Hugging Face and contains 12 layers, 768 hidden size and 12 self-attention heads. In this study, we utilized silver standard datasets to fine tune the DisasterBERT model.

6.3.5 BERTweet

A pre-trained language model for English Tweets (BERTweet)[145], is the first public large-scale pre-trained language model for English Tweets. BERTweet contains the same architecture as RoBERTa and is pre-trained on 850M Tweets collected from Internet Archive from 01/2012 to

08/2019. In this work, the BERTweet-base model was utilized containing 12 layers, 768 hidden size, 12 self-attention heads and 135M parameters.

7 OUTLINE OF RESEARCH

In this study, we utilize a heuristic based approach to label data and generate silver standard datasets from social media data. Our data acquisition process is listed in chapter 3, where we collect 16 billion tweets from three different sources to use for our applications. In the first application, “Identifying drug mentions from Twitter”, we curate a drug dictionary as our heuristic, generate a silver standard dataset and train several machine learning models in a binary classification setting. In our second application, “Characterizing different types of Natural Disasters: Hurricanes, Earthquakes, Floods”, we curate a heuristic from past natural disasters and create a silver standard dataset and train various machine learning models in a binary classification setting. In our third application, “Detecting epidemic tweets and evaluation of large scale epidemic corpus”, we use regular expressions as our heuristic and test the weak supervision approach in both binary and multi-classification settings. In our final application, “Separating health related Twitter chatter”, we present a methodology to curate a “pseudo gold standard dataset” and use keywords as our heuristic to create the silver standard dataset. We evaluate the silver standard dataset in a multi classification setting. We experiment with several training samples, class imbalances and evaluate the results for each application. We compute the theoretical bounds as detailed in Chapter 3 and verify the certainty of theoretical bounds in each application (Chapter 8-11). We summarize the results of applications in Chapter 12 and list the limitations of work in Chapter 13. We identify possibilities for future work in Chapter 14 and finally conclude the study in Chapter 15.

8 APPLICATION 1: IDENTIFYING DRUG MENTIONS FROM TWITTER

Data which contains drug usage, side effects or beneficial information is very difficult to curate and is not easily available due to the sensitive nature of the data. Twitter contains an abundance of drug data as users tend to share their experience on social media[146]. In the past, researchers have acquired data from Twitter using a keyword based search. Leaman et al.[147] utilized only 4 drugs to gather comments from Daily Strength[148], to extract adverse drug reactions from user posts. Sarker and Gonzalez[149,150] employed 250 drug keywords to obtain tweets from Twitter and released a drug chatter language model, which aids research in pharmacovigilance. In the past, SMEs were consulted for obtaining the drug keywords. While this approach obtains relevant data, it is heavily reliant on the number of keywords. In Twitter, the queries required to pull the stream are restricted to 500 characters, resulting in using less number of keywords. Consequently, this ensues:

a. working with data acquired with lesser number of keywords, limiting the breadth of the research

b. working with lesser data due to non-availability of data from keywords

c. Devoting an increased amount of time to obtain more data

While the intent of the aforementioned studies was to prove the credibility of automated methods using machine learning, none of them experimented with a large lexicon. Health Processing lab[151] at University of Pennsylvania is the largest research contributor in the field of pharmacovigilance and have released most of their annotated and validated datasets for public research. In this application, we utilized a weak supervision approach to identify drug mentions in Twitter and created a silver standard dataset that aids Pharmacovigilance[152] applications. We first curated a silver standard dataset using a drug dictionary (i.e heuristic) and trained several machine learning models using the silver standard dataset in a binary classification setting. The silver standard dataset was used to train the models in a stratified ratio of 75:25 (train:validation), with 25% of the data used for either improving the model or terminating the training when there was no improvement after a predetermined number of training steps. Furthermore, we tested the model using the gold standard data and presented results on four different metrics, determining the performance of silver standard data in identifying the gold standard data.

8.1 Heuristic Curation

The heuristic for this application is designed to include a large number of drug terms to add extensive coverage. In this application, we used a drug dictionary curated using UMLS[153] as a heuristic. The RxNorm[154] vocabulary was utilized to obtain the drug terms. We only utilized the terms with language set to "English" as we utilized only English tweets in this work. Initially, we included five different term types from RxNorm that are listed in Table 5. After an initial analysis[155], we observed that all the term types are not required for this application. Since the dictionary was used on Twitter data and the total number of characters in a tweet were restricted to 140 (until 2017) and 280 (from October 2017), we eliminated all the strings of length less than or equal to 3 (too ambiguous) and greater than or equal to 100. This was due to a less likely chance for tweets to contain drug names that were as short as 3 characters or as long as 100 characters. Further, we removed strings such as "2,10,15,19,23-pentahydrosqualene" which are chemical compounds. After an initial analysis[155], we observed that the heuristic gathered irrelevant noise because of the colloquially used terms like "patch, bar soap, disk, foam". We applied the heuristic on 9 billion tweets. Post analysis[155], we removed all the terms with length greater than 38 characters, since the longest term tagged from 9 billion tweets was only 37 characters. Each row in the dictionary has a Concept Unique Identifier (CUI), which links terms with similar meanings.

The CUI is used in order to ensure that the meanings are preserved over time, regardless of the different terms that are used to express those meanings. All the strings have been converted to lowercase and trimmed of white spaces. The final heuristic is a dictionary, containing 19,643 terms. Table 6 presents a sample of rows from the drug dictionary.

Table 5. Term types and number of terms in each term type

Term Type	Example	No of Terms
Ingredients (IN)	Fluoxetine	11,427
Semantic Clinical Drug Component (SCDC)	Fluoxetine 4 MG/ML	27,038
Semantic Branded Drug Component (SBDC)	Fluoxetine 4 MG/ML [Prozac]	17,938
Semantic Clinical Drug (SCD)	Fluoxetine 4 MG/ML Oral Solution	35,112
Semantic Branded Drug (SBD)	Fluoxetine 4 MG/ML Oral Solution [Prozac]	20,003

Table 6. Sample from drug dictionary

Concept Unique Identifier (CUI)	Term
C0290795	adderall
C0700899	benadryl
C0025219	melatonin
C0162373	prozac
C0699142	tylenol

8.2 Generating the Silver Standard Dataset

To create the silver standard dataset, we utilized the drug dictionary to filter tweets from Internet Archive, Publicly available datasets and Regular Stream. We mined a total of 9.4 billion tweets from the Internet Archive and separated 4,908,922 (4.9 million) clean English drug tweets using the drug dictionary between 2012 and 2018. The data collection from the Internet Archive is presented in Table 7.

Table 7. Drug tweets filtered from Internet Archive

Year	**Total tweets**	**Filtered Tweets**
2019	489,560,143	563,498
2018	1,102,507,263	511,634
2017	1,448,114,354	687,585
2016	1,427,468,805	856,515
2015	1,224,040,556	792,810
2014	1,086,859,898	592,260
2013	1,871,457,526	852,349
2012	1,245,785,016	52,271
Total	**9,895,793,561**	**4,908,922**

8.2.1 Publicly Available Datasets

We utilized 25 publicly available datasets in this application. We filtered 1,571,365 clean English drug tweets from 1,953,230,363 tweets. This demonstrates that data can be found in datasets which are outside the scope of this application domain. While we hydrated 1.9 billion tweets from publicly available datasets, only 0.12% of the tweets were filtered for this application. Table 8 lists the details of the drug tweets found in publicly available datasets.

Table 8. Drug tweets from publicly available datasets

Dataset	**Filtered tweets**	**Percentage of filtered tweets**
2016 presidential election[75]	181,943	0.11
Trump Tweet Ids[81]	19,960	0.10
Women’s March[83]	1,494	0.01
Winter Olympics[87]	2,636	0.03
US Govt Ids[84]	60,250	0.64
Solar Eclipse[76]	4,374	0.05

Social Sensor[156]	2,324	0.09
Nipsey Tweets[86]	1,470	0.02
News Outlets[89]	208,630	0.54
Immigration Exec Order[92]	8,681	0.08
hurricaneHarvey_irma[96]	8,101	0.04
Hurricane Florence[97]	6,492	0.11
Tweets to Donald Trump[95]	246,265	0.06
Hurricane Harvey[98]	2,493	0.05
Irish English News[93]	278,687	0.24
End of Term[85]	34,391	0.63
Election2012[77]	47,568	0.20
Dallas Shooting[88]	1,356	0.03
Datarelease[78]	181,724	0.30
Beyond the Hashtag[79]	12,869	0.05
Black Lives Matter[94]	3,313	0.04
Climate Change[80]	69,348	0.24
Twitter-Events-2012-2016[91]	81,865	0.09
Health Care[82]	103,787	0.06
Charlottesville TweetIds[90]	1344	0.03
Total	**1,571,365**	**0.12**

8.2.2 Regular Stream

In this application, we utilized tweets from the regular stream between 2019-10-06 and 2020-10-31. We applied the drug dictionary on our regular stream and filtered a total of 810,628 tweets from 773,059,908 clean tweets. Table 9 presents the details of stream collection totals for this application. Since we used Internet Archive tweets until 09-2019, we utilized tweets from 10-2019 from this collection to avoid duplicates.

Table 10 summarizes the details of all the data collected. A total of **13,406,947,422** (13.4 billion) tweets were mined for this application combining all our data collection methods. Only clean English tweets were preprocessed which resulted in 7,290,915 drug tweets. However, there is an overlap in the data collection which resulted in duplicate data and the reasons are listed below.

1. Overlap between similar data sets - Most studies utilized Twitter to collect tweets using keywords from the 1% sample which would narrow down the collection stream to relevant tweets. However, there is a good chance of having similar tweets for two different topics, if their search criteria had common keywords. For example, in the Publicly available datasets, there is an overlap in tweets between 2016 Presidential Election and Tweets to Donald Trump.
2. Overlap in the time frame of collection - Few of the tweets from publicly available datasets (2016 presidential election, Hurricane Tweets) overlap with Internet Archive (2011-2018) tweets because they were collected during the same time.

Table 9. Drug tweets from Regular Stream

Year	Filtered Tweets	Percentage of filtered tweets
2019 (Oct - Dec)	162,255	0.11
2020 (Jan - Oct)	648,373	0.10
Total	**810,628**	**0.10**

Table 10. Summary of data collection

Data Collection	Collection Period	Total number of tweets	Number of filtered tweets
Internet Archive	01/2011 - 09/2019	9,895,793,561	4,908,922
Regular Stream	10/2019 - 10/2020	1,557,923,498	810,628
Publicly Available Datasets	01/2012 - 12/2017	1,953,230,363	1,571,365
Total		**13,406,947,422**	**7,290,915**

We removed duplicate tweets between the datasets. The silver standard dataset consists of **7,007,551** clean English drug tweets. Figure 1 presents the top 10 drug terms in the silver standard dataset. Listed below are a few samples of the preprocessed tweets obtained through heuristics. The drug terms are highlighted in bold.

1. “health **melatonin** and exercise key combination for helping with alzheimers”
2. “i hate having breathing problems where i have to take up to 2-3 **xanax** at once just to slow down my heart beat”
3. “hopefully this **tylenol** breaks my fever”

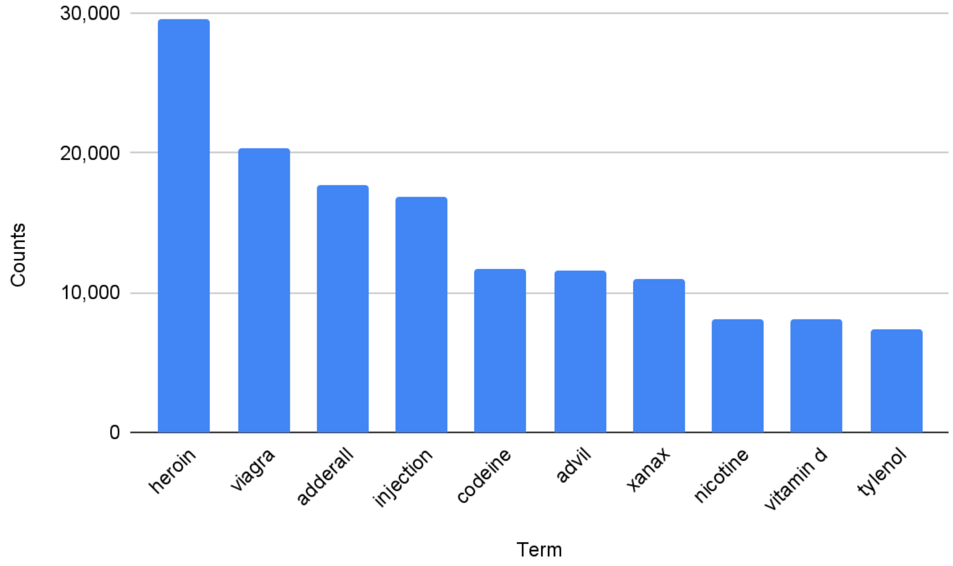


Figure 1. Top 10 drug terms in the silver standard dataset

8.3 Calculating Theoretical Bounds

To compute the theoretical bounds, we trained several machine learning models on the gold standard data and presented the theoretical bounds for a high and low performing model. We split the gold standard data into 75:25 for training and test and obtained the accuracy of the machine learning models. We use accuracy to calculate the theoretical bounds.

8.3.1 Calculating theoretical bounds for a high performing model

In this computation, we consider "BERT" to be a model with high performance with an accuracy score of 99%. For an error bound(γ = 0.05), probability(δ = 0.05), accuracy score(0.9978), and clean samples (m=14,430), the minimum number of noisy samples are calculated in the following way

noisy samples = m/(1-(2*(1-τ)))**2

noisy samples = 14,430/(1-2*(1-0.9978)))**2

noisy samples = 14,458

We would require **14,458** noisy samples to achieve the performance similar to the performance of models trained on 14,430 clean samples for a high performing model.

8.3.2 Calculating theoretical bounds for a low performing model

In this computation, we consider "Decision Tree" to be a model with low performance with an accuracy score of 79%. For an error bound(γ = 0.05), probability(δ = 0.05), accuracy score (0.79), and clean samples (m = 14,430), the minimum number of noisy samples are calculated in the following way

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 14,430/(1-2*(1-0.7930)))**2

noisy samples = 42,022

We would require **42,022** noisy samples to achieve the performance similar to the performance of models trained on 14,430 clean samples for a high performing model.

To summarize, the minimum number of noisy samples required for the best performing model (BERT) is **14,458** and the minimum number of noisy samples required for the least performing model (decision tree) with accuracy score(0.7930) is **42,022**.

8.4 Experimental Setup

To examine the performance of noisy data, we performed 7,700 experiments. We started the experiments with a class balanced ratio of 1:1 ratio i.e drug:non drug tweets and systematically increased the non-drug ratio all the way to 1,000, representing the realistic ratio of drug to non-drug tweets on Twitter. For each training ratio, we experimented with training size starting at 10,000 tweets and increasing it to 3,000,000 tweets. For example, in the 1:15 drug to non-drug tweets ratio, for training size with 1,000,000 samples, we trained the models with 66,667 drug tweets and 933,333 non drug tweets. A total of 7 training ratios (1:1, 1:5, 1:15, 1:25, 1:50, 1:100, 1:1000), 11 training sizes (10,000, 30,000, 50,000, 70,000, 100,000, 200,000, 300,000, 500,000, 1,000,000, 2,000,000, 3,000,000), 10 machine learning models (SVM, Naive Bayes, Random Forest, Decision Tree, Logistic Regression, BioBERT, BERT, RoBERTa, CNN and LSTM) and 10 seeds for each training size and ratio were used in the experiments. We used the silver standard dataset for training the models and labeled all the samples in the silver standard dataset as positive samples. For CNN and LSTM models, we used the RedMed[130] embedding model which was trained on 3M tokens from Reddit drug posts and contained 64 dimensions. We collected 3 million non-drug tweets and labeled them as negative tweets. A non-drug tweet is a tweet which does not match with any of our terms in the heuristic. As training and validation data, we employed a stratified ratio of 75:25 of the dataset. The validation data was utilized to obtain metrics to either improve the performance of the model or terminate the training using early stopping methods. We do not present any validation results since it was used to only enhance training steps. To test our models, we utilized a publicly available gold standard dataset which is detailed in the next section.

8.4.1 Gold Standard Data

To test the models, we collected publicly available manually and expertly curated datasets[146,157]. While the original dataset contains over 15,000 tweet IDs, we could only use 7,215 annotated drug tweets since we could not hydrate the tweet IDs, which were deleted. To these 7,215 annotated drug tweets, we added 7,215 non drug tweets to create a balanced gold standard dataset of **14,430** tweets. We emphasize that we did not manually annotate any tweets and instead used publicly available, manually and expertly annotated drug tweets in our test set. In this application, we evaluate the performance of the silver standard dataset in identifying the gold standard dataset in this application.

8.5 Results

In order to evaluate the performance, we used the following metrics: Precision (P), Recall (R), F-Measure (F) and Accuracy (A). For each training size, we used 10 seeds, which resulted in 10 experiments. Hence, in order to avoid bias and not show only the best results, we present the mean of 10 experiments in each training size. Figures 2-6 represent the performance of F-measure in classical models starting from sample size 10,000 to 1,000,000 samples for training rations 1:1 to 1:50. However, for the imbalanced training ratios (Eg: 1:25 and 1:50), the recall metric is more valuable than the precision metric. Figures 7-8 present the progression of the recall metric for training ratios 1:25 and 1:50 for classical models. Figures 9-13 illustrate the F-measure performance of deep learning models for each ratio. Figures 14-15 present the recall metric for the ratios 1:25 and 1:50 training rations for the deep learning models. All the precision and remaining recall metrics plots for both the classical and deep learning models are enclosed in the Appendix section. Additionally, for this application, we experimented with 1:100 and 1:1000 ratios as they represent the realistic ratio of drug mentions in Twitter[158]. However the results for 1:100 and 1:1000 are enclosed in the appendix as they are additional evaluations.

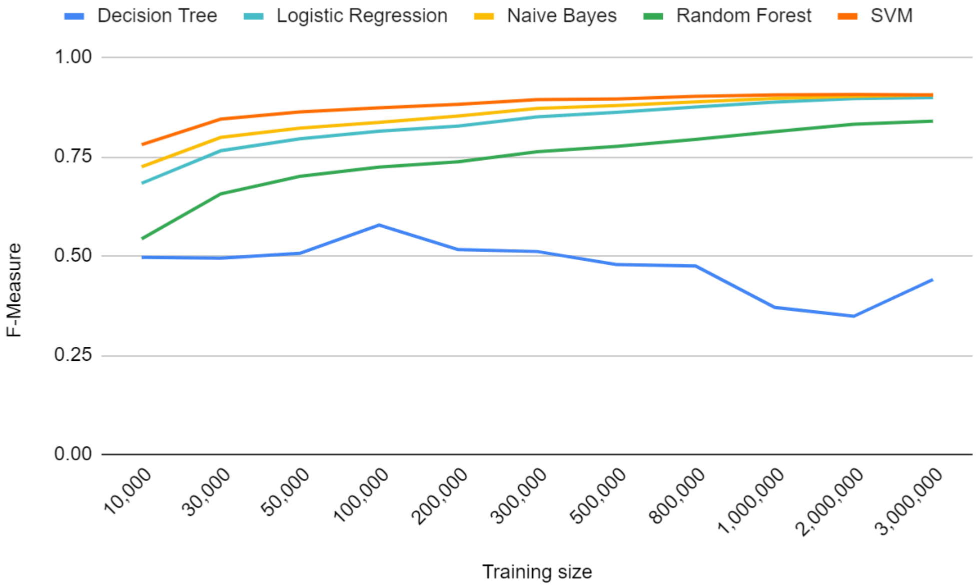


Figure 2. Classical models mean F-measure for 1:1 ratio

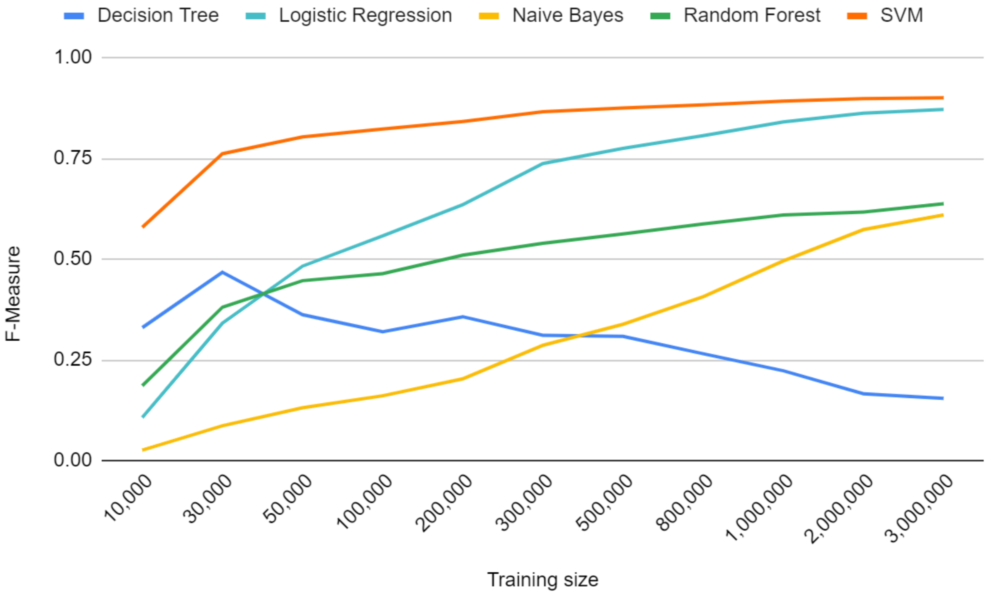


Figure 3. Classical models mean F-measure for 1:5 ratio

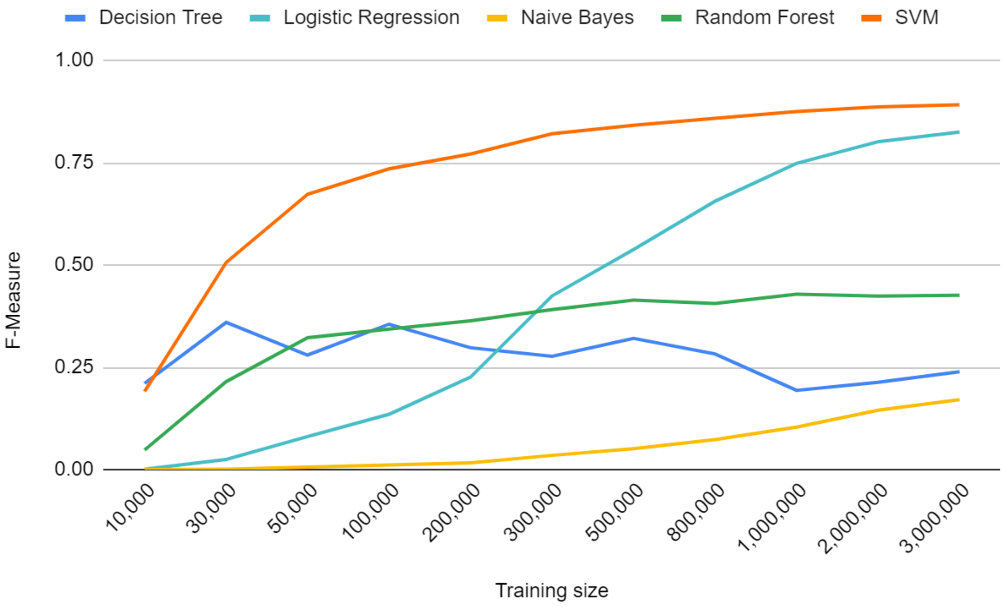


Figure 4. Classical models mean F-measure for 1:15 ratio

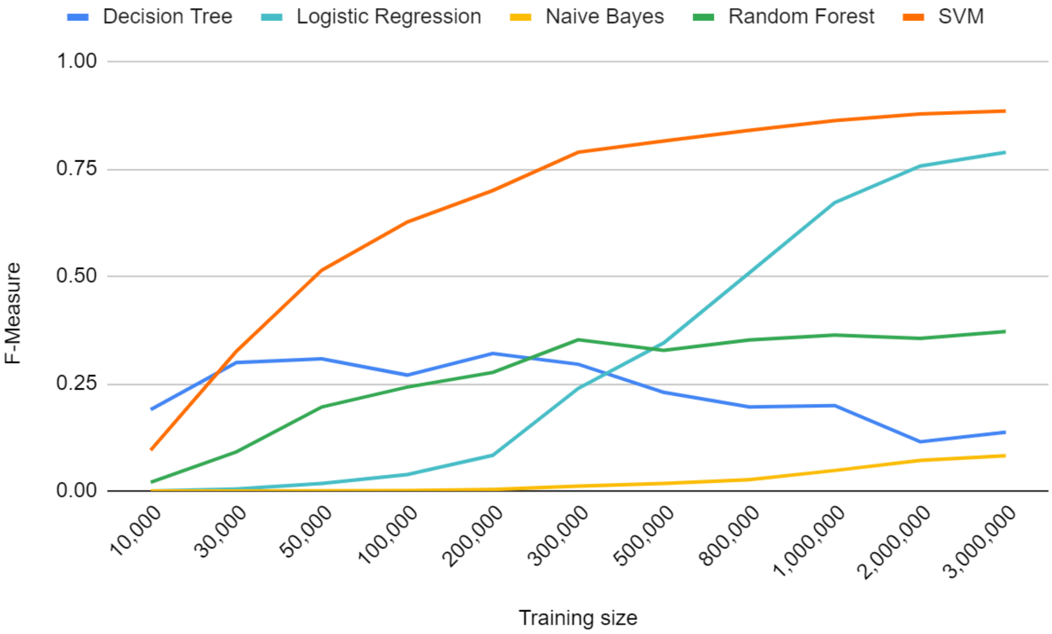


Figure 5. Classical models mean F-measure for 1:25 ratio

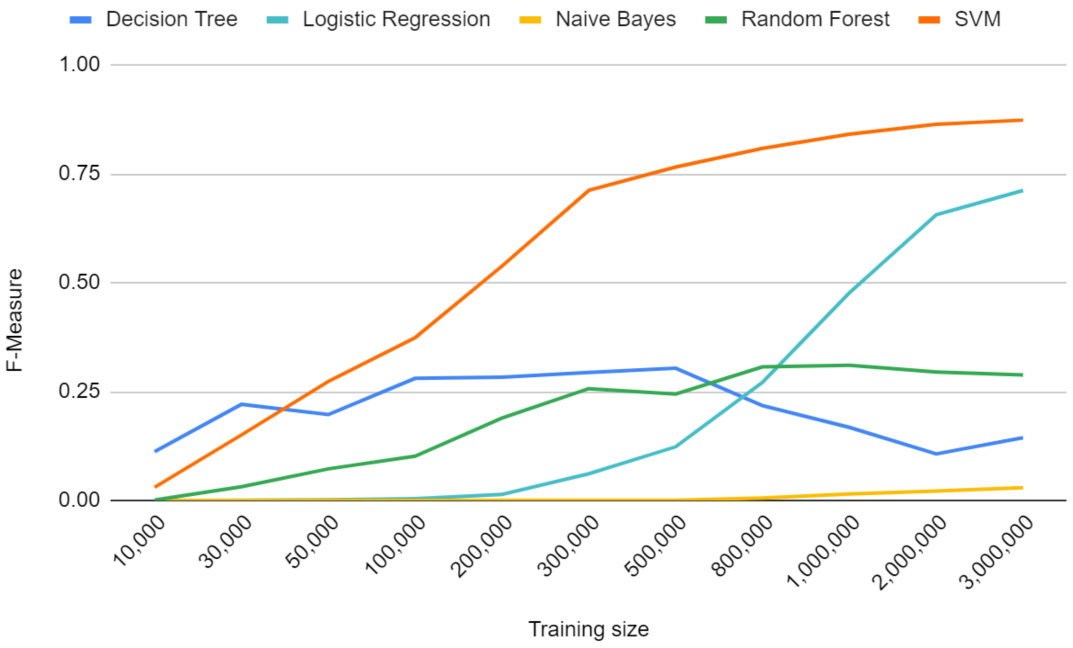


Figure 6. Classical models mean F-measure for 1:50 ratio

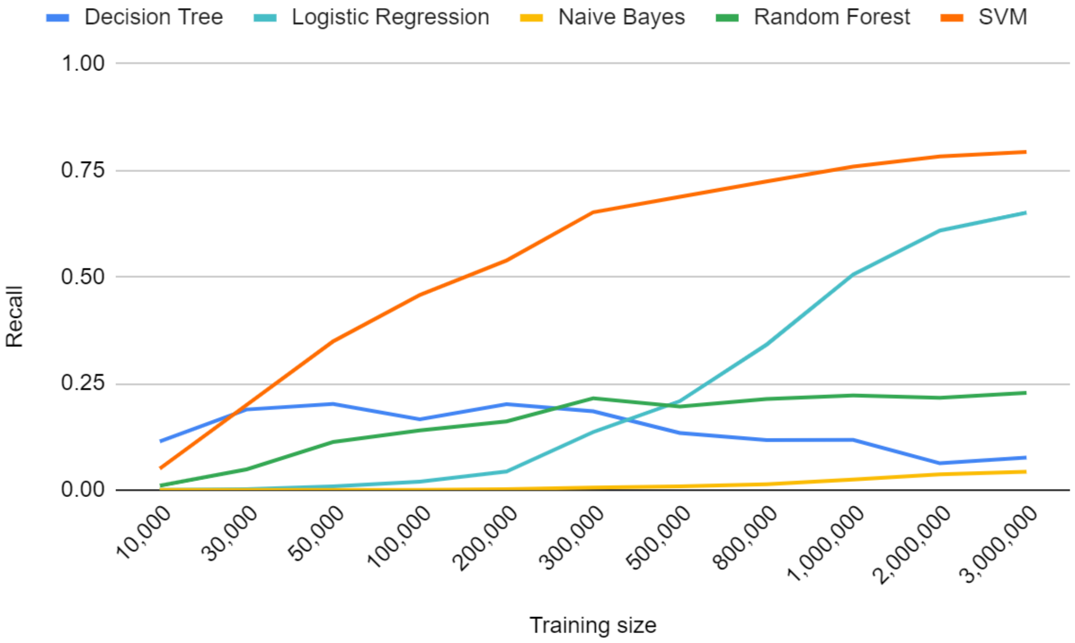


Figure 7. Classical models mean Recall for 1:25 ratio

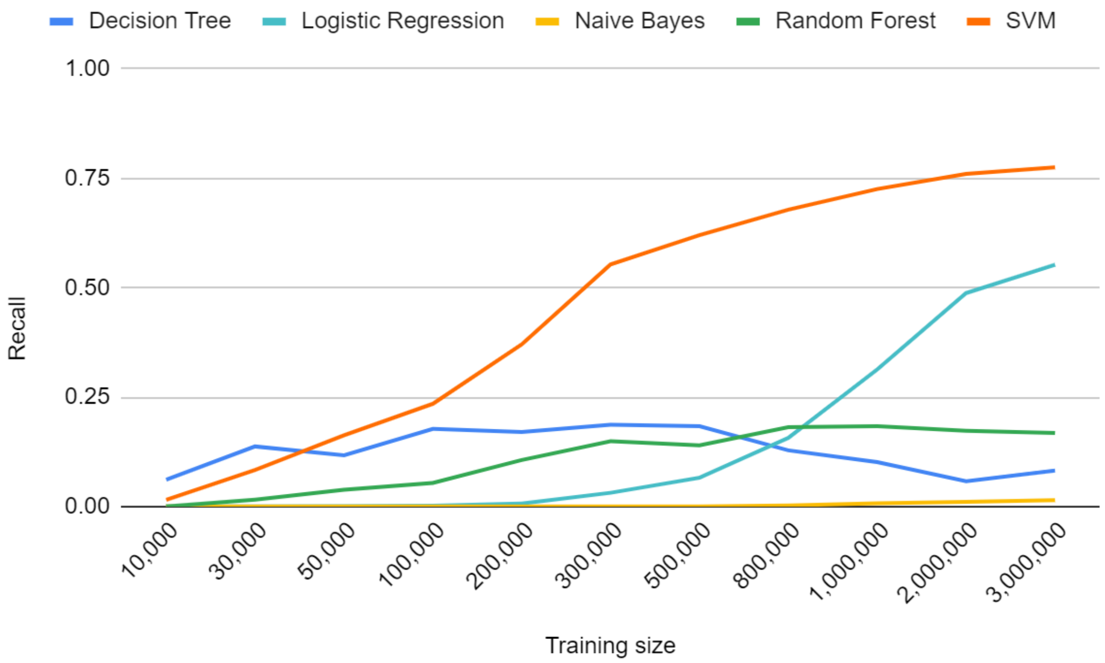


Figure 8. Classical models mean Recall for 1:50 ratio

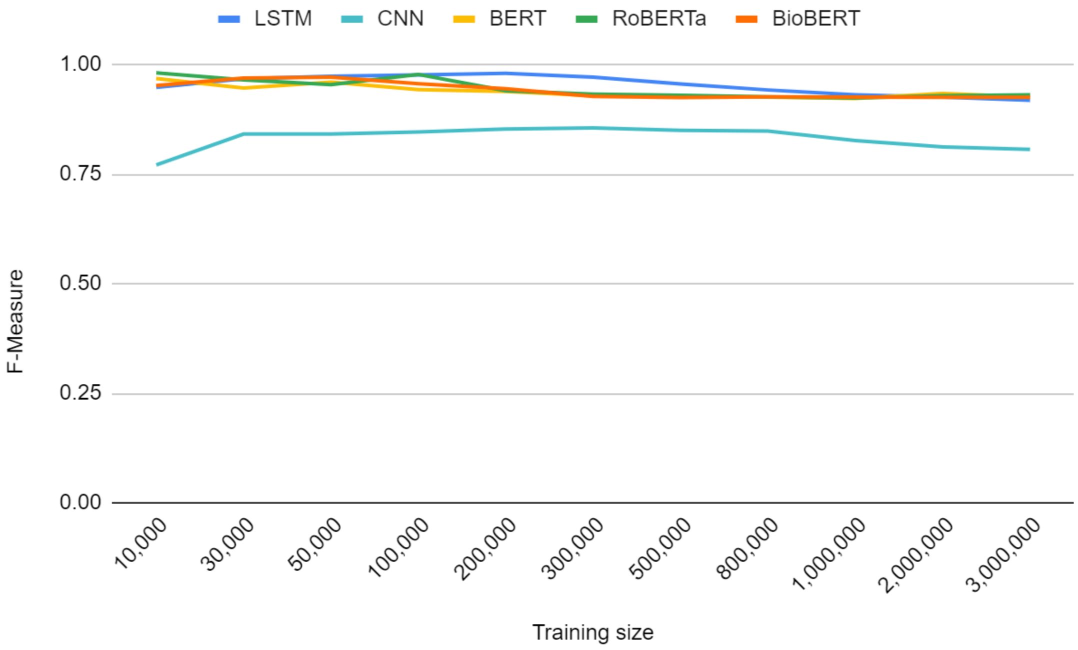


Figure 9. Mean of F-measure for 1:1 ratio for deep learning models

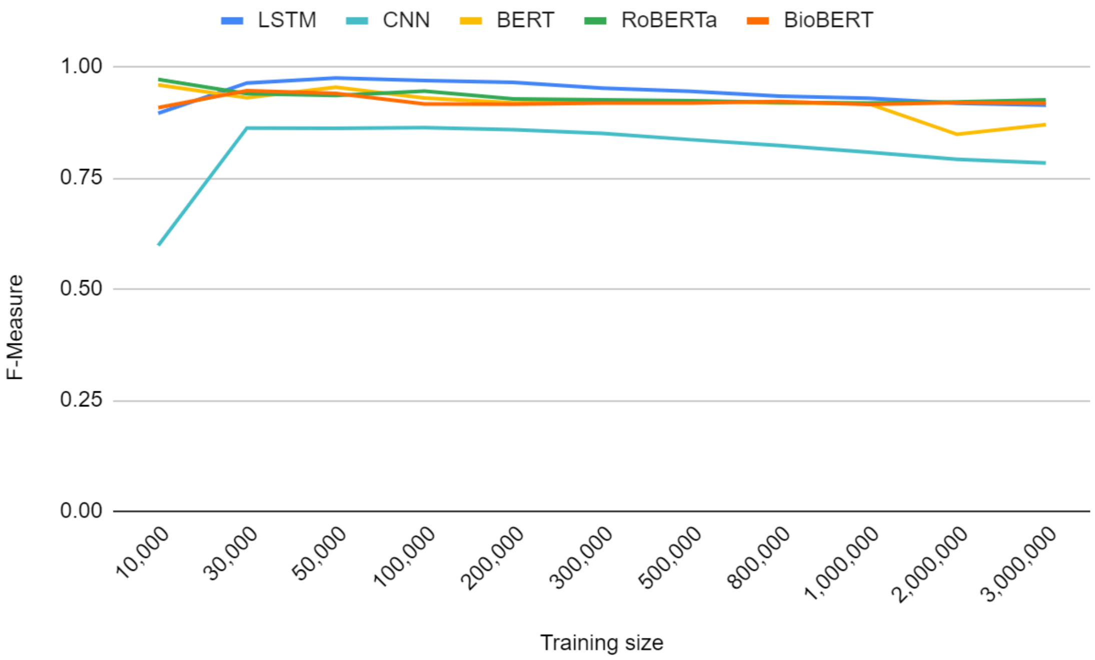


Figure 10. Deep learning models mean F-measure for 1:5 ratio

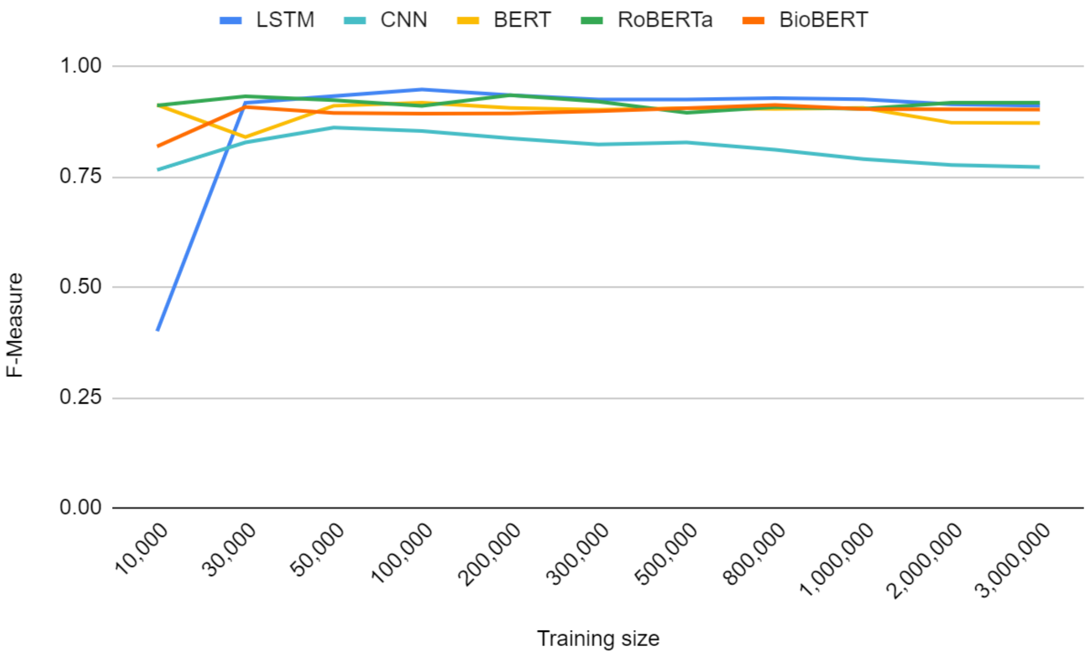


Figure 11. Deep learning models mean F-measure for 1:15 ratio

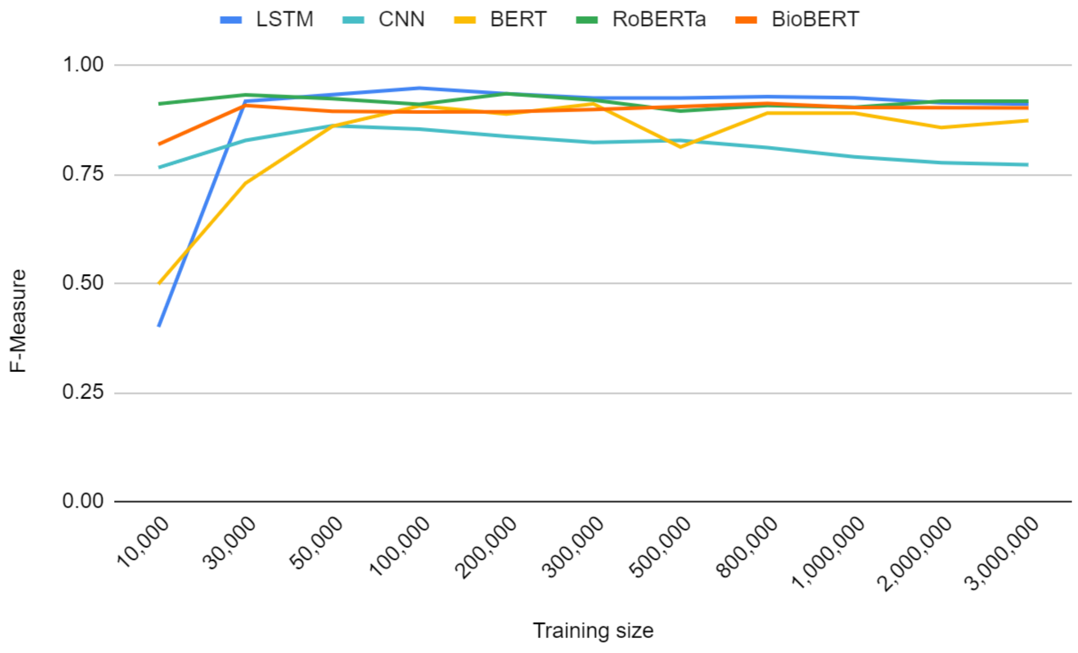


Figure 12. Deep learning models mean F-measure for 1:25 ratio

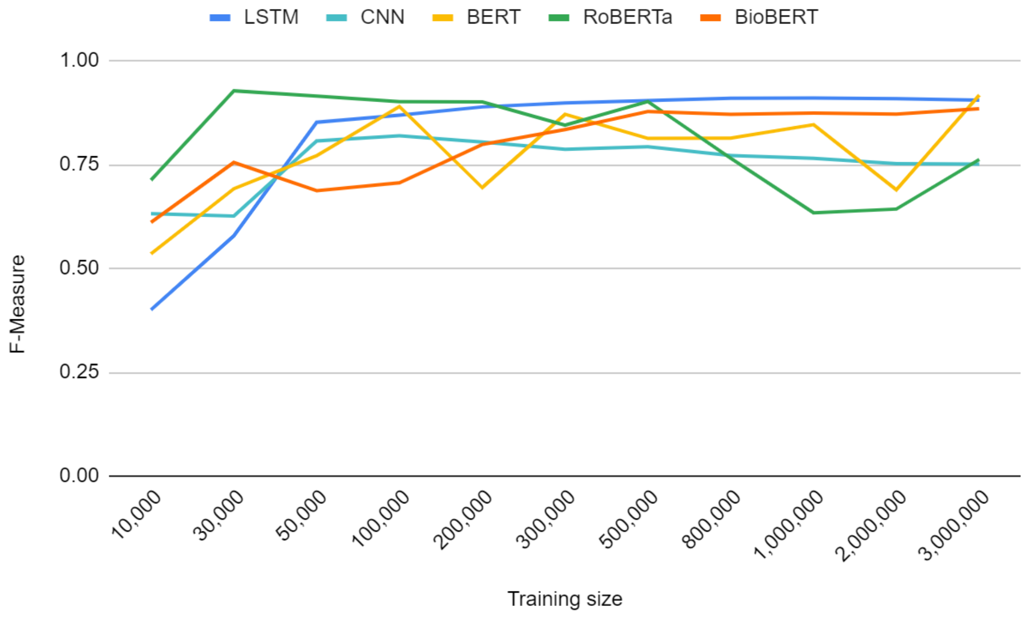


Figure 13. Deep learning models mean F-measure for 1:50 ratio

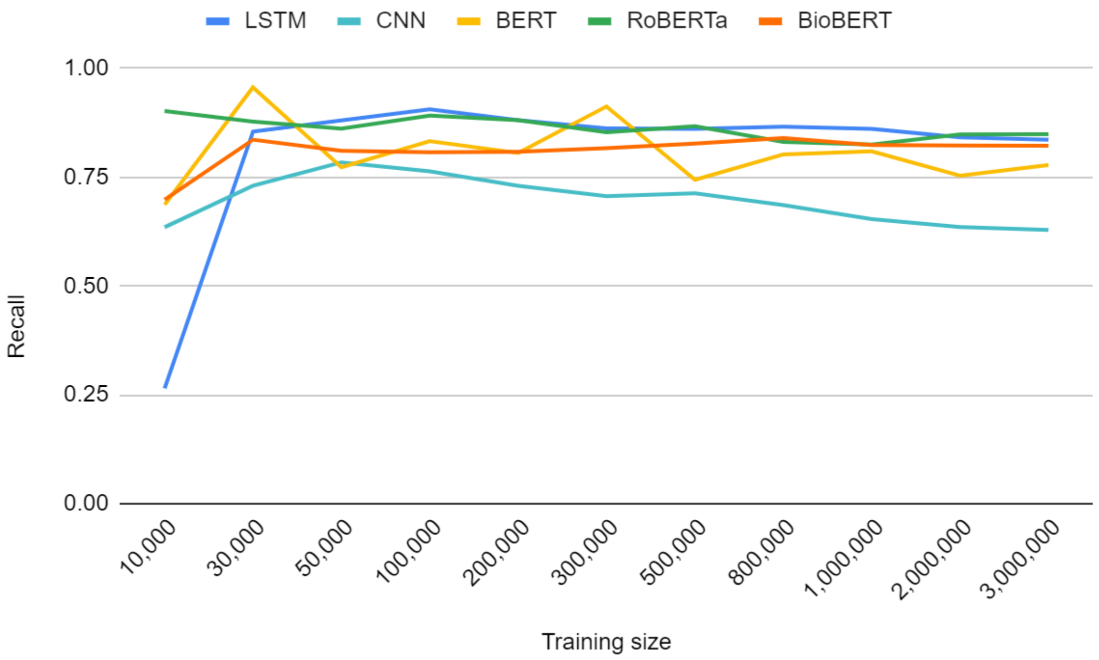


Figure 14. Deep learning models mean Recall for 1:25 ratio

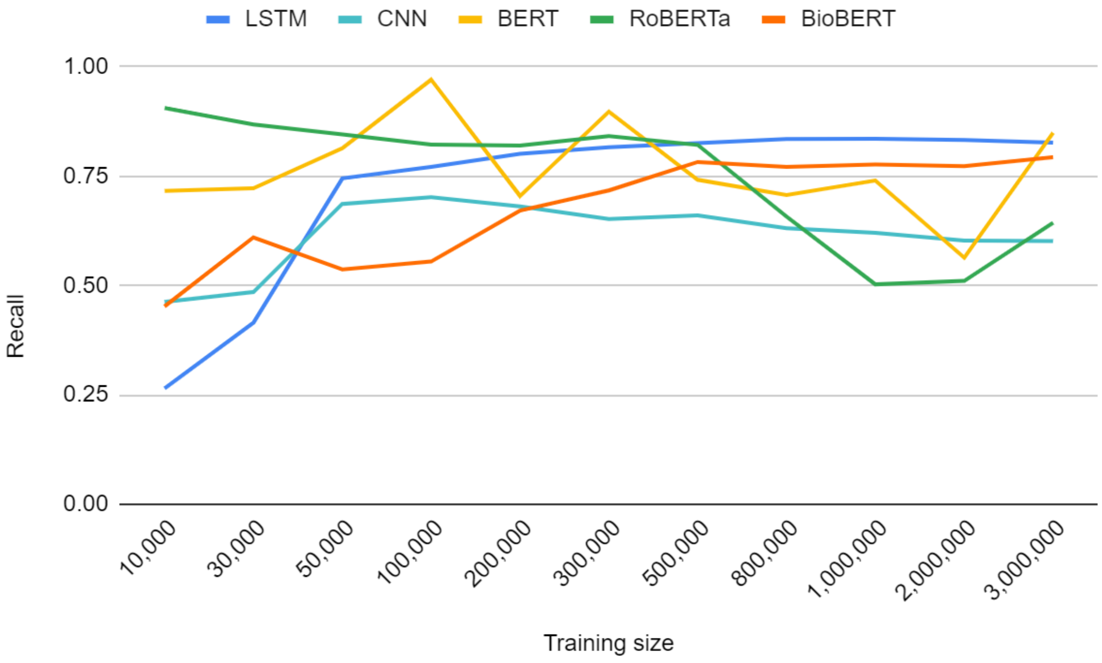


Figure 15. Deep learning models mean Recall for 1:50 ratio

On the classical models front, the SVM model outperformed all the other models by achieving the best performance when compared to other models in every training ratio. In the heavily imbalanced ratios (1:25 and 1:50), most classical models have consistently low performance. Surprisingly, SVM and Logistic Regression demonstrate an improvement in performance as the training size increases. This demonstrates that the models could learn drug signals despite heavy imbalances in the training data as the sample size increases.

Unsurprisingly, the deep learning models outperformed the classical models in the both balanced and heavily imbalanced ratios. Except for CNN, most models performed consistently (F-Measure > 0.85) in the balanced experiment (i.e 1:1 ratio). As the imbalance increased, there was a dip in the performance for the training sizes with lesser samples. In the heavily imbalanced ratios (1:25 and 1:50), the performance of the models increased as the sample size increased. This demonstrates that deep learning models are efficient in identifying the signals despite noise in the data when large noisy samples are available when compared to classical models.

This application is our proof of concept application for this study. In this application, we demonstrate a heuristic approach to create a silver standard dataset and train machine learning models in a weak supervision setting. We experimented with a binary classification setting and tested the models using a gold standard dataset. We evaluated the performance of the silver standard dataset in identifying the gold standard dataset using four different metrics and presented F-Measure and Recall metrics in the Results section. We observed an increase in the performance of the models with an increase in the sample size in both class balanced and imbalanced settings. We calculated theoretical bounds which indicate that when 14,430 clean samples (gold standard data) are available, we require 14,458 noisy samples for a best performing model and 42,021 noisy samples for the least performing model. Based on the theoretical bounds, we set up our experiments starting from 10,000 samples and systematically increased the training samples to 3 million. The results demonstrate the theoretical bounds to be accurate and also present an improvement in performance as the sample size increases. As discussed in chapter 3, it is relatively easier to obtain 42,022 samples of silver standard data than to obtain 14,430 samples of gold standard data.

9 APPLICATION 2: CHARACTERIZING DIFFERENT TYPES OF NATURAL DISASTERS: HURRICANES, EARTHQUAKES, FLOODS

Twitter has been extensively used as an active communication channel, especially during many crisis events, such as natural disasters like earthquakes, floods, typhoons, and hurricanes[159]. A wide range of information is tweeted during a disaster by people who are in need of help (e.g., food, shelter, medical assistance, etc.) or by people who are willing to donate or offer volunteering services or by the government to inform people of the latest updates[160,161]. Hence, it is essential to identify valuable information from the sea of information[162]. Several studies in the past have demonstrated the role of machine learning in analyzing natural disasters. Ofli et al.[163] utilized machine learning to make sense of aerial data during disasters. Resch et al.[164], utilized topic modeling and spatio-temporal analysis of social media data for disaster footprint and damage assessment. Several NLP techniques have been developed to detect and extract relevant information[165–167]. Nguyen et al. utilized convolutional neural networks to classify crisis related data on social networks. Madichetty and M, Sridevi[168] demonstrated that contextual representations improve supervised learning when using Twitter data for natural disasters. Several individual disasters were analyzed in the past and released the datasets to encourage reproducible research[55,162,169]. In this application, we first developed a heuristic to curate a silver standard dataset consisting of data from three different types of disasters i.e Hurricanes, Floods and Earthquakes. We then trained several machine learning models and compared the results to observe how efficient the silver standard trained models are in identifying ground truth labels.

9.1 Heuristic Curation

The heuristic is designed to identify signals that occur during a natural disaster. The objective here is to curate a heuristic independent to the type of specific natural disaster. In order to have a comprehensive list of signals, we generated n-grams (n=2 and n=3) from natural disaster datasets. To generate n-grams, we preprocessed the tweet text to remove emojis, emoticons, stopwords and lowercased the text. The generation of terms for each type of natural disaster is explained in the following sections. An initial analysis on the terms presented overlaps between bigrams and trigrams and hence we used only bigrams (n=2) as the heuristic.

9.1.1 Hurricanes

Hurricanes have been examined most frequently in supervised learning and NLP, particularly for text content analysis and multimedia content analysis[55]. For hurricanes, we hydrated the publicly available datasets for Hurricane Maria[102], Hurricane Sandy[103], Hurricane Irma and Hurricane Harvey[96] and obtained 51,500,116 tweets. We filtered the tweets and acquired only 9,789,940 clean English tweets. An initial analysis on the terms presented overlaps between bigrams and trigrams and hence we used only bigrams to attain the list of terms. Once we generated the bigrams, we sorted the terms in descending order of the counts and retained the top 150 terms. We removed terms in the format hurricane <Name> / <Name> hurricane / <hurricane name> term (Example: irma relief) and filtered 62 terms (“hurricane victims”, “power outages”, “heavy rain”) for hurricanes. Figure 16 displays the top 10 most frequent bigrams for hurricanes after filtering the hurricane names.

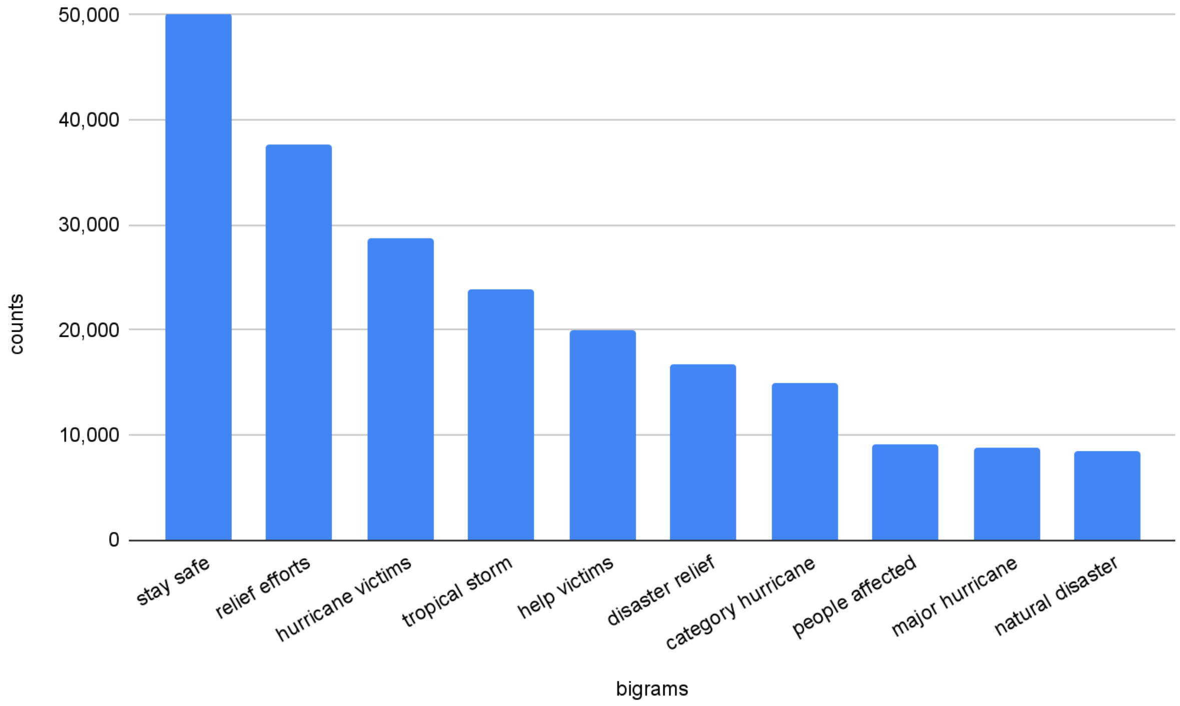


Figure 16. Top 10 most frequent bigrams for hurricanes

9.1.2 Earthquakes and Floods

We did not find any exclusive publicly available social media datasets for floods and earthquakes. Hence, we compiled the list of floods and earthquakes that occurred between 2018 and 2020 and extracted all relevant tweets from our longitudinal collection of Twitter data. We included 24 different floods and 4 different earthquakes to obtain relevant tweets from Regular Stream and generated the bigrams. We sorted the terms in descending order of the counts and retained the top 150 terms. We eliminated terms which contain the format <Country Name> floods, floods <Country Name>, <Country Name> earthquake. Post filtering, our final list of terms contain 58 unique flood terms and 48 unique earthquake terms. Figures 17 and 18 depict the top 10 most frequent bigrams for earthquakes and floods after filtering the country names.

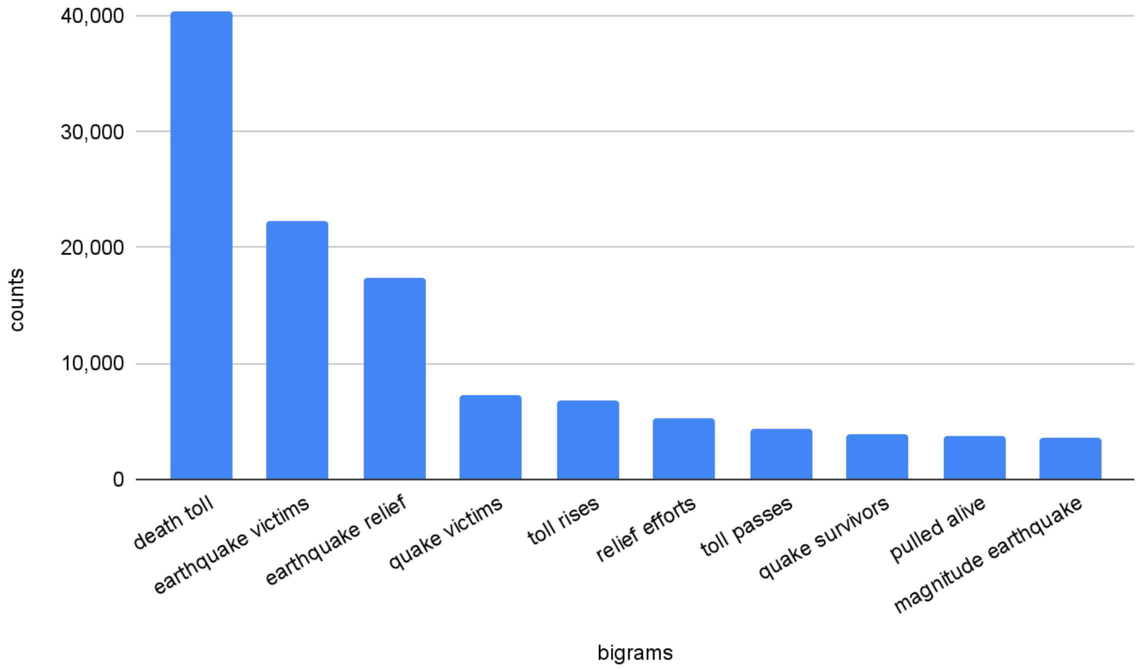


Figure 17. Top 10 most frequent bigrams for earthquakes

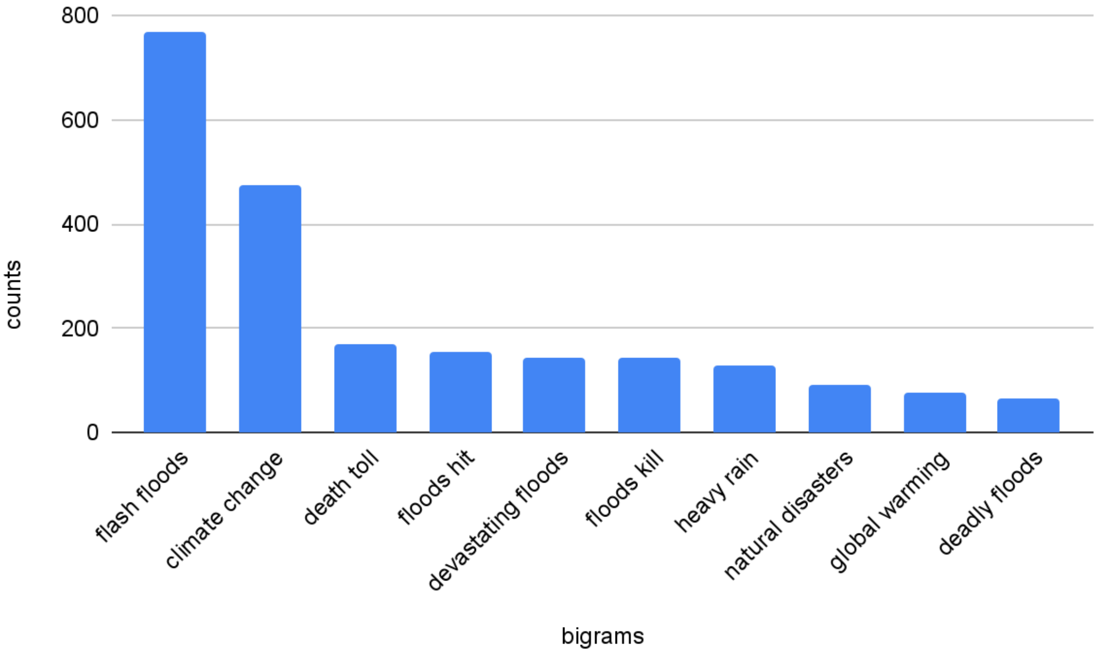


Figure 18. Top 10 most frequent bigrams for floods

The primary reason to eliminate the terms which are tied to a particular country (Eg: <country name> floods) or specific hurricane is to remove all the terms that can identify one specific event of a disaster. This enhances generalizability when using the heuristic for future natural disasters. Table 11 presents the details of the natural disasters utilized in this application and the events utilized for each natural disaster and the number of terms for each natural disaster. There is an overlap (Eg: “death toll”) in terms between the three different types of natural disasters. Since our objective is to identify the terms relevant to natural disasters, we filtered the terms from individual disasters and eliminated duplicate terms and created a comprehensive heuristic containing 155 unique bi-gram terms. As an additional filtering rule, we add an additional comprehensive list to the heuristic which contains a list of generic natural disaster terms i.e. [“hurricane”, “floods”, “earthquake” and “quake”]. When applying the heuristic for filtering, the bi-gram from the list of bi-grams and a term from the list of generic natural disasters must match to retrieve a tweet. Our final heuristic contains 155 bi-gram terms and a list of 7 generic natural disaster terms.

Table 11. No of terms for obtained for each natural disaster

Natural Disaster	**Events Included**	**Number of terms**
Hurricane	Maria, Sandy, Irma, Harvey	62
Floods	Rwanda, Kenya, Somalia, Burundi, Djibouti, Ethiopia, Uganda, Japan, Kerala, Vietnam, India, Indonesia, European, Spain, France, Italy, United Kingdom, Portugal, Maryland, Townsville, Venice, Thailand, Pakistan,Iran	58
Earthquakes	Indonesia, Albania, Fiji, Peru	48
Total unique terms		**155**

9.2 Generating the silver standard dataset

To create the silver standard dataset, we utilized the heuristic to filter tweets from Publicly available datasets and Regular Stream. We mined over 7 billion tweets from the two sources, and separated **977,353** clean English drug tweets using the heuristic. The data collection from each source is presented in the following sections.

9.2.1 Regular Stream Details

In this application, we used tweets collected between 2018 and 2021. Table 2 lists the details of tweets collected and filtered from the Twitter Stream. We used only clean English tweets for this stream. Table 12 lists the number of filtered tweets from the regular stream. The % tweets column represents the percentage of tweets filtered from the clean tweets. We filtered **38,260** natural disaster tweets from a total of 2,129,383,609 clean English tweets.

Table 12. Filtered Tweets from Regular Stream

Year	Filtered Tweets	Percentage of filtered tweets
2018	13,276	0.0029
2019	11,778	0.0021
2020	10,506	0.0014
2021 (Jan - May)	2,700	0.0008
Total	**38,260**	**0.0018**

9.2.2 Publicly Available Datasets

We filtered tweets from 34 different publicly available datasets using the heuristic. The publicly available datasets yielded more tweets than the regular stream because they are targeted datasets. 11 of the datasets are related to natural disasters (Eg: Hurricane Harvey, Dorian, climate change). Of all the datasets, hurricane Dorian dataset has the maximum percentage (19.23 %) of the clean tweets. Hurricane datasets retrieved only a small percentage of the total hurricane data. We believe that our heuristic eliminated the noise in the dataset since the datasets were collected based on keywords like “hurricane harvey, harvey”. The percentage of tweets in the following table is the percentage of filtered natural disaster tweets in clean tweets. Table 13 presents the number of filtered tweets from publicly available datasets.

Table 13. Filtered Tweets from publicly available datasets

Dataset	**Filtered tweets**	**Percentage of filtered tweets**
2016 presidential election[75]	4,428	0.01
Solar Eclipse[76]	23	0.00
hurricaneHarvey[96]	108,011	5.04
Hurricane Florence[97]	62,618	4.49
Hurricane Florence[101]	50,776	6.82
Hurricane Harvey[98]	35,540	4.02
Hurricane Irma[96]	53,378	2.28
Hurricane Maria[102]	2,276	1.41
Hurricane Sandy[103]	89,348	1.74
Hurricane Dorian[104]	80,068	19.23
Hurricane Dorian[105]	30,244	1.75
Election 2012[77]	8,772	0.04
Datarelease[78]	766	0.00
Beyond the Hashtag[79]	53	0.00
Climate Change[80]	93,494	1.16
Trump Tweet Ids[81]	294	0.00
Health Care[82]	2,645	0.01
2018 Congregational Election[106]	1,956	0.02
News Outlets[89]	99,508	0.11
Women's March[83]	3	0.00
US Govt Ids[84]	24,750	0.36
End of Term[85]	7,222	0.18

Nipsey Tweets[86]	3	0.00
Winter Olympics[87]	23	0.00
Dallas Shooting[88]	53	0.00
Charlottesville[90]	0	0.00
Twitter-Events-2012-2016[91]	315,954	0.89
115th U.S. Congress Tweet Ids[99]	2,037	0.13
Immigration Exec Order[92]	10	0.00
Irish news English tweets [93]	59,211	0.13
Black Lives Matter[94]	688	0.03
2020 Presidential Election[100]	7,449	0.01
Tweets to Donald Trump[95]	28,574	0.02
Total	**1,170,175**	**0.17**

The silver standard dataset contains tweets from three different types of natural disasters, i.e. hurricanes, earthquakes, and floods. To summarize, we created a heuristic by generating bigrams from existing natural disasters datasets. To the heuristic we added a list of generic natural disaster terms which aids in identifying relevant tweets. Our heuristic of 155 bi-grams (n=2) and 7 generic natural disasters terms could filter **977,353** natural disaster tweets which is termed as silver standard dataset[170]. The heuristic does not contain any of the labels from the gold standard dataset and we did not use any annotated dataset to create the heuristic. The following are a sample of tweets from the silver standard dataset.

1. "flood waters as deep as four feet close roads in many southern wisconsin counties"
2. "number of terengganu flood victims swells to 2,000"
3. "taiwan earthquake: buildings tilt on sides after at least four killed and scores missing amid rescue operation"
4. "death toll rises further, hundreds left homeless as hurricane irma devastates the caribbean"

9.3 Calculating Theoretical Bounds

To compute the theoretical bounds, we trained several machine learning models on the gold standard data and presented the theoretical bounds for a high and low performing model. We split the gold standard data into 75:25 for training and test and obtained the accuracy of the machine learning models. We use accuracy to calculate the theoretical bounds.

9.3.1 Calculating theoretical bounds for a high performing model

In this computation, we consider “RoBERTa” to be a model with high performance with an accuracy score of 98%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score(0.98), and clean samples(m = 5,692), the minimum number of noisy samples are calculated below

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 5,692/(1-2*(1-0.98)))**2

noisy samples = 6,177

We would require 6,177 noisy samples to achieve the performance similar to the performance of models trained on 5,692 clean samples for a high performing model.

9.3.2 Calculating theoretical bounds for a low performing model

In this computation, we consider “Decision Tree” to be a model with low performance with an accuracy score of 85%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score (0.85), and clean samples (m = 5,692), the minimum number of noisy samples are calculated below

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 5,692/(1-2*(1-0.85)))**2

noisy samples = 11,617

We would require 16,576 noisy samples to achieve the performance similar to the performance of models trained on 5,692 clean samples for a high performing model.

To summarize, the minimum number of noisy samples required for the best performing model (RoBERTa) is **6,177** and the minimum number of noisy samples required for the least performing model (Decision Tree) with accuracy score(0.85) is **11,617**.

9.4 Experimental Setup

We started with a class balanced ratio, i.e 1:1 of natural disaster:non-natural disaster samples and systematically increased the non-natural disaster samples ratio all the way to 50. For each training ratio, we started with 10,000 samples and incrementally increased the sample size all the way to 1,000,000. For each training size, we also experimented with 10 different seeds. For example, we have 10,000 positive labeled samples and 40,000 negative labeled samples in a training ratio of 1:5 with a sample size of 50,000. In total, we experimented with 5 different training ratios (1:1, 1:5, 1:15, 1:25, 1:50), 9 different sample sizes (10,000, 30,000, 50,000, 100,000, 200,000, 300,000, 500,000, 800,000, 1,000,000), 10 seeds for each training size, and 11 different machine learning models (SVM, NB, LR, RF, DT, CNN, LSTM , BERT,RoBERTa, BERTweet, DisasterBERT), totaling to **4,950 experiments**. We used the silver standard dataset and labeled all the samples in the silver standard dataset as positive samples. For CNN and LSTM models, we used the Glove embedding model that was trained on 840B tokens, 2.2M vocab, cased, and 300d vectors. We collected 1.5 million non-natural disaster tweets and labeled them as negative tweets. A non-natural disaster tweet is a tweet which does not match with any of our terms in the heuristic. We utilized a stratified ratio of 75-25 of the dataset as training and validation data. The validation data was utilized to either improve the performance of the model or to terminate the learning process when there is no significant improvement. To test our models, we utilized a publicly available gold standard dataset which is detailed in the next section.

9.4.1 Gold Standard Data

To evaluate the machine learning models, we used a publicly available gold standard dataset[160] that was released in 2016. The labeled dataset contained data labeled by paid workers[171] and volunteers for several natural disasters like hurricanes, earthquakes, floods, typhoons, and landslides. In this application, we utilized data labeled by paid workers to maintain uniform standards. We utilized the data for three different types of natural disasters, i.e hurricanes, floods and earthquakes. Only one of the 9 different labels were available for each tweet in the dataset. **Injured or dead people** indicate reports of casualties and/or injured people due to the crisis. **Missing, trapped, or found people** signify reports and or questions about missing or found people. **Displaced people and evacuations** denote information about people who have relocated due to the crisis, even for a short time (includes evacuations). **Infrastructure and utilities damage** imply reports of damaged buildings, roads, bridges, or utilities/services interrupted or restored. **Donation needs or offers or volunteering services** reveal reports of urgent needs or donations of shelter and/or supplies such as food, water, clothing, money, medical supplies or blood; and volunteering services. **Caution and advice** contain reports of warnings issued or lifted, guidance and tips. **Sympathy and emotional support** indicate prayers, thoughts, and emotional support. **Other useful information** indicates other useful information that helps understand the situation and **not related or irrelevant** indicate unrelated to the situation or irrelevant. We did not use tweets labeled with Donation needs or offers or volunteering services, Sympathy and emotional support and Other useful information in our gold standard dataset as they do not provide any strong signals describing a natural disaster. Tweets labeled as “Not related” are used as negative sets (label 0). We cleaned up the gold standard dataset by removing retweets and incomplete tweets. Post clean up, we found that the dataset was imbalanced, hence to balance the dataset, we added few negative tweets to the dataset. The tweets which do not match with any of the patterns in our heuristic were added as negative tweets. A total of **5,692** tweets are used in the gold standard dataset with 2,846 tweets labeled as positive (label 1) and 2,846 labeled as negative (label 0).

9.5 Results

To evaluate the performance of the models, we used four different metrics, Precision (P), Recall (R), F-Measure (F), and Accuracy (A). For each training size, we used 10 seeds which resulted in 10 experiments. Hence, in order to avoid bias and not show only the best results, we present the mean of 10 experiments in each training size. Figures 19-23 represent the performance of F-measure in classical models starting from sample size 10,000 to 1,000,000 samples for each ratio. However, for the imbalanced training ratios (Eg: 1:25 and 1:50), the recall metric is more valuable than the precision metric. Figures 24 and 25 present the progression of the recall metric for training ratios 1:25 and 1:50. Figures 26-30 present the F-measure performance of deep learning models for each ratio. Figures 31 and 32 present the recall metric for the ratios 1:25 and 1:50 training ratios for the deep learning models. All the additional results (all precision plots, balanced and lightly imbalance recall plots) for both the classical and deep learning models are enclosed in the Appendix section.

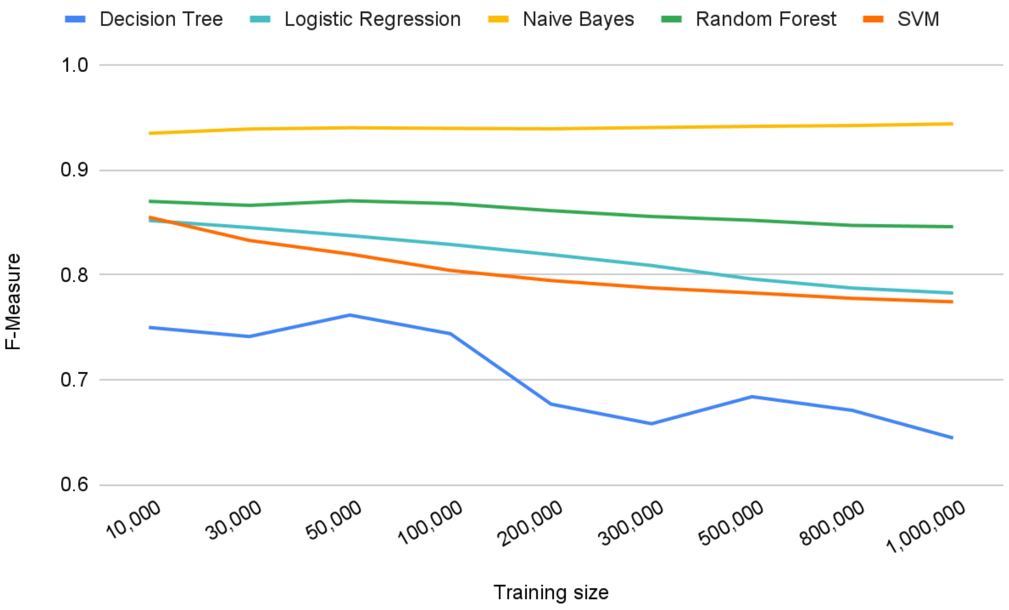


Figure 19. Classical models mean F-measure for 1:1 ratio

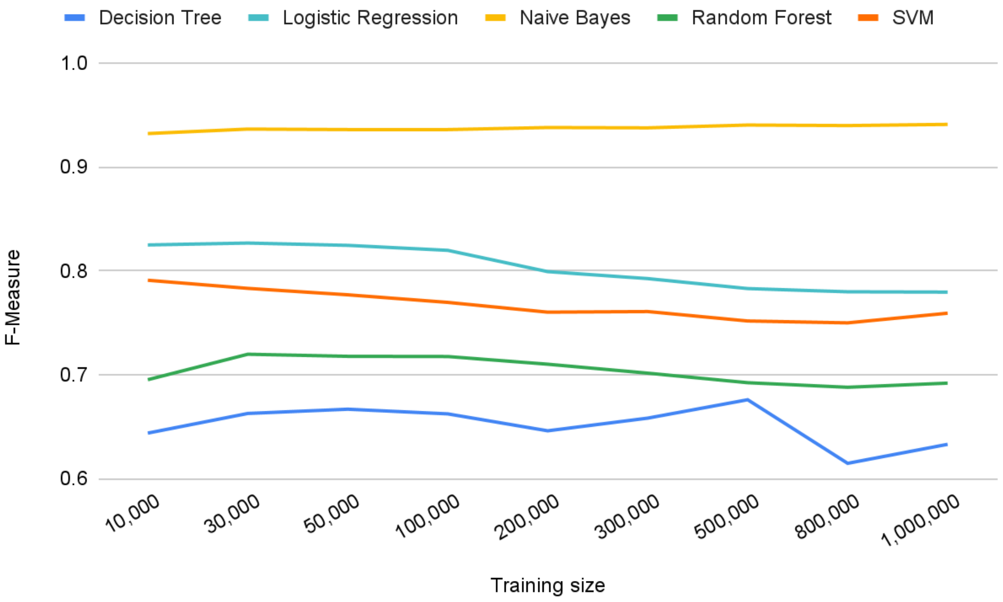


Figure 20. Classical models mean F-measure for 1:5 ratio

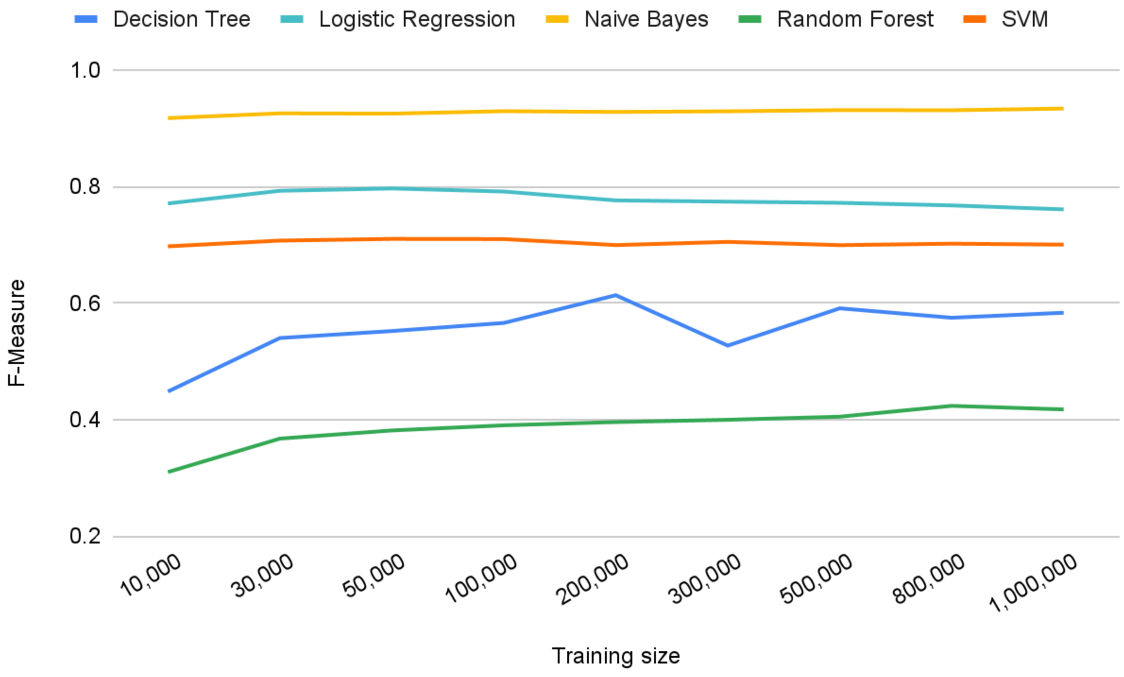


Figure 21. Classical models mean F-measure for 1:15 ratio

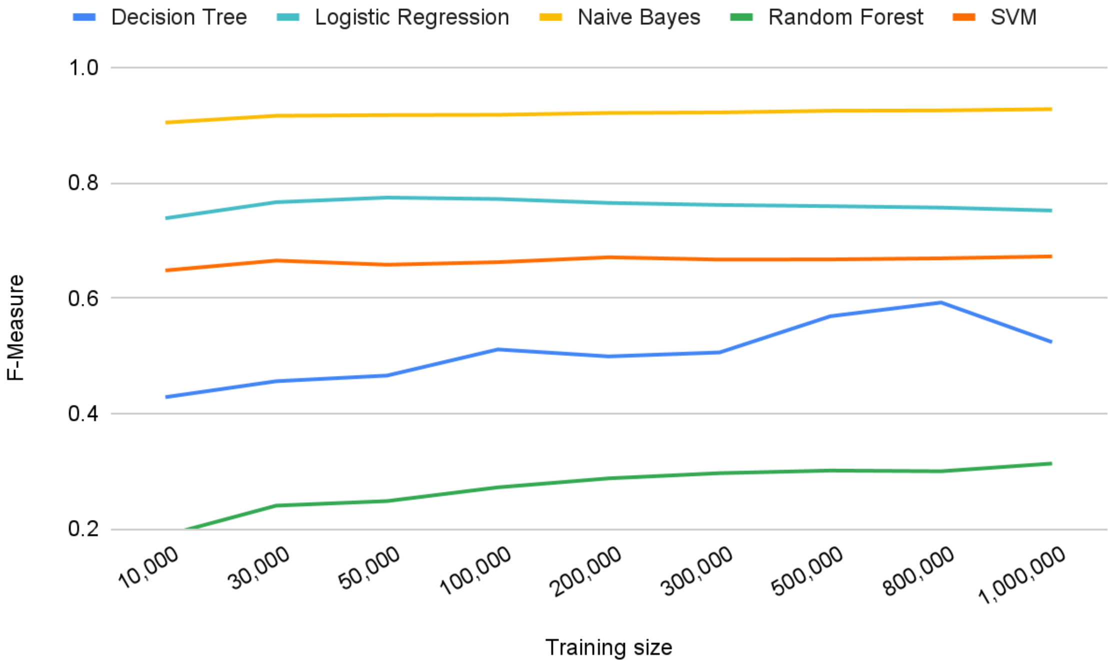


Figure 22. Classical models mean F-measure for 1:25 ratio

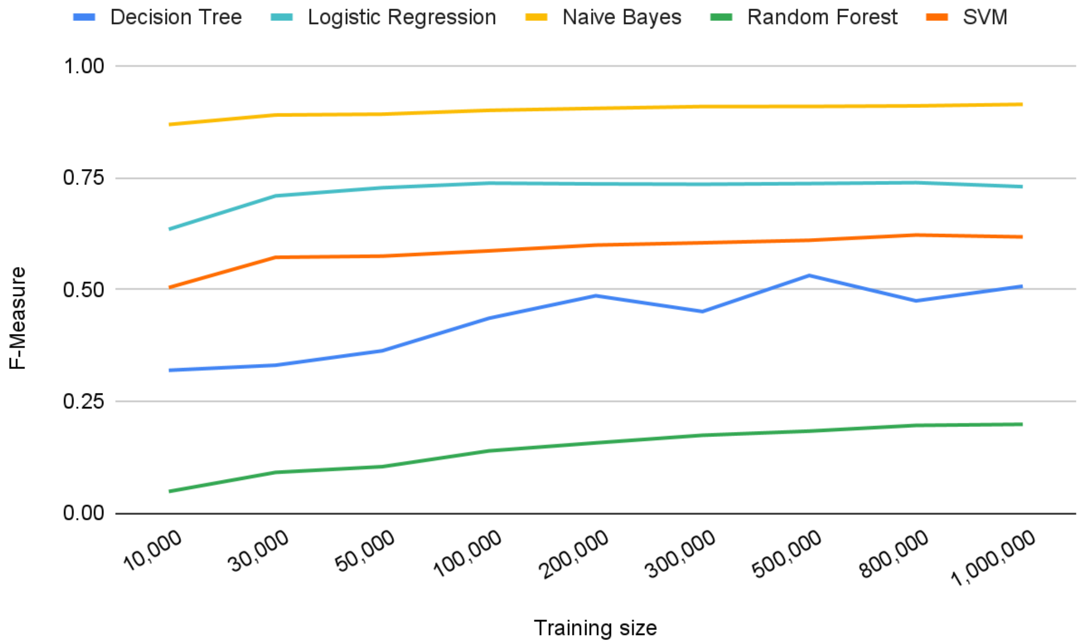


Figure 23. Classical models mean F-measure for 1:50 ratio

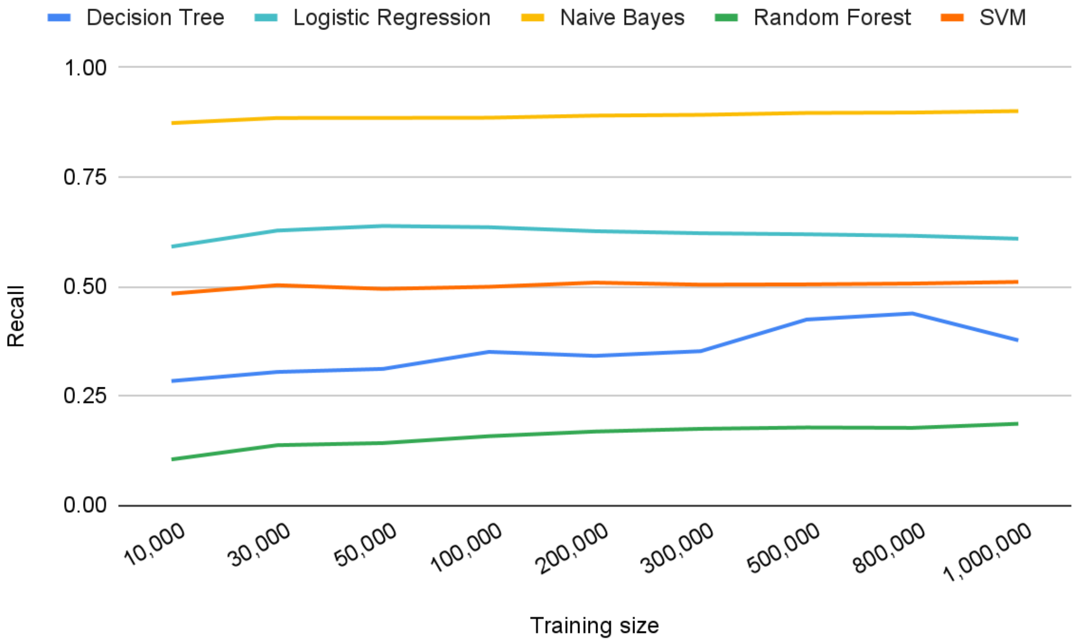


Figure 24. Classical models mean Recall for 1:25 ratio

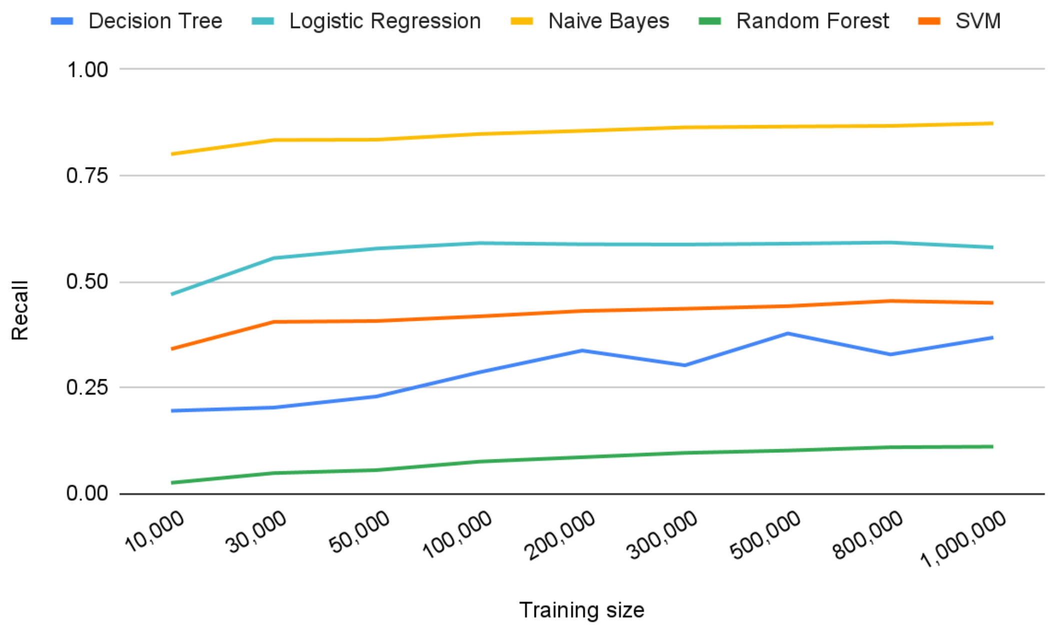


Figure 25. Classical models mean Recall for 1:50 ratio

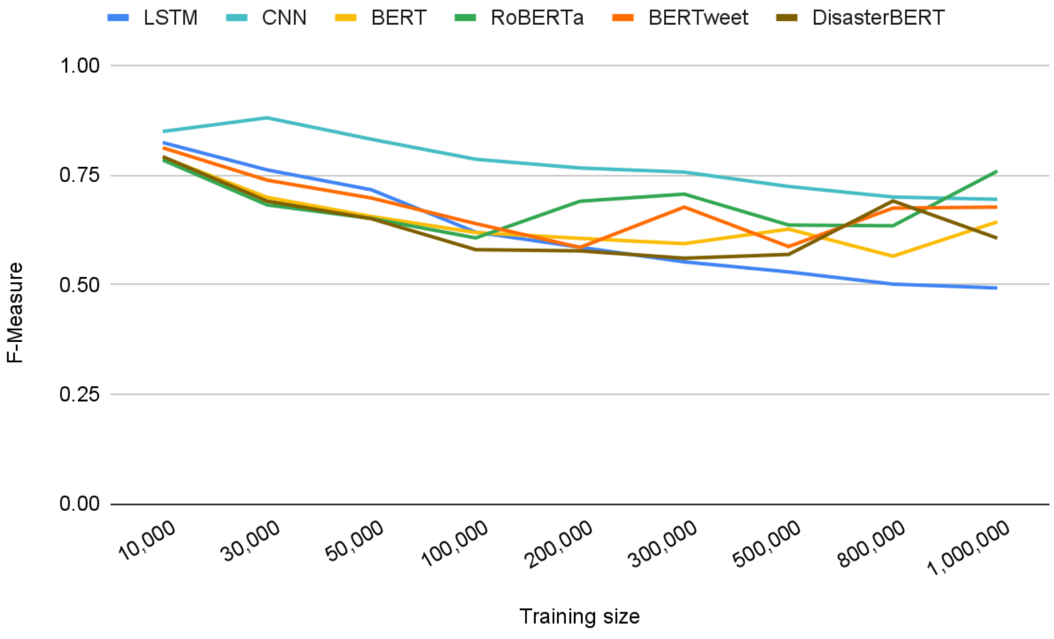


Figure 26. Deep learning models mean F-measure for 1:1 ratio

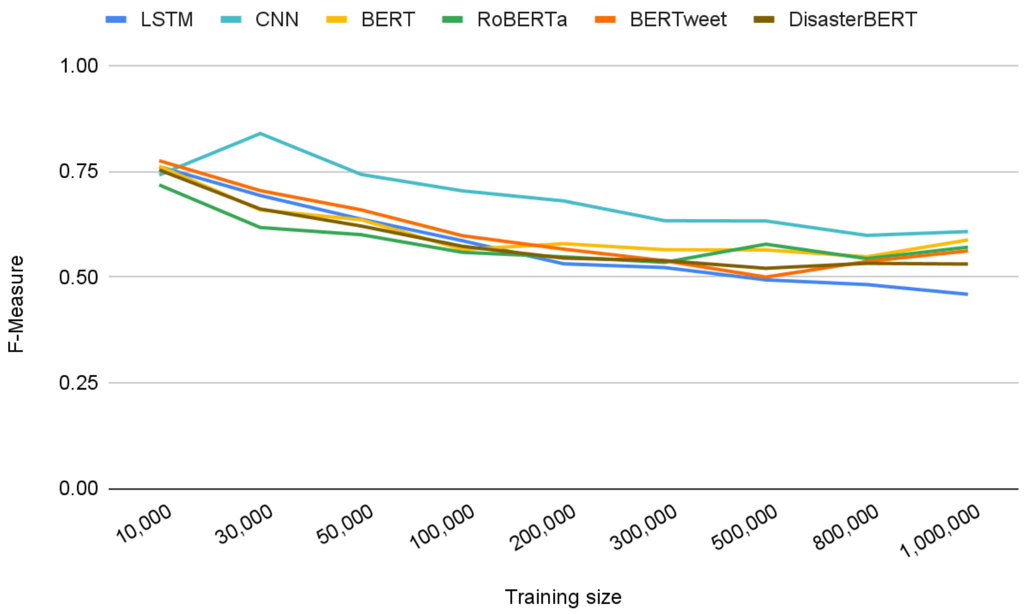


Figure 27. Deep learning models mean F-measure for 1:5 ratio

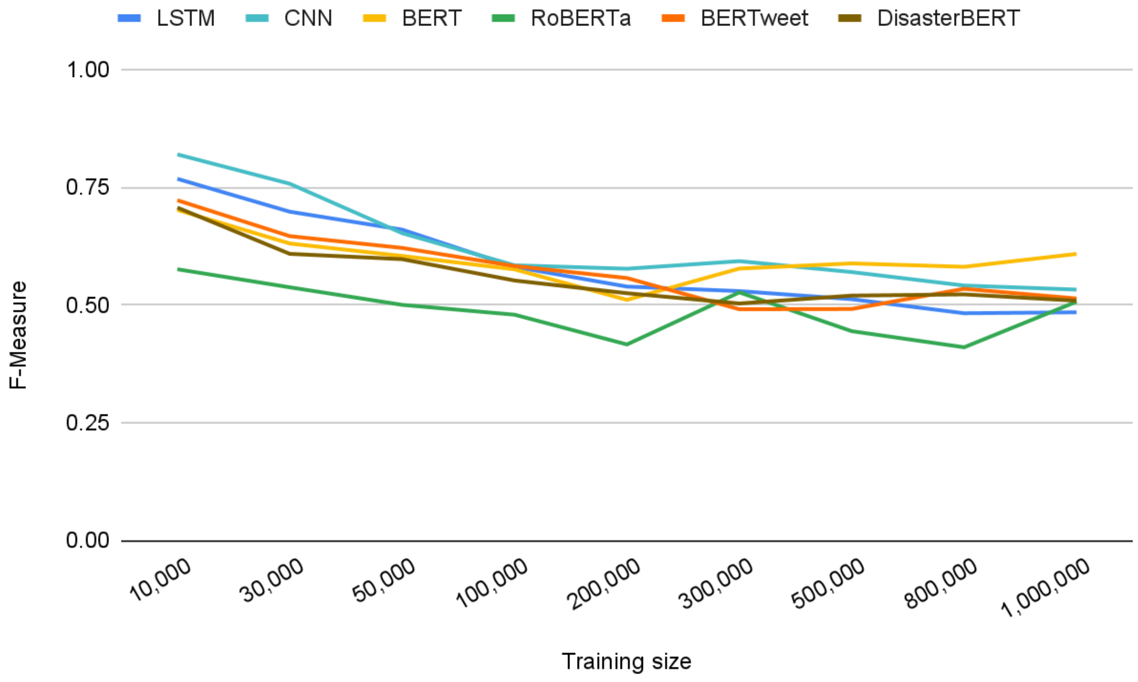


Figure 28. Deep learning models mean F-measure for 1:15 ratio

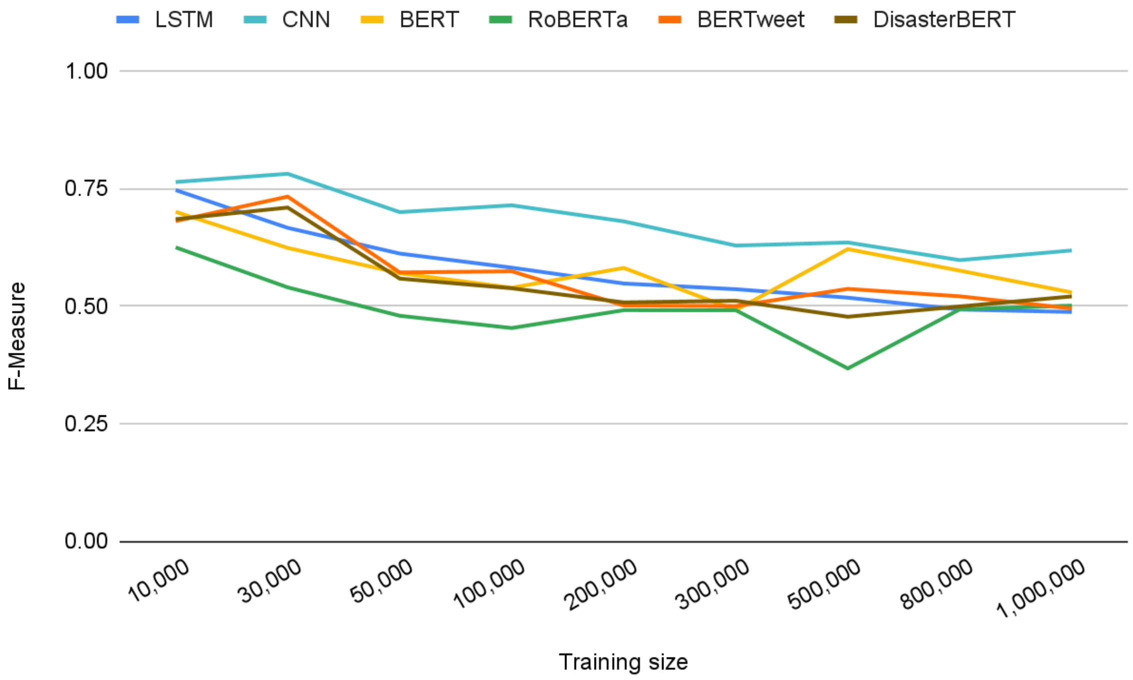


Figure 29. Deep learning models mean F-measure for 1:25 ratio

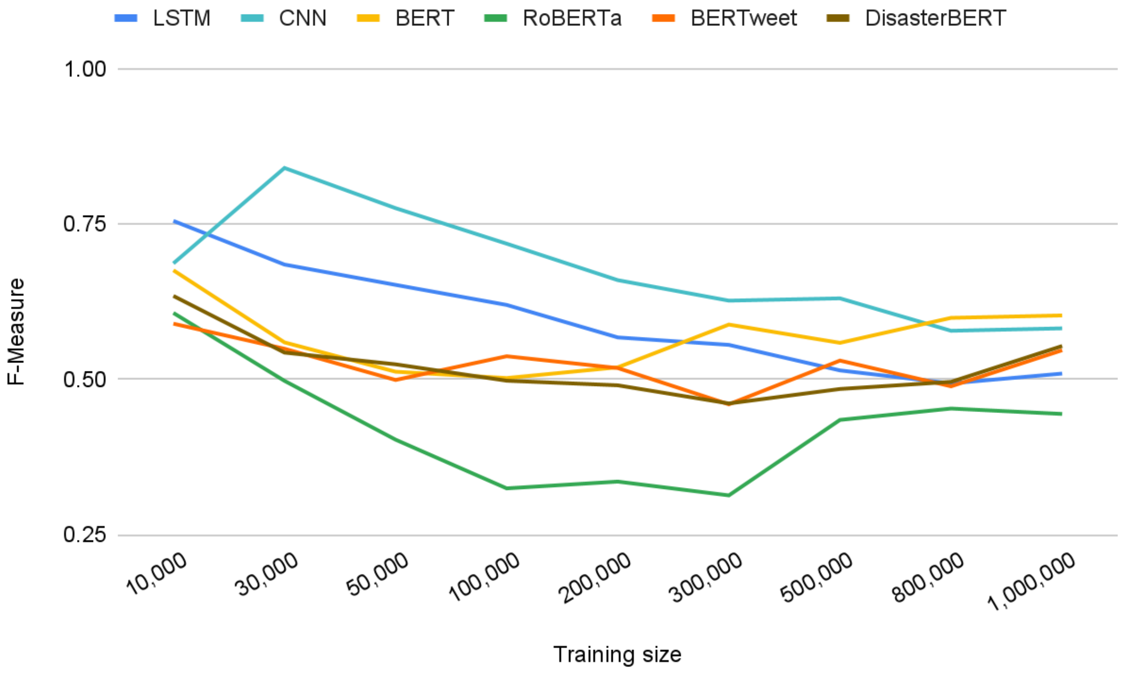


Figure 30. Deep learning models mean F-measure for 1:50 ratio

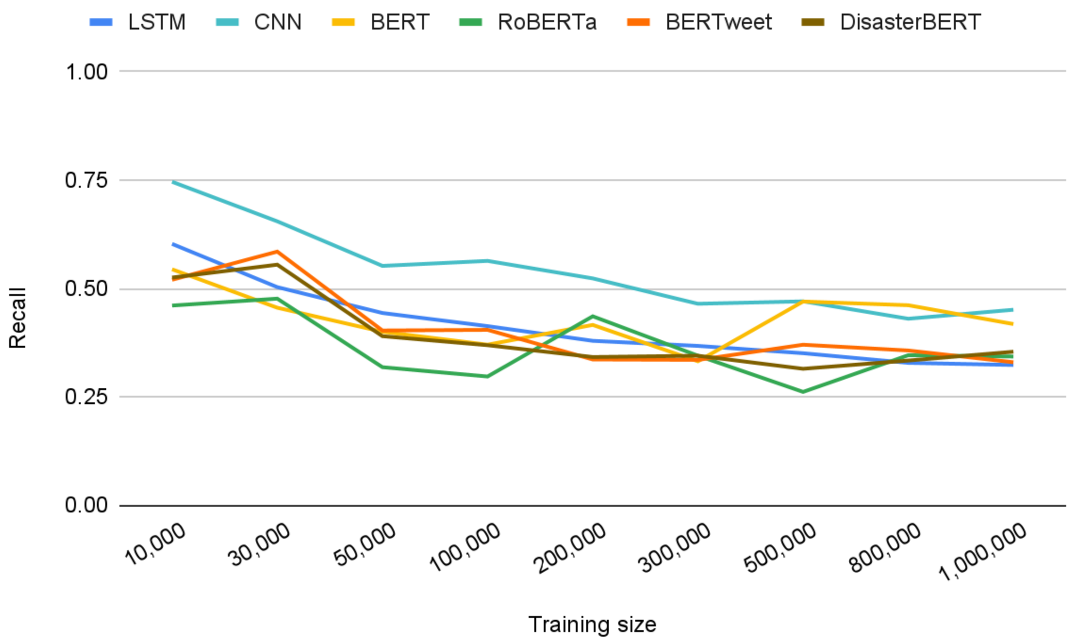


Figure 31. Deep learning models mean Recall for 1:25 ratio

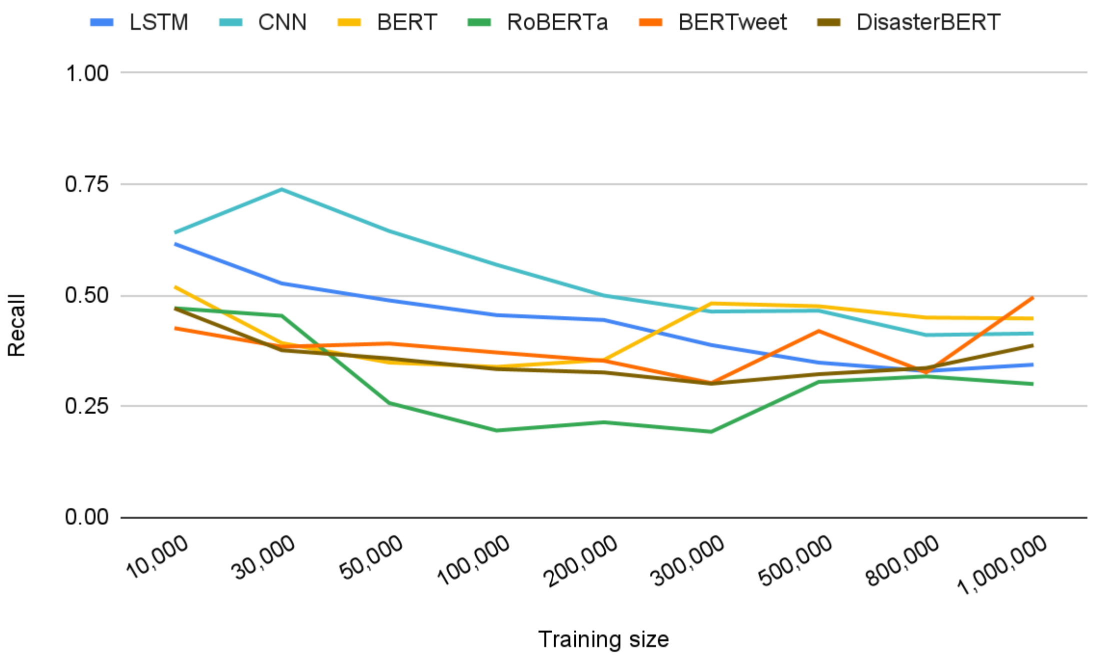


Figure 32. Deep learning models mean Recall for 1:50 ratio

In all our experiments, the classical models, especially the Naive Bayes model, had the best performance when compared to all the other models for each training size and training ratio. For all the training ratios, the Naive Bayes model achieved performance greater than 89% when trained with the noisy silver standard dataset. Logistic Regression and SVM also had performance greater than 75% for training ratios until 1:25. As the unbalanced ratio increases, there is a decline in performance. A decline in the performance of random forest and decision tree can also be observed as the unbalanced samples in the training data increase. Surprisingly, the deep learning models did not outperform the classical models. The CNN model consistently performed better than other models. The CNN model hits 88% F-measure on 1:1 training ratio and has a performance greater than 70% for most training ratios. As the training ratio increases, there is an increase in the negative samples in the training data. Hence, there is a decrease in the performance in imbalance classes. Further, we experience a decrease in the performance as the sample size increases for each training ratio for imbalance classes. We expected the transformer models to perform better, however, they were not the top performing models in this application. Noticeably, there is a decline in the performance of transformer models in the evenly balanced and lightly imbalanced ratios. We believe the increase of noisy labels to be the reason for a decreased performance.

To summarize, in this application we utilized a heuristic which contains bigrams generated from past natural disasters and a list of generic natural disaster terms to create the silver standard dataset. We experimented with both class balanced and imbalanced data and trained several machine learning models in a binary classification setting. Our results demonstrate the performance of silver standard data in identifying publicly available gold standard data. We calculate theoretical bounds which indicate that a minimum of 6,177 noisy samples were required for the least performing model while a minimum of 11,617 noisy samples were required for the best performing model.

We experimented with sample sizes starting from 10,000 which is within the limit of the theoretical bounds and present our results that demonstrate the accuracy of theoretical bounds.

10 APPLICATION 3: DETECTING EPIDEMIC TWEETS AND EVALUATION OF LARGE SCALE EPIDEMIC CORPUS

Social media is where people digitally converge during disasters and use it as a lifeline for communication during natural disasters, epidemics, war and other crises. In the past monitoring disease outbreaks using the Internet, typically involved either mining newspaper articles[172,173] or mining health related websites[148,174,175]. However, with an increase in the microblogging websites such as Twitter and Facebook, people often tend to utilize these platforms to communicate, which results in large amounts of valuable information. For example, Covid-19 is a recent epidemic in which Twitter was extensively used by users across the globe. There have been over 1.3 billion Covid-19 tweets retrieved from the 1% sample of the Twitter data over a period of 2 years[176]. This demonstrates that people tend to heavily rely on Twitter for communication during epidemics, and additionally displays that Twitter contains an abundance of data signals which can be used for research. Several studies in the past demonstrated successful results using NLP[177,178] and supervised learning techniques[179,180]. However, in recent times there has been a shift in relying towards other forms of machine learning techniques to avoid the manual curation process involved in supervised learning and weak supervision methods have not been utilized thus far for epidemic research. In this application, we created a heuristic using regular expressions to identify epidemic related tweets and collected over 7 billion tweets from Twitter between 2013 and 2021. We filtered 8 different types of epidemic tweets using a heuristic approach and curated a silver standard dataset. We trained several machine learning models using the silver standard dataset and validated the performance of the models using a large epidemic corpus[108] containing over 30 million epidemic tweets. To further validate the silver standard dataset, we used a gold standard dataset to determine the performance of models in identifying a gold standard dataset. To the best of our knowledge, we are the first to utilize weak supervision techniques for epidemics research.

10.1 Heuristic Curation

To create a heuristic, we first identified all the epidemics that occurred between 2006 and 2019. 2006 was our starting point since Twitter was established in 2006. We intentionally did not include Covid-19 as there are several large datasets on Covid-19[176,181,182] and very limited datasets on other epidemics. Several studies in the past were on identifying and analyzing influenza[177,179,183,184] and a few other individual epidemics like Dengue[185], Swine Flu[186–188], HIV[189,190]. However, none of these studies utilized a longitudinal dataset or have multiple epidemics in the dataset. In order to build a longitudinal and multi epidemic dataset, we identified 8 different deadly epidemics including Cholera, Ebola, H1N1, HIV, Influenza, MERS, SARS and Yellow Fever. In addition to the epidemics, we also identified virus variants for few epidemics (Eg: Swine flu is a virus variant of H1N1; AIDS is caused by HIV). We used regular expressions as our labeling heuristic since for epidemics like "cholera", we wanted to retrieve all the tweets irrespective of case. To summarize, for epidemics Cholera, Ebola, H1N1, Influenza, flu, HIV, MERS and SARS we used expressions which would filter tweets irrespective of case. Regular expressions have the advantage of enabling faster searches than a list of terms, especially when the text cannot be divided into tokens. Epidemic tweets usually have valuable information in hashtags and regular expressions decrease the search time in such cases. The regular expression used for filtering epidemic tweets is presented in the appendix.

10.2 Generating the silver standard dataset

To create the silver standard dataset, we filtered 8 different types of English epidemic tweets using the heuristic and filtered tweets from both publicly available datasets and Twitter regular stream. We removed duplicate tweets and preprocessed the tweet text by removing emojis, emoticons, URLs and striped white spaces.

10.2.1 Regular Stream Details

In this application, we used tweets collected between 2018 and 2021. Table 14 lists the details of tweets collected and filtered from the Twitter Stream. We used only clean English tweets from this stream. The % tweets column represents the percentage of tweets filtered from the clean tweets. There is an increase in the count of relevant tweets due to Covid-19. While we did not use Covid-19 in our heuristic, several people on Twitter compared similarities between Covid-19 and flu, as Covid-19 also causes respiratory illness. We filtered a total of 325,125 tweets from 2,129,383,609 clean tweets using the heuristic on the regular stream.

Table 14. Filtered Tweets from Regular Stream

Year	**Filtered tweets**	**Percentage of filtered tweets**
2018	51,647	0.01
2019	50,647	0.01
2020	183,901	0.02
2021 (Jan - May)	68,762	0.01
Total	**325,125**	**0.02**

10.2.2 Publicly Available Datasets

We filtered tweets from 34 different publicly available datasets using the heuristic. The publicly available datasets yielded more tweets than the regular stream, since the datasets contain tweets that have been collected since 2013. Only 2 datasets are related to Epidemics (Health Care and ATAM dataset). While the other datasets are not relevant to Epidemics, we could obtain a significant number of tweets from the publicly available datasets. This demonstrates the availability of epidemic tweets in non-epidemic datasets. A total of **2,095,057** tweets were filtered using the heuristic from a total of 3,050,058,283 tweets. Table 15 presents the details of the total number of filtered tweets for this application.

Table 15. Filtered tweets from publicly available datasets

Dataset	filtered tweets	Percentage of filtered tweets
2016 presidential election[75]	16,657	0.03
Solar Eclipse[76]	241	0.02
hurricaneHarvey[96]	324	0.02
Hurricane Florence[97]	180	0.01
Hurricane Florence[101]	102	0.01
Hurricane Harvey[98]	207	0.02
Hurricane Irma[96]	168	0.01
Hurricane Maria[102]	85	0.05
Hurricane Sandy[103]	1,397	0.03
Hurricane Dorian[104]	254	0.06
Hurricane Dorian[105]	42	0.00
Election 2012[77]	3,483	0.02
Datarelease[78]	10,671	0.03
Beyond the Hashtag[79]	8,220	0.11
Climate Change[80]	3,395	0.04
Trump Tweet Ids[81]	1,782	0.02
Health Care[82]	41,411	0.18
2018 Congregational Election[106]	3,357	0.03
News Outlets[89]	107,736	0.12
Women's March[83]	154	0.01
US Govt Ids[84]	53,734	0.77
End of Term[85]	38,804	0.94
Nipsey Tweets[86]	1,385	0.11
Winter Olympics[87]	241	0.02
Dallas Shooting[88]	106	0.01
Charlottesville[90]	23	0.01
Twitter-Events-2012-2016[91]	315,301	0.89
115th U.S. Congress Tweet Ids[99]	3,095	0.20
Immigration Exec Order[92]	359	0.02
Irish news English tweets[93]	62,180	0.14

Black Lives Matter[94]	783	0.03
2020 Presidential Election[100]	379,256	0.26
Tweets to Donald Trump[95]	169,410	0.10
ATAM dataset[107]	870,514	1.16
Total	**2,095,057**	**0.27**

Combining our data filtered from the Twitter stream and hydrated datasets, we obtained 2,420,182 tweets and pre-processed the filtered tweets. We also lowercased the tweet text for data standardization. After removing duplicate tweets, the silver standard dataset contains 2,302,924 tweets which belong to 9 different epidemics[191]. Table 16 lists the number of tweets in the silver standard dataset for each epidemic.

Table 16. Counts of Epidemic Tweets in Silver Standard Dataset

Epidemic	**Counts**
Cholera	18,375
Ebola	441,035
Flu	1,340,557
H1N1	100,146
HIV/AIDS	200,291
Influenza	41,060
MERS	8,993
SARS	66,980
Swine Flu	76,784
Yellow Fever	8,703
Total	**2,302,924**

Unsurprisingly flu has the most number of tweets (58.2%) of the epidemic tweets, since it is more prevalent than other epidemics. 19.2% of the filtered tweets are from the epidemic Ebola. We intentionally separated flu and influenza tweets, since flu is more prevalent in Internet language than influenza. The following are a sample of tweets from the silver standard dataset.

1. “so sick headache, fever, chills, nausea... guess the **flu** finally got me. nyquil and bed.”

2. “59 persons came in contact with **ebola** victim -lasg via @360nobs”
3. “the horrifying spread of **cholera** epidemic has claimed the lives of 2906 people in yemen #yemenforgottenwar”
4. “london man may be cured of **hiv** after stem-cell transplant, researchers say”
5. “having an allergic reacting to the yellow fever vaccine... #disgusting”

10.3 Calculating Theoretical Bounds

To compute the theoretical bounds, we trained several machine learning models on the gold standard data and presented the theoretical bounds for a high and low performing model. We split the gold standard data into 75:25 for training and test and obtained the accuracy of the machine learning models. We use accuracy to calculate the theoretical bounds.

10.3.1 Calculating theoretical bounds for a high performing model

In this computation, we consider “RoBERTa” to be a model with high performance with an accuracy score of 99%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score(0.99), and clean samples (m = 4,590), the minimum number of noisy samples are calculated below

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 4,590/(1-2*(1-0.99)))**2

noisy samples = **4,780**

We would require 4,780 noisy samples to achieve the performance similar to the performance of models trained on 4,590 clean samples for a high performing model.

10.3.2 Calculating theoretical bounds for a low performing model

In this computation, we consider “Decision Tree” to be a model with low performance with an accuracy score of 79%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score (0.7930), and clean samples (m = 4,590), the minimum number of noisy samples are below noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 4,590/(1-2*(1-0.7930)))**2

noisy samples = 13,367

We would require 13,367 noisy samples to achieve the performance similar to the performance of models trained on 4,590 clean samples for a high performing model.

To summarize, the minimum number of noisy samples required for the best performing model (RoBERTa) is **4,780** and the minimum number of noisy samples required for the least performing model (Decision Tree) is **13,367**.

10.4 Evaluation on a large scale corpus

In the previous applications, we utilized a relatively smaller gold standard dataset to test the weak supervision approach. In this application, we set to evaluate the silver standard data using a large epidemic corpus. We identified only 1 large multi class epidemic corpus and this further highlights the need to have a publicly available large epidemic corpus open for scientific research which our silver standard intends to achieve. The EPIC corpus[108] contains 30 million tweets from 4 epidemics (Cholera, Ebola, MERS and Swine Flu) in several languages collected between 2009 and 2020. We could hydrate only 27,903,463 tweets from the Epic corpus as tweets were not available since they were either removed or deleted. Out of the 27,903,463 hydrated tweets only 18,367,000 were English language tweets. Since we built our silver standard dataset using a language filter set to English, we only utilized English language tweets from EPIC corpus. In the English tweets, 4,548,519 tweets were Swine Flu tweets, 1,020,094 tweets were Cholera tweets, 11,092,583 tweets were Ebola tweets and 133,011 were MERS tweets. This is the largest publicly available non-covid19 epidemic corpus. This corpus was purely collected based on keywords from Twitter streams. Table 17 depicts the distribution of different kinds of epidemic tweets on the EPIC corpus.

The majority of tweets belong to Ebola and Swine Flu epidemics while Cholera and MERS were in minority.

Table 17. Distribution of epidemics in EPIC corpus

Epidemic	**Total Tweets**	**English Tweets**
Swine Flu	5,965,868	4,586,012
Cholera	2,180,427	1,032,900
Ebola	19,516,570	11,414,459
MERS	240,598	134,306
Total	**27,903,463**	**17,167,677**

10.4.1 Experimental Setup

To test the weak supervision approach, we trained several machine learning models using the silver standard. We used the filtered tweets from the **Cholera, Ebola, MERS and Swine Flu** and labeled each class separately. We collected an equal number of non-epidemic tweets i.e. tweets that do not contain any of the epidemics in the tweet text and labeled them as non-epidemic samples. We utilized a stratified ratio of 75-25 of the dataset as training and validation data. The validation data was utilized to either improve the performance of the models or to incorporate early stopping techniques. To test the models, we utilized the EPIC corpus English tweets as a test set. Since the silver standard dataset contains less number of tweets than EPIC Corpus we removed tweets from EPIC corpus which were already available in silver standard dataset. We evaluated this corpus using a multi class classification instead of a binary setting.

We experimented with three classical models including SVM, Decision Tree, and Logistic Regression models using the Scikit learn[112] python library and 2 different Transformer models which include BERT[137], and BERTweet[145]. For the classical models, the TF-IDF vectorizer was used to convert raw tweet text to TF-IDF features and return the document-term matrix, which is sent to the model for training. We utilized LinearSVC for the SVM model and used the default parameters SVM. For the logistic regression model we set “max_iter to 1,000” and for the decision tree we set max_features to 'auto', criterion to "entropy" and max_depth to 150. We performed three different types of experiments for evaluating the silver standard corpus. The description for each experiment is outlined below

Experiment 1: A balanced corpus of silver standard dataset matched to the minimum number of samples available for all epidemics used as training data, i.e. each class contains 8,993 samples in the training data. For the test set, we balanced the test set using the minimum number of samples available in the EPIC corpus. In this scenario, each class in the test set contains 133,011 samples.

Experiment 2: This is a completely unbalanced experiment where we utilized all the samples available from silver standard dataset as training data and all the samples available in the EPIC corpus as test data.

10.4.2 Results of Multi-classification

To evaluate the performance of the models, we used four different metrics, Precision (P), Recall (R), F-Measure (F) for each class and also calculated Accuracy (A). Table 18,19,20 presents the results of F-Measure, Precision and Recall for all the 5 machine learning models.

Table 18. F-Measure of machine learning models

	Experiment 1					**Experiment 2**				
class	**Classical Models**			**Deep Learning**		**Classical Models**			**Deep Learning**	
	DT	LR	SVM	B	BT	DT	LR	SVM	B	BT
Cholera	0.6817	0.9866	0.9880	**0.9925**	0.9625	0.1826	0.9816	0.9836	**0.9834**	0.9822
Ebola	0.8545	0.9818	0.9803	**0.9922**	0.9860	0.474	0.9736	0.9797	**0.9925**	0.9924

MERS	0.9722	0.9866	0.9939	0.9947	**0.9961**	0.2673	0.9346	0.9337	0.8947	**0.874**
Swine Flu	0.7639	0.8644	0.8658	**0.8788**	0.8508	0.1843	0.8441	0.8598	**0.8231**	0.8163
non epidemic	0.7209	0.8889	0.8913	**0.8967**	0.8956	0.7244	0.9723	0.9726	**0.9592**	0.9576
weighted avg	0.7987	0.9417	0.9439	**0.9510**	0.9382	0.5503	0.9555	0.9599	0.9455	**0.9504**

Table 19. Precision of machine learning models

	Experiment 1					**Experiment 2**				
class	**Classical Models**			**Deep Learning**		**Classical Models**			**Deep Learning**	
	DT	**LR**	**SVM**	**B**	**BT**	**DT**	**LR**	**SVM**	**B**	**BT**
Cholera	0.6963	0.9950	0.9969	**0.9975**	0.9418	0.5372	0.9914	0.989	**0.9959**	0.9810
Ebola	0.8362	0.9808	0.9769	**0.9952**	0.9844	0.6249	0.9617	0.9727	0.994	**0.9974**
MERS	0.9745	0.9849	0.9926	0.9924	**0.9948**	0.3924	0.9016	0.8852	**0.8118**	0.7786
Swine Flu	0.9157	0.9973	0.9979	0.9982	**0.9985**	0.7171	0.9977	0.9977	**0.9997**	0.9962
non epidemic	0.6409	0.8011	0.8045	0.8143	**0.8188**	0.5974	0.9464	0.9469	**0.9224**	0.9193
weighted avg	0.8127	0.9518	0.9538	**0.9595**	0.9477	0.6201	0.9596	0.9634	0.9583	**0.9568**

Table 20. Recall of machine learning model

	Experiment 1					**Experiment 2**				
class	**Classical Models**			**Deep Learning**		**Classical Models**			**Deep Learning**	
	DT	LR	SVM	B	BT	DT	LR	SVM	B	BT
Cholera	0.6678	0.9783	0.9793	**0.9875**	0.9842	0.5372	0.9914	0.989	0.9959	**0.9810**
Ebola	0.8737	0.9829	0.9836	**0.9893**	0.9877	0.6249	0.9617	0.9727	0.994	**0.9974**
MERS	0.9699	0.9882	0.9953	0.9970	**0.9974**	0.3924	0.9016	0.8852	**0.8118**	0.7786
Swine Flu	0.6553	0.7627	0.7646	**0.7849**	0.7411	0.7171	0.9977	0.9977	**0.9997**	0.9962
non epidemic	0.8237	0.9984	0.9991	**0.9976**	0.9882	0.5974	0.9464	0.9469	**0.9224**	0.9193

weighted avg	0.6046	0.9421	0.9444	**0.9513**	0.9397	0.6201	0.9579	0.9619	**0.9549**	0.9532

Table 18-20 represents the performance of the silver standard dataset in identifying a large epidemic corpus. Except for the decision tree model, all models had a satisfactory performance (F-measure > 85%) for five different classes. Additionally, we calculated weighted F-measure for each model, and all the models performed at a level of 94%, compared to a score of 55% for the decision tree. This indicates the performance of silver standard data in identifying different classes in a large scale corpus. Since this is a multi-classification experiment, we plotted confusion matrices to determine how accurately the classes were predicted. Figures 33,34 depict the confusion matrices for the best models of each experiment. From Figure 34, we can observe that in the extremely imbalanced experiment, swine flu class tweets were incorrectly predicted as other classes.

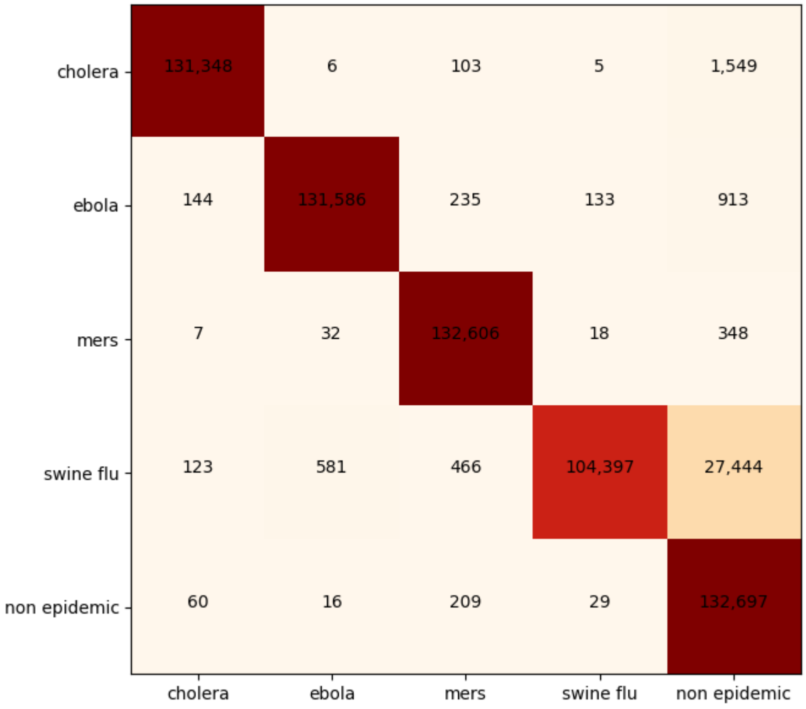


Figure 33. Confusion Matrix for BERT model for experiment 1

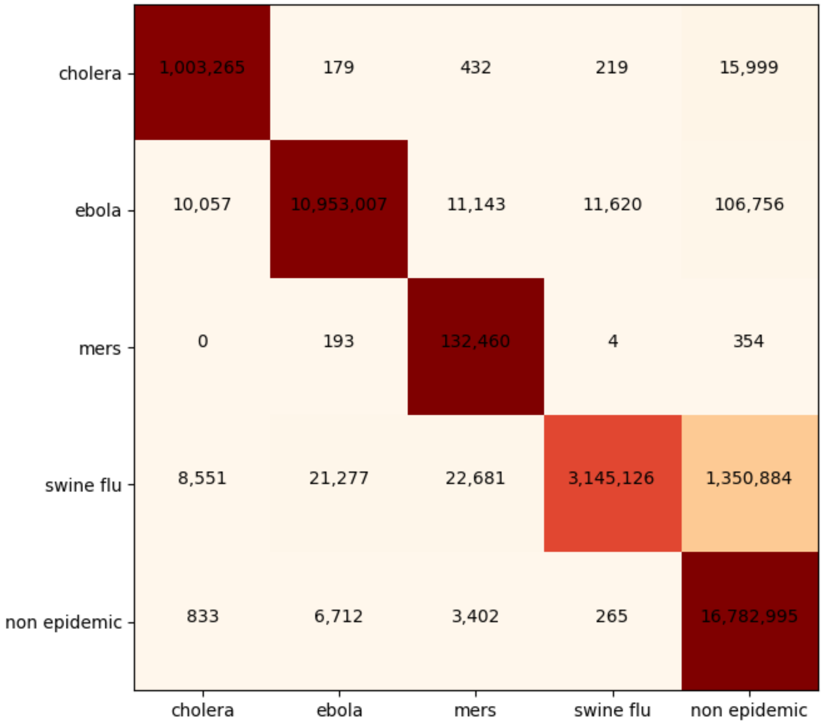


Figure 34. Confusion Matrix for BERTweet model for experiment 2

The primary objective of this experiment is to demonstrate the evaluation of a large epidemic corpus using the silver standard dataset. However, the large corpus is not gold standard and is noisy as it was curated using keyword based search on Twitter. In the next section, we demonstrate additional validation of the silver standard dataset.

10.5 Additional experiments to validate silver standard dataset

It is extremely difficult to obtain gold standard data for all the classes in the silver standard dataset, as the process of curation is tedious and laborious. Hence, we sought to identify publicly available gold standard datasets and discovered an "Influenza" gold standard dataset[192]. The data was annotated by using the Amazon Mechanical Turk service, and the annotated data was made available by Mark Dredze's group from Johns Hopkins University. Two different sets of labeled tweets were released - a) Self vs others indicating whether the condition is self-reported or not b) Awareness vs Infection tweets to indicate whether the tweet is about infection or awareness. However, since we are interested in influenza tweets, all the tweets were considered as "related" tweets. To perform a binary classification, we required a negative class. Hence, we added an equal number of non-influenza/ flu tweets also termed as "not related". A tweet is considered to be not related if the tweet text does not contain the term flu or influenza. Out of 15,131 tweets, we could hydrate only 8,731 tweets out of which only 4,816 were unique tweets. We filtered relevant tweets from the unique tweets and added an equal number of not related tweets to the gold standard. The final gold standard data contains 2,295 related (label 1) tweets and 2,295 not related tweets (label 0). The gold standard data was utilized as a test set for testing the models.

10.5.1 Experimental Setup

To examine the performance of silver standard data, like in previous applications, we experimented with several training sizes and ratios which include both class balanced and unbalanced data. We started with a class balanced ratio, i.e. 1:1 of relevant flu samples: not relevant flu samples and systematically increased the not relevant flu samples ratio all the way to 50. For each training ratio, we started with 10,000 samples and incrementally increased the sample size all the way to 1,000,000. For each training size we experimented with 10 different seeds. For example, we have 10,000 positive labeled samples and 40,000 negative labeled samples in a training ratio of 1:5 with a sample size of 50,000. In total, we experimented with 5 different training ratios (1:1, 1:5, 1:15, 1:25, 1:50), 9 different sample sizes (10,000, 30,000, 50,000, 100,000, 200,000, 300,000, 500,000, 800,000, 1,000,000), 10 seeds for each training size, and 10 different machine learning models (SVM, NB, LR, RF, DT, CNN, LSTM , BERT,RoBERTa, BERTweet) which totals to **4,500 experiments**. We used the "flu" samples from the silver standard dataset and labeled all the samples in the silver standard dataset as positive samples. For CNN and LSTM models we used the Glove embedding model which was trained on 840B tokens, 2.2M vocab, cased, 300d vectors. We collected 1.5 million non-flu tweets and labeled them as negative tweets. A non-flu tweet is a tweet which does not match with any of our terms in the heuristic. We utilized a stratified ratio of 75-25 of the dataset as training and validation data. To summarize, we used the flu class tweets from our silver standard dataset as positive samples and additionally added negative samples to the training data. The publicly available gold standard data was utilized to test the machine learning models.

10.6 Results

Similar to previous applications, we used the same metrics (Precision, Recall, F-Measure and Accuracy) to evaluate the machine learning models. Since we used 10 seeds for each training size, which resulted in 10 experiments, we calculated the mean of the 10 experiments per training size to avoid bias and presented the results. Table 21 presents the mean F-Measure for all the models for 1:1 and 1:50 ratios, which are class balanced and extremely imbalanced. We present the progression of the F-measure metric for classical models in Figures 35-39 and deep learning models in Figures 42-46 for all training ratios. Since recall is an important metric for highly imbalanced ratios, we present the progression of recall metric in classical models in Figures 40 and 41 and deep learning models in Figure 47 and 48 for 1:25 and 1:50 ratio. The other results are added to the appendix. The 'k' in Table 21 represents samples in thousands (Eg: 10k is 10,000 samples) and 'M' represents samples in millions.

Evidently, the classical models performed as good as the deep learning models in the class balanced ratio. The Decision Tree model had an inconsistent performance when compared to all other models. In classical models SVM, Logistic Regression and Naive Bayes performed equally well as the deep learning models. In the extremely imbalanced ratio i.e 1:50 the deep learning models outperform the classical models. Naive Bayes and Decision Tree classifiers could not fit the imbalanced data, while all the deep learning models consistently performed well in 1:50 ratio. In fact, the deep learning models' results are comparable to 1:1 ratio.

Table 21. Mean F-Measure of all the models for 1:1 and 1:50 ratio

Ratio	**Size**	**Classical Models**					**Deep Learning Models**				
		SVM	**LR**	**NB**	**DT**	**RF**	**B**	**RB**	**BT**	**CNN**	**LSTM**
1:1	10k	0.9989	0.9963	0.8097	0.8779	0.997	0.9994	0.9995	0.9991	0.8957	0.9912
	30k	0.9994	0.9986	0.8314	0.8765	0.9981	0.9996	0.9997	0.9998	0.9324	0.9974
	50k	0.9996	0.999	0.8542	0.8568	0.9982	0.9996	0.9998	0.9997	0.9426	0.9983
	100k	0.9996	0.9992	0.8535	0.7817	0.9977	0.9993	0.9998	0.9997	0.9532	0.9989
	200k	0.9998	0.9993	0.8652	0.8588	0.9983	0.9993	0.9998	0.9997	0.9568	0.9993
	300k	0.9998	0.9995	0.8699	0.7843	0.9981	0.9992	0.9998	0.9998	0.961	0.9994
	500k	0.9998	0.9996	0.8766	0.6902	0.9982	0.9657	0.9996	0.9997	0.9616	0.9998
	800k	0.9998	0.9997	0.8883	0.6229	0.9979	0.9988	0.9997	0.9997	0.9637	0.9998

	1M	0.9998	0.9997	0.8913	0.6112	0.9982	0.9969	0.9996	0.9997	0.964	0.9998
1:50	10k	0.9994	0.6752	0.0008	0.5931	0.8203	0.9967	0.9997	0.9969	0.8613	0.9967
	30k	0.9996	0.9246	0.001	0.7245	0.9341	0.9996	0.9998	0.9963	0.8715	0.9996
	50k	0.9997	0.9677	0.0018	0.6027	0.9499	0.9994	0.9996	0.997	0.8887	0.9994
	100k	0.9998	0.9905	0.0081	0.5273	0.9599	0.9996	0.9996	0.9991	0.9065	0.9996
	200k	0.9998	0.9973	0.0251	0.4656	0.937	0.9994	0.9998	0.9998	0.9016	0.9994
	300k	0.9998	0.9987	0.0516	0.4532	0.9394	0.9995	0.9998	0.9998	0.9086	0.9995
	500k	0.9998	0.9989	0.0642	0.452	0.9464	0.9994	0.9989	0.9991	0.9177	0.9994
	800k	0.9998	0.9992	0.0738	0.5178	0.9397	0.9996	0.9989	0.9998	0.9107	0.9996
	1M	0.9998	0.9994	0.0826	0.3622	0.9347	0.9989	0.9987	0.9996	0.9146	0.9989

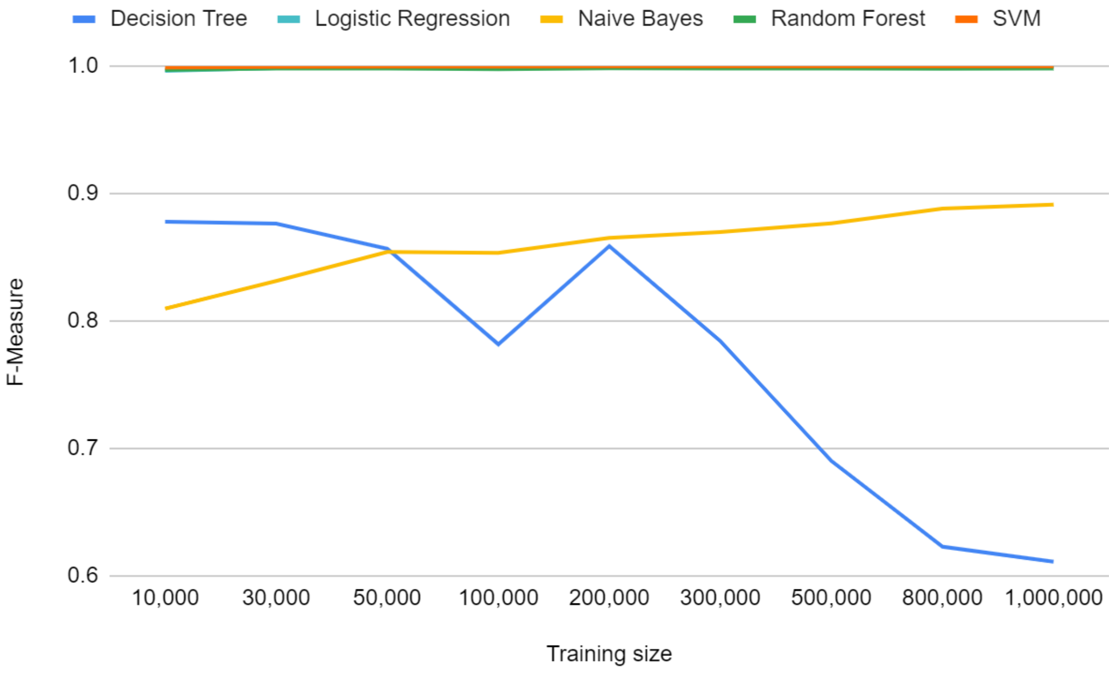


Figure 35. Progression of F-Measure mean for all the classical models in 1:1 ratio

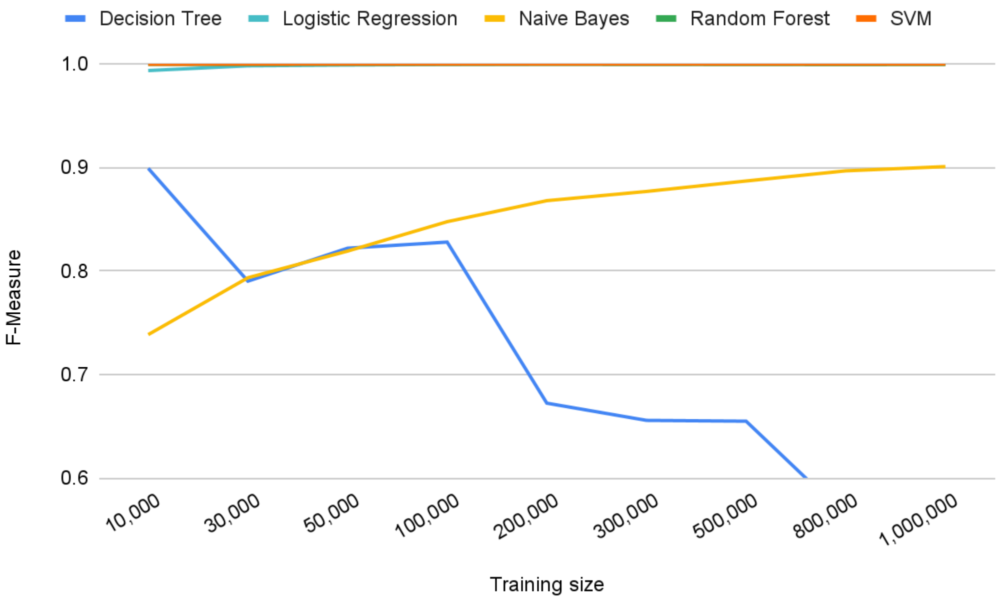


Figure 36. Progression of F-Measure mean for all the classical models in 1:5 ratio

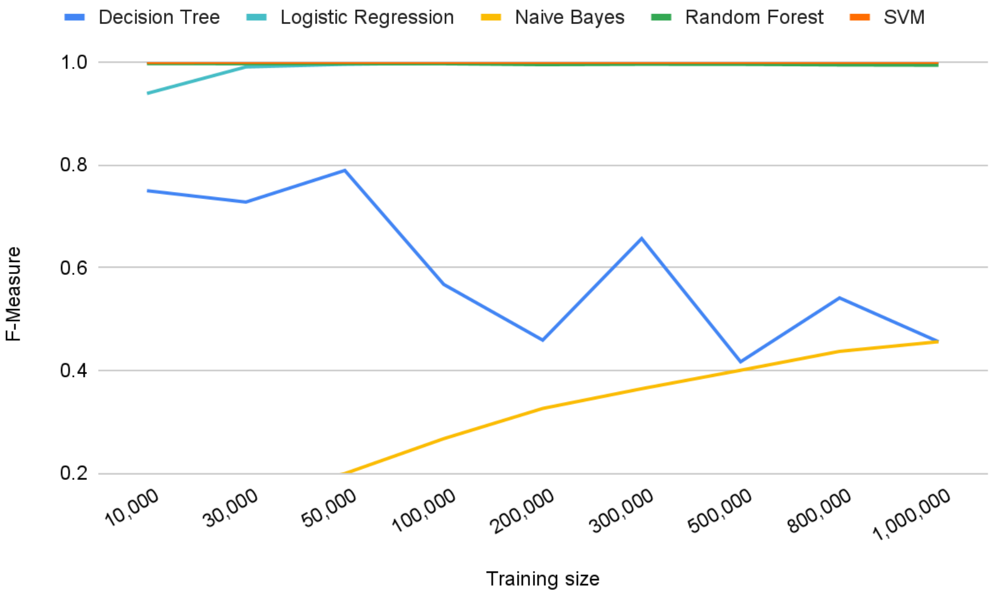


Figure 37. Progression of F-Measure mean for all the classical models in 1:15 ratio

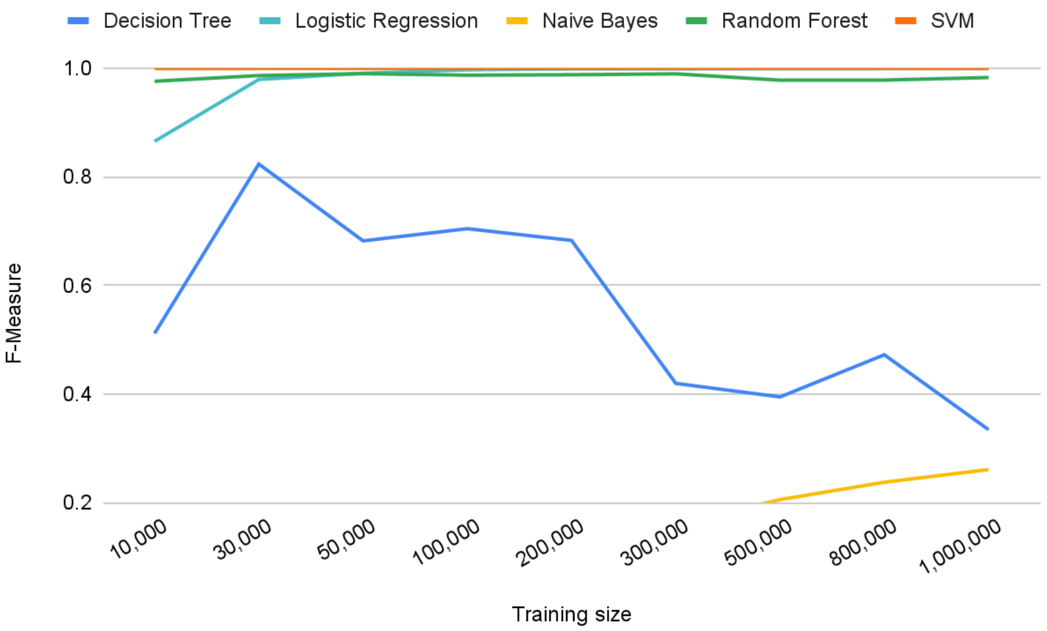


Figure 38. Progression of F-Measure mean for all the classical models in 1:25 ratio

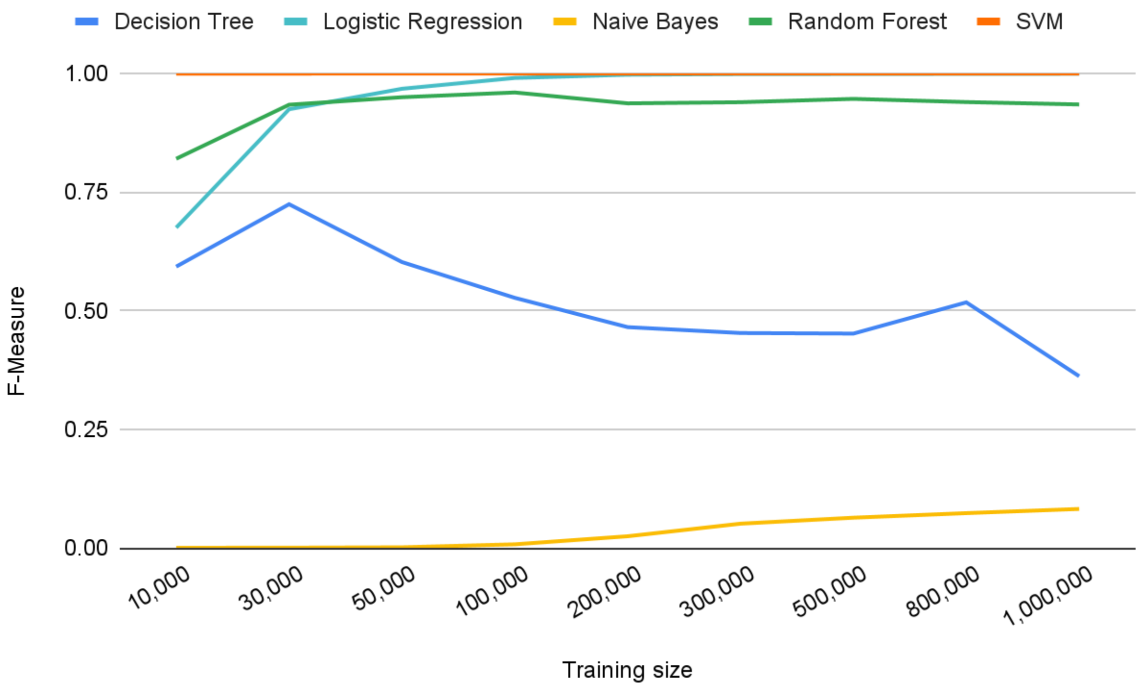


Figure 39. Progression of F-Measure mean for all the classical models in 1:50 ratio

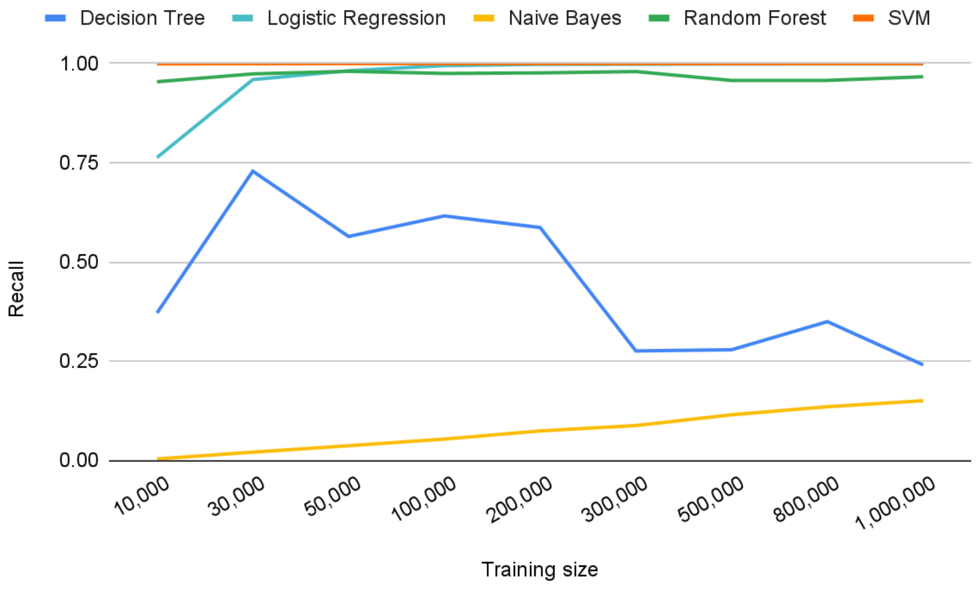


Figure 40. Progression of Recall mean for all the classical models in 1:25 ratio

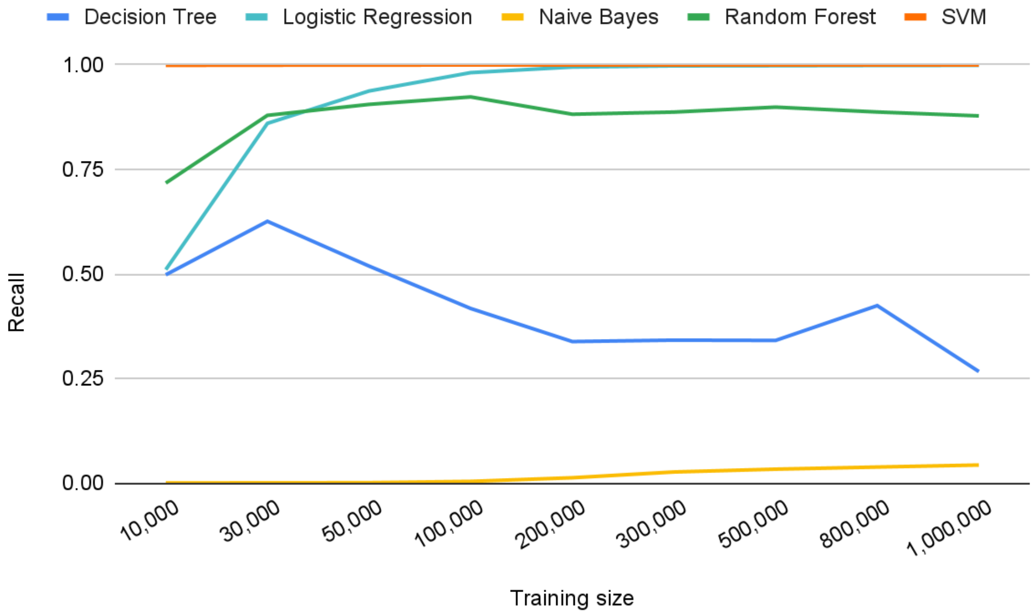


Figure 41. Progression of Recall mean for all the classical models in 1:50 ratio

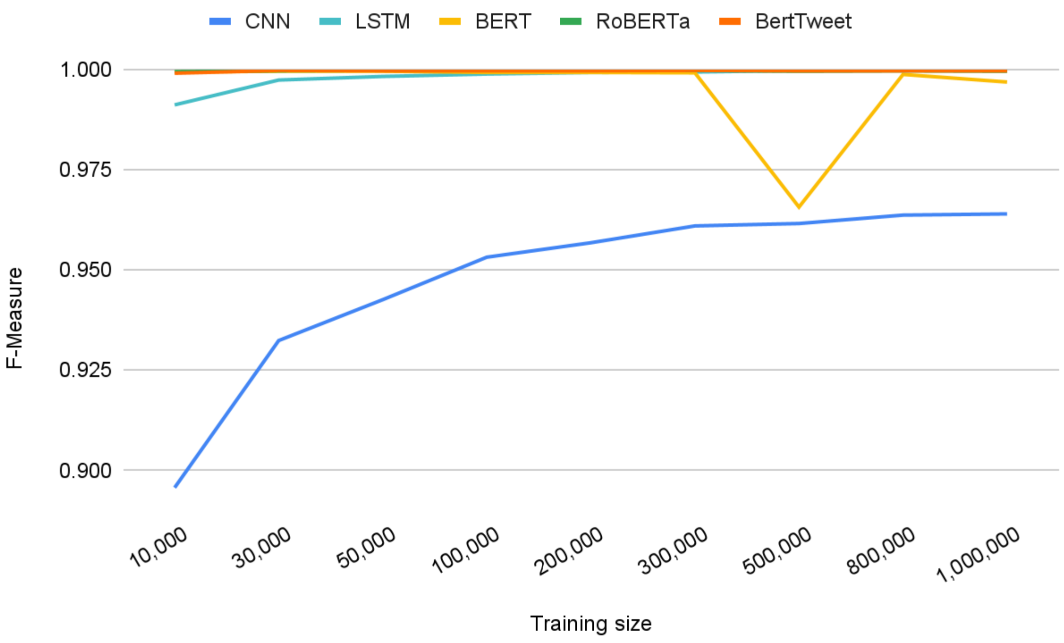


Figure 42. Progression of F-Measure mean for all the deep learning models in 1:1 ratio

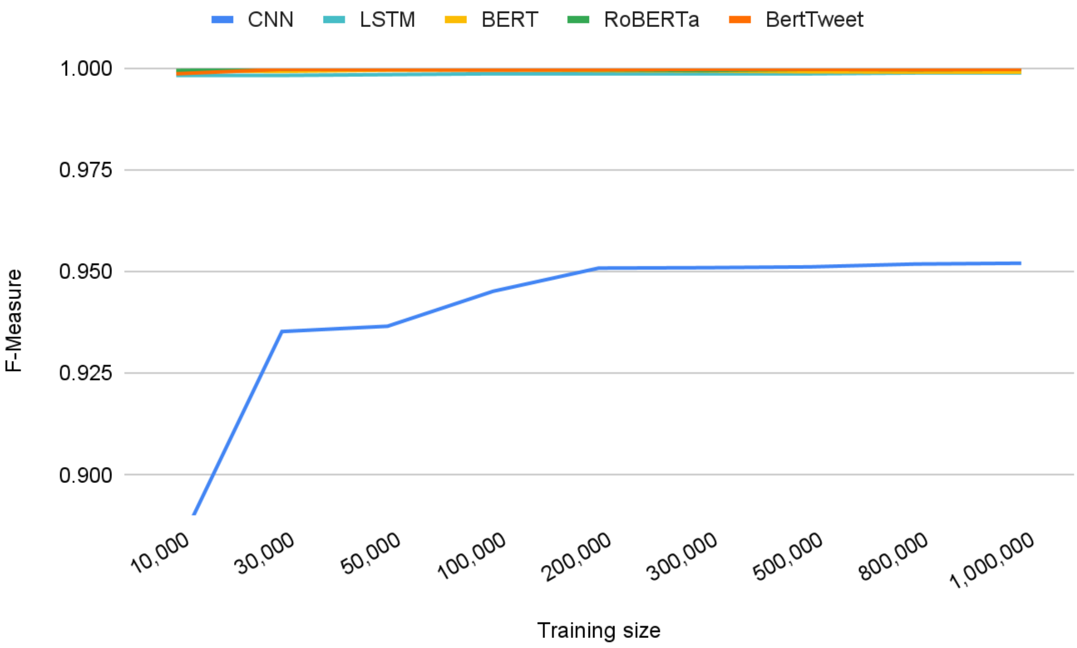


Figure 43. Progression of F-Measure mean for all the deep learning models in 1:5 ratio

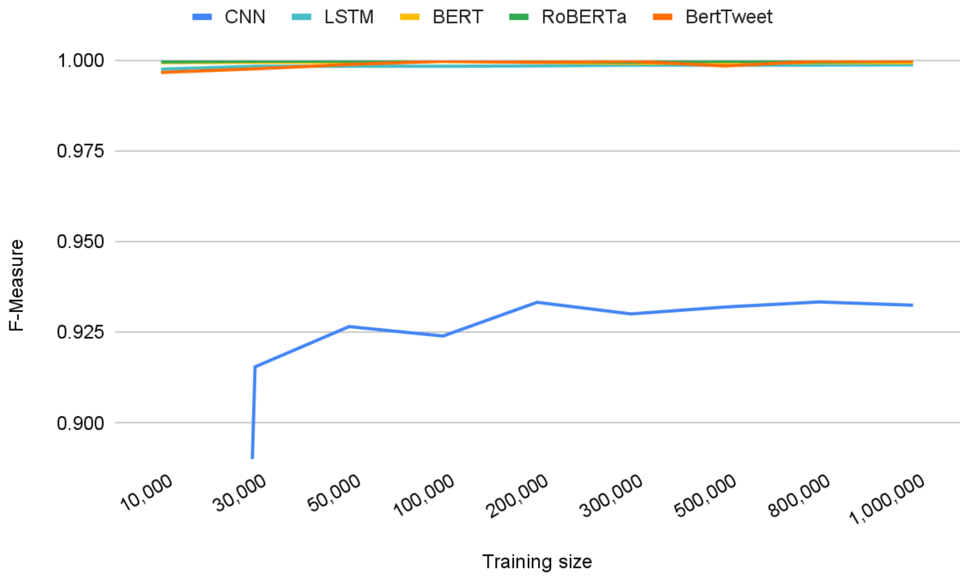


Figure 44. Progression of F-Measure mean for all the deep learning models in 1:15 ratio

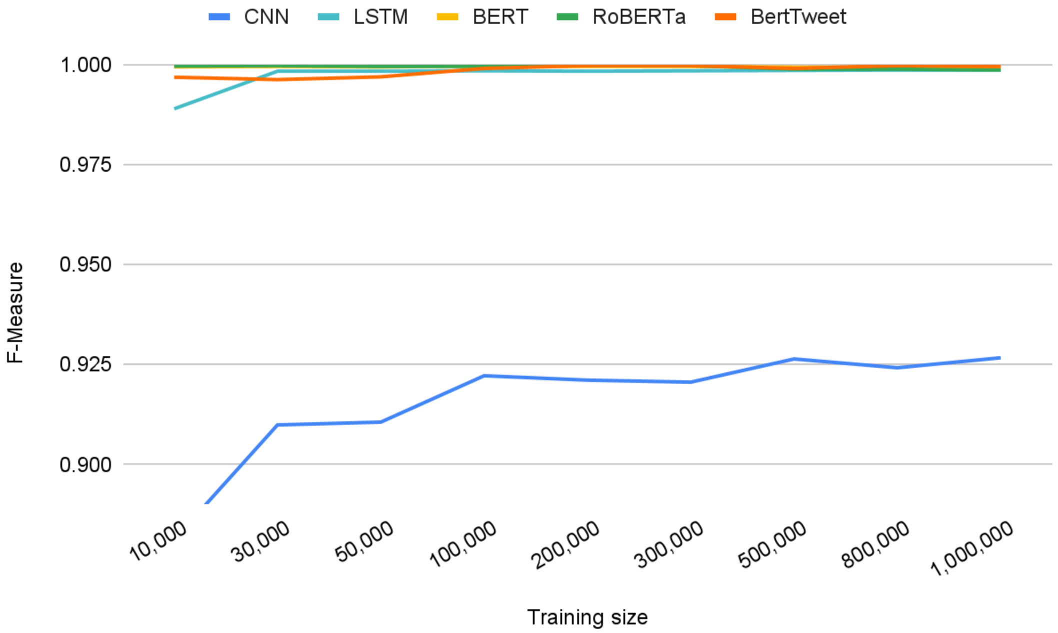


Figure 45. Progression of F-Measure mean for all the deep learning models in 1:25 ratio

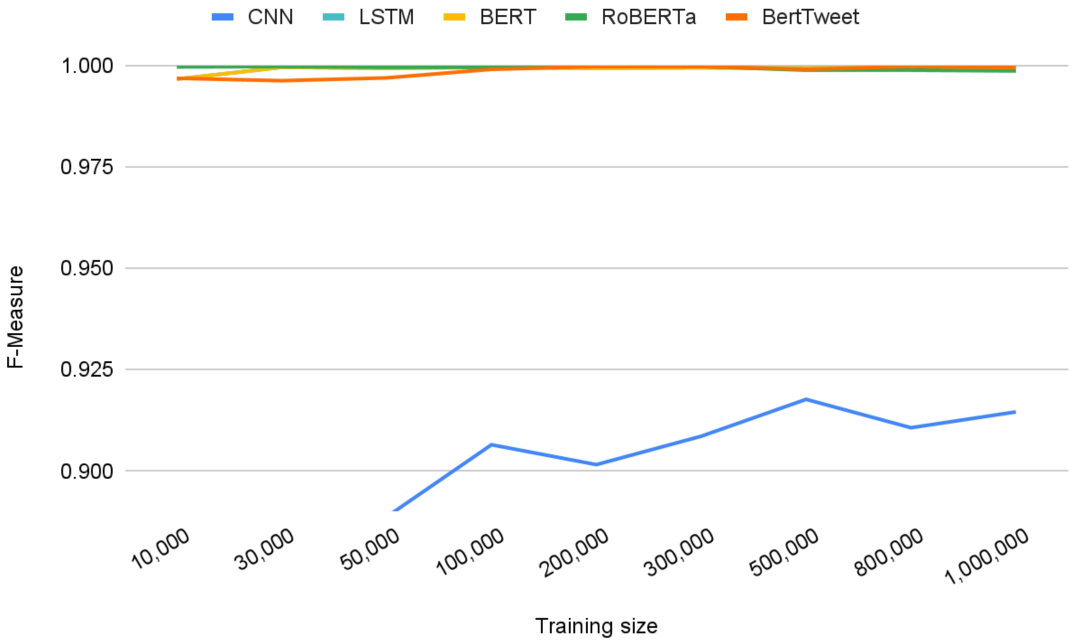


Figure 46. Progression of F-Measure mean for all the deep learning models in 1:50 ratio

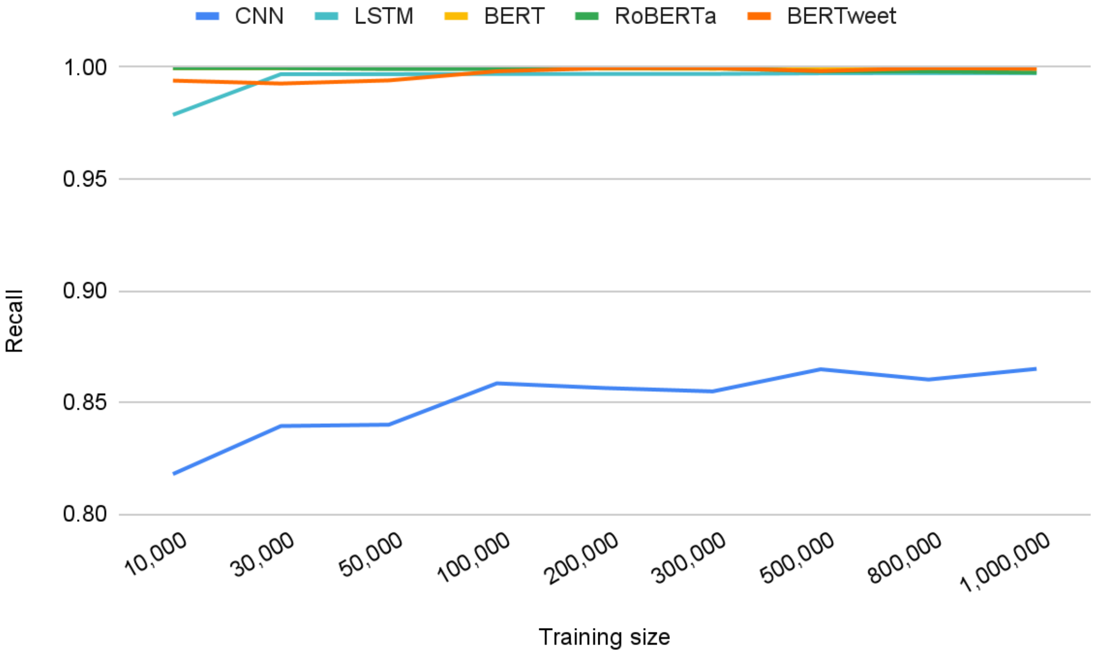


Figure 47. Progression of Recall mean for all the deep learning models in 1:25 ratio

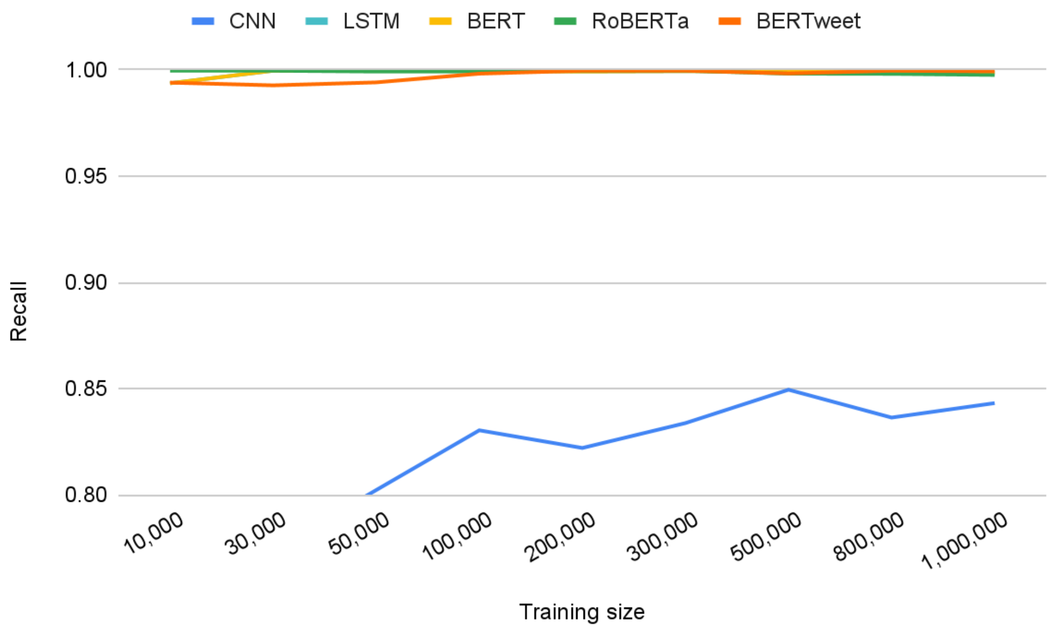


Figure 48. Progression of Recall mean for all the deep learning models in 1:50 ratio

We believe several factors to be the reason for the surprisingly very high performance. Firstly, the gold standard data contains well separated positive and negative samples, as the positive samples are definitely flu samples and the negative samples do not contain any flu tweets. The negative samples in the training data also do not contain any flu tweets. Secondly, the total number of samples in the test set are comparatively less and in most cases the models had consistently a few false negatives in the larger training sizes and more false positives in the smaller training size. Further, a recent research on detecting influenza tweets using Deep Learning[193] also demonstrated similar results using deep learning methods on a gold standard dataset containing samples in both English and Arabic languages. This demonstrates that flu class tweets can be easily separable on Twitter.

One observation in this application is that the heuristic contains patterns which are similar to the keywords used for collecting EPIC corpus. The heuristic can be utilized to collect relevant tweets from several datasets along with Twitter Regular stream. However, a machine learning model would be able to filter/ separate tweets that are missed by the heuristic. In this aspect, the Google flu trends project[194] was used by Google to identify trends and calculate predictions. This was based on Google searches, and projections were made as an early warning that matched the reports made public by the CDC. However two years after its inception, researchers identified an over-estimation resulting in inaccuracy and also determined that few searches were not relevant to "flu" and finally terminated the project in 2015. While the purpose of our application is different to the objective of the Google trends project, evidently, the methodology of utilizing the noisy data and training several machine learning models is able to identify the gold standard and can be definitely adopted for research with "Influenza".

To summarize, in this application, we utilized regular expressions as our heuristic and created a silver standard dataset of epidemic tweets. We experimented with a multi-classification setting and evaluated the performance of silver standard data using a large-scale epidemic corpus. The results from both class balanced and imbalanced experiments demonstrate the success in adopting a weak supervision approach in a multi-classification setting. Since the large-scale epidemic corpus is not a gold standard dataset, we performed empirical evaluation on one class of the silver standard data in a binary classification setting. We calculated theoretical bounds indicating that a minimum of 13,367 noisy samples were required for the least performing model. We experimented with sample size starting at 10,000 and systemically increased the sample size to 1 million and presented our results demonstrating the accuracy of theoretical bounds. While the empirical evaluation on one class of silver standard validated the silver standard dataset partially, we present successful results on evaluating a large multi class epidemic corpus, which has never been demonstrated in the past.

11 APPLICATION 4: SEPARATING HEALTH RELATED TWITTER CHATTER

From previous applications, we determined the extent of information overflow on Twitter, during crises, epidemics and in generic pharmacovigilance chatter. Additionally, Twitter data has been extensively utilized to analyze content on health-related topics, including influenza outbreak, alcohol abuse[195], dental pain[196], vaccinations[197], breast cancer[198], mental health[199] and childhood obesity[200]. Twitter can definitely be utilized for public health research as several users communicate openly and willingly about various health related topics. In this aspect, Sinnenberg et al.[201] presented a systematic review of use of Twitter in health research, where 137 articles were explored and constituted a new taxonomy to describe Twitter use in health research with 6 categories. With the advance of research on public health, several researchers explored utilizing machine learning on a multitude of health applications. Michael J. Paul and Mark Dredze presented a methodology to model and mine several health topics from Twitter and released over 144 million tweets[107,192]. Prieto et al. used regular expressions to filter relevant health tweets and tested their approach on 4 different health topics[202].

Apart from identifying different topics, several studies also demonstrated successful results in several applications which dived deep into a single health topic (Eg: Pregnancy). The Health Language Processing Lab at UPENN annotated a dataset for identifying women reporting adverse pregnancy outcomes on Twitter[50] and also presented a cohort study of drug safety[203] and monitored COVID-19 vaccine safety[204] during pregnancy. The Social Dynamics and Wellbeing Lab at Georgia Tech have been researching on several mental health issues like depression[205], suicide[206], and other self-disclosure posts on anxiety, stress and other mental health conditions[207].

While most of the studies and their methodology is available via research papers, labeled data is not available to reproduce the results or utilize the data for other applications. The primary reason for non-availability of the data corresponds to the sensitive nature of the text in the tweet. Hence a data annotation process is needed when large scale data has to be labeled.

In this application, our objective is to employ a weak supervision approach to evaluate the silver standard dataset on three different health topics. However, we did not find any publicly available multi-class gold standard datasets. Hence, we utilized weak supervision to create a “pseudo gold standard dataset” which utilizes a fraction of manual samples when compared to traditional manual labeling. We first curated a heuristic to generate a silver standard dataset containing three different health topics (i.e Pregnancy, Mental health and Heart Conditions) and further identified several sub topics for each topic. We trained several machine learning models in a multi classification setting using both class balanced and imbalanced samples. We calculated theoretical bounds and discussed our findings on the performance of a silver standard dataset in identifying the pseudo gold standard dataset.

11.1 Heuristic Creation

Unlike previous applications, we employed a basic heuristic, or "keywords", for this application. The objective of this application is to identify sub-classes for each class of health topic. Table 22 enlists the Health topic class, sub-class and the keywords used to retrieve the tweets.

Table 22. Health topics and sub classes with keywords

Health Topic	**Sub Class & keyword used**
Pregnancy	pregnant
	miscarriage
	abortion

Mental Health	anxiety attack
	insomnia
	panic attack
	suicidal
	depression
Heart Conditions	chest pain, chest pains
	heartburn
	acid reflux, reflux

11.2 Pseudo Gold Standard Dataset Creation

To create the pseudo gold standard data, we first use the heuristic to obtain relevant samples for each class. To label the samples, we adopted an iterative process where a small set of manual labeled (gold standard) data is utilized to train a machine learning model until an optimal performance is acquired. In this application, since there were no publicly available gold standard datasets, we manually labeled a small set of data for each health topic and subtopic. We then use the trained model to assign probabilities to the unseen samples. Based on a cut-off threshold, we labeled all the samples with probabilities greater than threshold as positive samples and added the samples to the manually labeled samples hence creating the pseudo gold standard dataset. Figure 49 presents the construction steps of the pseudo gold standard dataset.

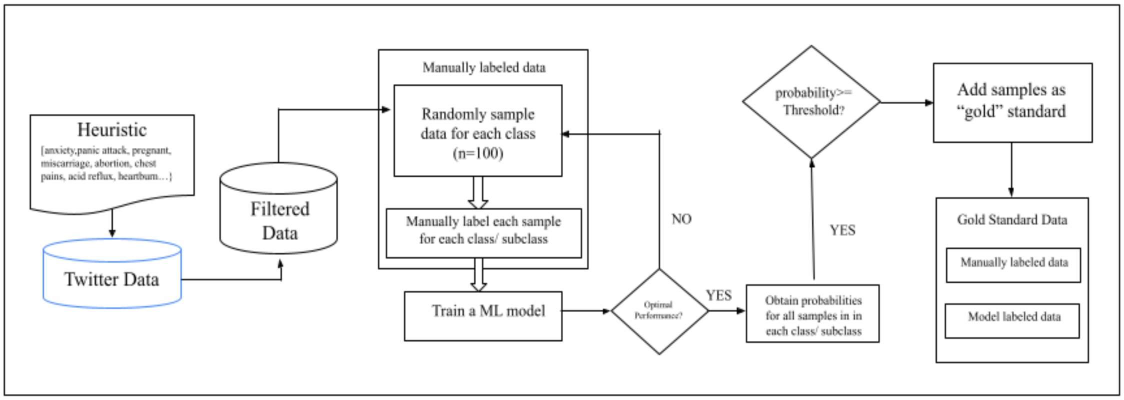

Heuristic
{anxiety,panic attack, pregnant, miscarriage, abortion, chest pains, acid reflux, heartburn…}
Twitter Data
Filtered Data
Manually labeled data
Randomly sample data for each class (n=100)
Manually label each sample for each class/ subclass
Train a ML model
Optimal Performance?
NO
YES
Obtain probabilities for all samples in in each class/ subclass
YES
probability>= Threshold?
Add samples as "gold" standard
Gold Standard Data
Manually labeled data
Model labeled data

Figure 49. Pseudo gold standard construction steps

In this application, we used the heuristic to filter data from the regular stream from 2018 to 2021. We preprocessed the filtered data and separated the data by each topic and subtopic. For each subtopic, we randomly sampled and manually labeled 100 samples for each subtopic. We ensured the samples were "self-reported" condition tweets. We labeled both positive (class 1) and negative (class 0) samples. We tested several classical models in a binary classification setting and finalized the SVM model with "rbf kernel" as our machine learning model since it had the best performance. Our target is to train a classifier with performance greater than 80%. We used F-Measure as our metric and iteratively labeled the data until optimal performance was achieved. Subsequently, we used the trained model to assign probabilities to all the unseen samples of a subtopic. We iteratively repeated the same process for each subtopic. Based on the distribution of probabilities, we determined a cut off threshold for each subtopic and then labeled all the samples greater than threshold as positive samples. Finally, we added the model labeled samples to the manually labeled samples and created the pseudo gold standard dataset. Table 23 presents the number of samples in the pseudo gold standard dataset for each subtopic.

Table 23. Total number of samples in Pseudo gold standard dataset

Class	Sub Class	Manually labeled samples	Model labeled samples	Total tweets in Pseudo Gold Standard	Threshold used
Pregnancy	pregnant	167	572	739	0.98
	miscarriage	134	153	287	0.96
	abortion	110	162	272	0.95
Mental Health	anxiety attack	116	2,266	2,382	0.99
	insomnia	137	231	368	0.987
	panic attack	143	460	603	0.98
	suicidal	138	325	463	0.99
	depression	100	3,626	3,726	0.99
Heart Conditions	acid reflux	101	177	278	0.67
	chest pain	110	382	492	0.98
	heartburn	101	305	406	0.98

11.3 Generating the Silver Standard dataset

To create the silver standard dataset, we applied the heuristic on Publicly available datasets and Regular Stream. We mined a total of 7.3 billion tweets from the two sources and separated tweets for 11 subclasses.

11.3.1 Regular Stream Details

In this application, we used tweets collected between 2018 and 2021. Table 24 lists the details of tweets collected and filtered from the Twitter Stream. We used only clean English tweets from this stream. The % tweets column represents the percentage of tweets filtered from the clean tweets. There is an increase in the count of relevant tweets in 2020 due to Covid-19, where users actively tweeted about battling with mental health issues due to lockdown and pandemic. While we obtained a total of 853,758 tweets from the regular stream, after removing duplicate tweets and tweets that existed in gold standard, only 799,554 tweets were filtered from the regular stream. Only 0.04% of relevant health tweets were identified from the regular stream.

Table 24. Total number of filtered tweets from Regular Stream

Year	Filtered Tweets	Percentage of filtered tweets
2018	184,999	0.04
2019	251,321	0.04
2020	267,440	0.03
2021 (Jan - May)	95,794	0.03
Total	**799,554**	**0.04**

11.3.2 Publicly Available Datasets

We filtered tweets from 34 different publicly available datasets using the heuristic. The publicly available datasets yielded more tweets than the regular stream, since the datasets contained tweets that were collected since 2013. Only 2 datasets were related to Health (Health Care and ATAM dataset). While the other datasets were unrelated to health, we could obtain a significant number of tweets from the publicly available datasets. A total of **1,924,235** tweets were filtered using the heuristic from a total of 3,050,058,283 tweets. Table 25 presents the data collection results from the publicly available datasets. While tweets from only two datasets were relevant to the current application, we observed that several other datasets (Eg: Natural Disasters, Election, Women's March) contained a significant number of tweets. A quick analysis into the filtered tweets determined that a lot of health relevant chatter was frequent during elections and crises. Various health care policies are usually discussed during elections and crisis situations always have health related tweets. For example, during natural disasters (Eg: hurricanes), users tweeted about the impact on mental health due to loss of properties while also dealing with misplacement and damage.

Table 25. Filtered tweets from publicly available dataset

Dataset	Filtered tweets	Percentage of filtered tweets
2016 presidential election[75]	70,185	0.14
Solar Eclipse[76]	331	0.02
hurricaneHarvey[96]	1,890	0.09
Hurricane Florence[97]	4,331	0.31
Hurricane Florence[101]	1,467	0.20
Hurricane Harvey[98]	1,430	0.16
Hurricane Irma[96]	1,414	0.06
Hurricane Maria[102]	40	0.02
Hurricane Sandy[103]	1,926	0.04
Hurricane Dorian[104]	258	0.06
Hurricane Dorian[105]	2711	0.16
Election 2012[77]	82,873	0.38
Datarelease[78]	62,221	0.20
Beyond the Hashtag[79]	8,267	0.11
Climate Change[80]	15,937	0.20
Trump Tweet Ids[81]	6,851	0.07
Health Care[82]	99,757	0.44
2018 Congregational Election[106]	13,354	0.14
News Outlets[89]	180,018	0.20
Women's March[83]	5,311	0.41
US Govt Ids[84]	14,417	0.21
End of Term[85]	7,916	0.19
Nipsey Tweets[86]	423	0.03
Winter Olympics[87]	328	0.02
Dallas Shooting[88]	259	0.02
Charlottesville[90]	70	0.02
Twitter-Events-2012-2016[91]	31,018	0.09
115th U.S. Congress Tweet Ids[99]	3,436	0.22
Immigration Exec Order[92]	647	0.03

Irish news English tweets [93]	68,741	0.15
Black Lives Matter[94]	2,080	0.08
2020 Presidential Election[100]	246,618	0.17
Tweets to Donald Trump[95]	140,304	0.08
ATAM dataset[107]	847,406	1.13
Total	**1,924,235**	**0.25**

Combining our data filtered from the Twitter stream and hydrated datasets, we obtained 2,516,574 tweets from 11 different subclasses. To create the silver standard dataset, we removed duplicate tweets and preprocessed the tweet text by removing emojis, emoticons, URLs and striped white spaces. We also lowercased the tweet text for data standardization. Table 26 lists the number of tweets in the silver standard dataset for each class and subclass.

Table 26. Number of tweets in silver standard dataset

Class	**Sub Class**	**Total number of Tweets**
Pregnancy	pregnant	518,240
	miscarriage	21,866
	abortion	620,724
Mental Health	anxiety attack	60,269
	insomnia	308,158
	panic attack	51,695
	suicidal	75,226
	depression	703,386
Heart Conditions	acid reflux	7,468
	chest pain	94,197
	heartburn	55,345
Total		**2,516,574**

11.4 Calculating Theoretical Bounds

To compute the theoretical bounds, we trained several machine learning models on the gold standard data and presented the theoretical bounds for a high and low performing model. We split the gold standard data into 75:25 for training and test and obtained the accuracy of the machine learning models. We use accuracy to calculate the theoretical bounds.

11.4.1 Calculating theoretical bounds for a high performing model

In this computation, we consider "BERTweet" to be a model with high performance with an accuracy score of 99%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score(0.99), and clean samples (m =10,016), the minimum number of noisy samples are calculated below

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 10,016/(1-2*(1-0.99)))**2

noisy samples = **10,429**

We would require 4,780 noisy samples to achieve the performance similar to the performance of models trained on 10,429 clean samples for a high performing model.

11.4.2 Calculating theoretical bounds for a low performing model

In this computation, we consider "Naive Bayes" to be a model with low performance with an accuracy score of 54%. For an error bound($\gamma = 0.05$), probability($\delta = 0.05$), accuracy score (0.54), and clean samples (m = 10,016), the minimum number of noisy samples are below

noisy samples = m/ (1-(2*(1-τ)))**2

noisy samples = 10,016/(1-2*(1-0.54)))**2

noisy samples = 1,564,999

We would require 13,367 noisy samples to achieve the performance similar to the performance of models trained on 10,016 clean samples for a high performing model.

To summarize, the minimum number of noisy samples required for the best performing model (BERTweet) is 10,429 and the minimum number of noisy samples required for the least performing model (Naive Bayes) is 1,564,999.

11.5 Experimental Setup

We experimented with both class balanced and extremely class imbalanced samples in a multi-classification setting. A stratified ratio of 75-25(train-validation) was used to split the silver standard dataset and to train five different classical models (SVM, Decision Tree, Naive Bayes, Logistic Regression and Random Forest) and five different deep learning models (BERT, BERTweet, RoBERTa, CNN and LSTM). The validation data was used to either improve the performance of the model or terminate the learning process when there is no significant improvement. To test the models, we used the pseudo gold standard data for each subclass. We performed the experiments in two different settings, as detailed below.

Experiment 1: A balanced corpus of silver standard dataset matched to the minimum number of samples available was used as training data. In the silver standard data, “acid reflux” subclass contains the least number of samples when compared to the other subclasses hence all the classes in this experiment were sampled to minimum number of samples (7,468). The test set is not balanced and we utilized all the samples in the test set for each subclass.

Experiment 2: All the samples from the silver standard dataset were utilized as the training data. This is a heavily imbalanced experiment. The test set is not balanced and we utilized all the samples in the test set for each subclass.

11.6 Results

As in the previous experiments, we calculated Precision, Recall, F-Measure, Accuracy. Additionally we computed the metrics for both individual and weighted metrics for all the subclasses. Tables 27 and 28 present the results of F-measure for all the models for all subclasses for experiment 1 and 2.

Table 27. F-Measure for each subclass for Experiment 1

Class	Sub Class	LR	SVM	DT	RF	NB	B	BT	RB	CNN	LSTM
Pregnancy	pregnant	0.9601	0.9633	0.5867	0.9583	0.9039	0.9293	0.9311	0.8928	0.6613	0.9423
	miscarriage	0.9594	0.9671	0.1444	0.9305	0.7232	0.9203	0.9242	0.9044	0.8381	0.9412
	abortion	0.9084	0.9032	0.3640	0.9228	0.7075	0.8826	0.9209	0.7741	0.7737	0.9102
Mental Health	anxiety attack	0.9897	0.9926	0.5474	0.9881	0.8979	0.8613	0.9762	0.7867	0.7652	0.9230
	panic attack	0.9788	0.9804	0.2959	0.9772	0.7386	0.9749	0.9757	0.9812	0.7417	0.9789
	insomnia	0.9797	0.9579	0.2558	0.9642	0.8293	0.9600	0.9707	0.5198	0.8279	0.8848
	suicidal	0.9817	0.9818	0.7931	0.9774	0.7933	0.6338	0.9517	0.9636	0.8291	0.9425
	depression	0.9915	0.9910	0.6286	0.9907	0.9074	0.9861	0.9852	0.9909	0.8419	0.9601
Heart Conditions	chest pain	0.9859	0.9919	0.2813	0.9839	0.9421	0.9828	0.9899	0.9889	0.6596	0.9879
	acid reflux	0.9782	0.9764	0.3938	0.9874	0.8683	0.9745	0.9798	0.9818	0.3755	0.9780
	heartburn	0.9826	0.9839	0.3928	0.9726	0.8908	0.9714	0.9763	0.9766	0.7591	0.9828

Table 28. F-Measure for each subclass for Experiment 2

Class	Sub Class	LR	SVM	DT	RF	NB	B	BT	RB	CNN	LSTM
Pregnancy	pregnant	0.9321	0.9395	0.4315	0.8848	0.6588	0.9435	0.9474	0.9365	0.7862	0.9077
	miscarriage	0.8748	0.8956	0.3046	0.7216	0.0000	0.9081	0.9145	0.8757	0.8931	0.9125
	abortion	0.9051	0.9134	0.2057	0.9094	0.7318	0.9173	0.9234	0.3467	0.8149	0.9222
	anxiety attack	0.9886	0.9890	0.1607	0.8779	0.1236	0.8400	0.9716	0.7880	0.6922	0.9615

Mental Health	panic attack	0.9796	0.9796	0.2033	0.9770	0.3785	0.9788	0.9781	0.9733	0.8849	0.9797
	insomnia	0.9577	0.9590	0.1721	0.9159	0.6394	0.9679	0.9719	0.9691	0.7812	0.9602
	suicidal	0.9408	0.9431	0.1967	0.7880	0.0000	0.9422	0.8806	0.9431	0.8292	0.9412
	depression	0.9830	0.9841	0.6193	0.9138	0.6627	0.9150	0.9868	0.9890	0.8515	0.9847
Heart Conditions	chest pain	0.9859	0.9838	0.7798	0.9780	0.7338	0.9800	0.9839	0.9859	0.6652	0.9879
	acid reflux	0.9490	0.9531	0.2691	0.8835	0.0000	0.9588	0.9761	0.9609	0.9055	0.9594
	heartburn	0.9704	0.9705	0.1245	0.9265	0.1480	0.9691	0.9751	0.9728	0.8156	0.9502

The class balanced experiment findings (Experiment 1) showed that the Transformer models were consistently superior to the neural network models. Three classical models (Logistic Regression, SVM, and Random Forest) outperformed Decision Tree and Naive Bayes in terms of performance. Surprisingly, the overall performance of the machine learning models for the two experiments did not differ much. However, we observed a performance drop in subclasses in the extremely imbalanced experiment. To determine the rationale, we plotted confusion matrices, which display the prediction distribution across classes for the RoBERTa and BERT models for experiment 2. From Figure 50 and 51, we observe that a few subclasses were incorrectly classified. However, they were classified under the same parent class. For example, several “abortion” samples were incorrectly classified as “pregnancy” or “miscarriage”. Several “anxiety attack” samples were incorrectly classified as “abortion” and “panic attack”. We believe that the model could calculate similarities between subtopics and hence misclassified the subclasses, since they are under the same parent class. Hence to determine the performance of models at parent class level, we designed two additional experiments.

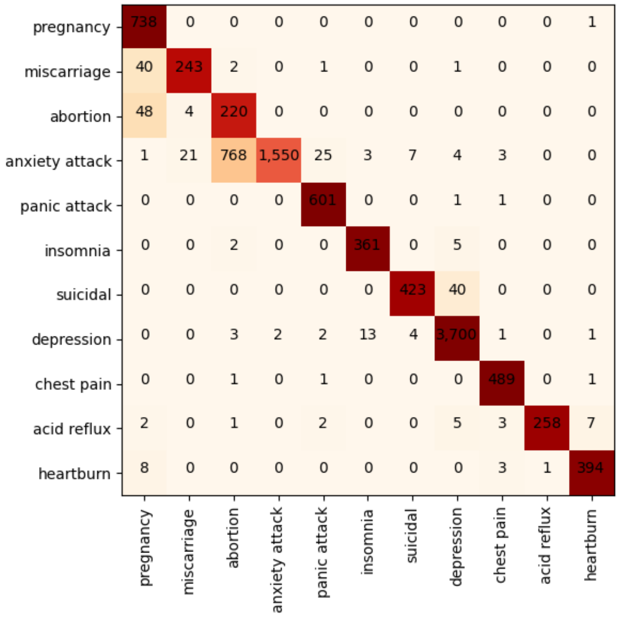


Figure 50. Confusion Matrix for RoBERTa model for experiment 2

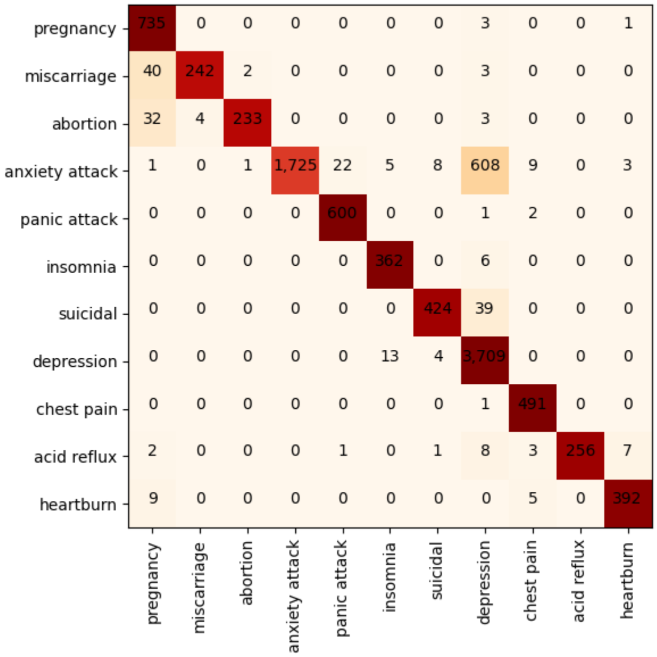


Figure 51. Confusion Matrix for BERT model for experiment 2

11.6.1 Experiments with Aggregations

In this set of experiments, we determined how the silver standard data performed against aggregated class data instead of a multi classification on different subclasses. In other words, we set to experiment with 3 different class level multi classification models. In this scenario, instead of 11 subclasses, we have 3 classes and all the labels of subclasses have been changed to class level labels. For example, in the pregnancy class, the subclasses pregnant, abortion and miscarriage were labeled as "pregnancy" samples. We use similar experiments as above, and the experiment setup is detailed below.

Experiment 3: A balanced corpus of silver standard dataset matched to the minimum number of samples available was used as training data. In the silver standard data, "heart conditions" class contains the least number of samples when compared to the other classes and hence all the classes in this experiment were sampled to a minimum number of samples (7,468). The test set is not balanced, and we utilized all the samples in the test set for each class.

Experiment 4: All the samples from the silver standard dataset were utilized as the training data. This is a heavily imbalanced experiment. The test set is not balanced and we utilized all the samples in the test set for each class.

While we calculated precision, recall, F measure and accuracy across all classes, we present only F-measure results for experiments 3 and 4 in Tables 29 and 30.

Table 29. Class level F-measure for all models for experiment 3

Class	**LR**	**SVM**	**DT**	**RF**	**NB**	**B**	**BT**	**RB**	**CNN**	**LSTM**
pregnancy	0.9942	0.9962	0.5261	0.9938	0.8148	0.9935	0.9977	0.9935	0.8275	0.9901
mental health	0.9989	0.9992	0.8741	0.9983	0.9616	0.9985	0.9993	0.9986	0.9302	0.9984
Heart conditions	0.9919	0.9932	0.6077	0.9932	0.8849	0.9945	0.9962	0.9949	0.6769	0.9915

Table 30. Class level F-measure for all models for experiment 4

Class	LR	SVM	DT	RF	NB	B	BT	RB	CNN	LSTM
pregnancy	0.9935	0.9935	0.3532	0.9760	0.8972	0.9831	0.9079	0.9481	0.8804	0.9885
mental health	0.9983	0.9984	0.6944	0.9916	0.9481	0.9963	0.9829	0.9911	0.9615	0.9983
heart conditions	0.9867	0.9867	0.6063	0.9311	0.5323	0.9774	0.9712	0.9796	0.8110	0.9876

The results from experiments 3 and 4 determine the performance of the silver standard dataset in identifying the pseudo gold standard dataset at class level. The results from experiments 1 and 2 determined the performance at subclass level. In both class balanced and imbalanced experiments at class and subclass level, the best models could successfully identify the pseudo gold standard dataset with a performance greater than 90%.

To summarize, in this application, we created a simple heuristic and curated silver standard dataset from the regular stream and publicly available datasets. Since there were no publicly available multi-class gold standard health topic datasets, we used a weak supervision approach to curate a gold standard dataset. We experimented with both class balanced and imbalanced samples at both class and subclass level and demonstrated the performance of machine learning models in identifying the pseudo gold standard dataset. We calculated theoretical bounds and determined that a total of 1,564,999 noisy samples were required for the least performing model and 10,429 noisy samples were required for the best performing model when a total of 10,016 clean samples were available. Since the gold standard is not manually validated, we experimented with all the samples of the data (>2 million) to demonstrate that the theoretical bounds are accurate. While we experimented with 3 different health topics and 11 distinct sub topics in this application, the methodology to curate the gold standard and silver standard can easily be extended and adapted to several other health topics.

12 SUMMARY

In this study, we demonstrated the viability of utilizing noisy social media data using weak supervision. Our study is motivated by the drawbacks of supervised learning which require massive amounts of labeling which is a tedious and expensive process. In this study, we utilized a heuristic based approach to label data and generated silver standard datasets for four different applications. In our first application, “Identifying drug mentions from Twitter”, we tested the weak supervision approach in a binary classification setting using a drug dictionary as our heuristic. In the second application, “Characterizing three different types of Natural Disasters: Hurricanes, Earthquakes and Floods”, we utilized bi-grams in conjunction with a list of generic natural disaster terms as our heuristic and tested the approach in a binary classification setting. In the third application, “Detecting epidemic tweets and evaluation of large scale epidemic corpus”, we employed regular expressions as our heuristic and tested the weak supervision approach in both binary and multi-classification. In our final application, “Separating health related Twitter chatter”, we used a weak supervision approach to generate a “pseudo gold standard dataset” and tested the noisy silver standard data in a multi-classification setting. For all applications, we utilized a gold or pseudo gold standard dataset to validate the approach and extensively evaluated the silver standard dataset on several training samples and class imbalances. We computed the theoretical bounds for each application and verified the accuracy of the theoretical bounds for each application. The results from the applications evidently demonstrates that the silver standard dataset identifies the gold standard dataset. Our findings in the four applications indicate that social media data can be utilized for weak supervision in both binary and multi classification settings.

13 LIMITATIONS

The study examined the usage of noisy labels for four different applications using different kinds of heuristics in different classification settings. We observed a few limitations for this work. While the methodology demonstrated successful results on broader applications, it might not perform well for applications which require fine-grained or specific labels. For example, we tested the methodology on a “hate speech detection” application[208] and the models could not differentiate between “hate” and “counter-hate” labels. Additionally we experimented with data augmentation and added more noisy data to the models which yielded poor results. We believe that the methodology might obtain poor results for specific applications which require detailed labels. A few other applications we haven't tested the methodology but believe might obtain poor performance are “Fake news detection”, “Differentiating between misinformation, fake news and disinformation”, “Differentiating the variants of flu virus”, “Characterizing Covid-19 strain variants”, “Separating or understanding the differences between bots and humans tweets relevant to a topic”. Secondly, Labeling functions with frameworks like snorkel offer more complex functionalities, especially with ambiguous tweets. For such ambiguous tweets, a heuristic might incorrectly label and might bring more noise into the silver standard dataset. In this study, we applied rule based, pattern matching and pre-trained models to obtain labeling data using a heuristic. Additionally, labeling functions can incorporate distant supervision and crowdworker labels into their framework. Labeling functions with the Snorkel framework also offer several summary statistics like “polarity”, “coverage”, “overlaps”, “conflicts”, “empirical accuracy” which are utilized to understand and analyze the labeling functions. We have to create separate functions to obtain the statistics when using only a heuristic based method which is time consuming. However, a heuristic is easier to use and can be easily adapted by non-computer science researchers when compared to labeling functions. Finally, when using a weak supervision approach, multiple machine learning models must be experimented, as there are no pre-approved models. Despite the limitations, the methodology can definitely be extended to other applications and obtain results similar to supervised learning.

14 FUTURE WORK

There is a great scope for expansion of this work in the future. Recently, Ratner et al. released Wrench[30], a comprehensive benchmark for weak supervision. They released 14 different benchmark datasets for weak supervision which can be utilized for several machine learning tasks such as classification and sequence tagging. None of the datasets in the study were extended to include social media data. This work can be expanded by creating a few benchmark weak supervision social media datasets. Secondly, social media data might be deleted or removed, resulting in loss of data. Hence to further help retain important signals from social media data, several BERT models can be trained with silver standard dataset and the pre-trained models can be released through Hugging Face[135] which can be utilized for several downstream tasks. Thirdly, since heuristics were used in this study, labeling functions could be incorporated in future studies and the efficiency of utilizing a labeling function versus a heuristic for social media data can be determined.

Furthermore, all the applications in this work never utilized the weak supervision methodology in the past. Since the methodology is based on using a labeling heuristic, this approach can certainly be extended to several applications. Few directions where social media data could be used for weak supervision applications are, “Classifying different emotions”, “Characterizing patterns of stock market”, “stance detection”. We demonstrated separation of health chatter between several health topics in Chapter 11. However, weak supervision can be extended to individual health applications like “identifying adverse pregnancy outcomes”, “detecting adverse mental health events”, “usage of stimulants and opioids”, “identifying symptoms associated with health conditions”, and “early detection of health conditions”. Few directions where weak supervision methodology could be applied outside of social media data are “Information extraction”, “multi-instance learning", "Automatic Speech Recognition", "Identifying adverse drug reactions", "Identifying cancer aggressiveness using weak patterns", "Classifying Unstructured Clinical Notes". Additionally, based on limitations presented in Chapter 13, there is an immense scope for expansion of weak supervision research in applications which require fine-grained labels. New methodologies or frameworks could be created which address the limitations.

15 CONCLUSION

In this work, we tested the theory of noisy learning using social media data to train machine learning models in a weak supervision setting. We utilized a heuristic based approach to label data and created large scale silver standard datasets. We mined over 16 billion tweets in a span of 3 years, from three different sources and documented the data collection process along with advantages and limitations for each kind of data collection. We identified four applications where the weak supervision methodology was not utilized in the past and exhaustively experimented with numerous sample sizes, class imbalances, and machine learning models in both binary and multi classification settings on four different applications. Additionally, we adopted a weak supervision approach to build a pseudo gold standard dataset when no social media gold standard datasets were available for the health application. Subsequently, after extensive evaluations, we conclude that noisy unstructured social media data can be utilized for weak supervision. Additionally, we draw the conclusion that social media data is useful for applications when employing generic labels rather than fine grained labels. We contribute a methodology that can be extended to several other applications by changing the heuristic and the curating silver standard data programmatically. We documented the limitations and additionally presented directions to expand this work for future research.

REFERENCES

1. Ratner, A., Bach, S., Varma, P. & Ré, C. Weak supervision: the new programming paradigm for machine learning. *Hazy Research. Available via https://dawn. cs. stanford. edu//2017/07/16/weak-supervision/. Accessed* 05–09 (2019).
2. Ratner, A. *et al.* Snorkel: rapid training data creation with weak supervision. *VLDB J.* **29**, 709–730 (2020).
3. Angluin, D. & Laird, P. Learning from noisy examples. *Mach. Learn.* **2**, 343–370 (1988).
4. Wang, Y. *et al.* A clinical text classification paradigm using weak supervision and deep representation. *BMC Med. Inform. Decis. Mak.* **19**, 1 (2019).
5. Deriu, J. *et al.* Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification. in *Proceedings of the 26th International Conference on World Wide Web* 1045–1052 (International World Wide Web Conferences Steering Committee, 2017).
6. Agarwal, V. *et al.* Learning statistical models of phenotypes using noisy labeled training data. *J. Am. Med. Inform. Assoc.* **23**, 1166–1173 (2016).
7. Dehghani, M., Severyn, A., Rothe, S. & Kamps, J. Learning to Learn from Weak Supervision by Full Supervision. *arXiv [stat.ML]* (2017).
8. Zamani, H. & Bruce Croft, W. On the Theory of Weak Supervision for Information Retrieval. *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* (2018) doi:10.1145/3234944.3234968.
9. Saab, K. *et al.* Doubly Weak Supervision of Deep Learning Models for Head CT. in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* 811–819 (Springer International Publishing, 2019).

10. Fries, J. A. *et al.* Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences. *Nat. Commun.* (2019) doi:10.1101/339630.

11. Saab, K., Dunnmon, J., Ré, C., Rubin, D. & Lee-Messer, C. Weak supervision as an efficient approach for automated seizure detection in electroencephalography. *NPJ Digit Med* **3**, 59 (2020).

12. Dai, J., He, K. & Sun, J. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. in *Proceedings of the IEEE international conference on computer vision* 1635–1643 (2015).

13. Xia, W., Domokos, C., Dong, J., Cheong, L.-F. & Yan, S. Semantic segmentation without annotating segments. in *Proceedings of the IEEE international conference on computer vision* 2176–2183 (2013).

14. Blaschko, M., Vedaldi, A. & Zisserman, A. Simultaneous Object Detection and Ranking with Weak Supervision. in *Advances in Neural Information Processing Systems* (eds. Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R. & Culotta, A.) vol. 23 235–243 (Curran Associates, Inc., 2010).

15. Weng, Z., Varma, P., Masalov, A., Ota, J. & Ré, C. Utilizing Weak Supervision to Infer Complex Objects and Situations in Autonomous Driving Data. in *2019 IEEE Intelligent Vehicles Symposium (IV)* 119–125 (2019).

16. Khattar, S. *et al.* Weak Supervision for Time Series: Wearable Sensor Classification with Limited Labeled Data. (2019).

17. Bishop, C. M. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Comput.* **7**, 108–116 (1995).

18. Suchanek, F. M., Ifrim, G. & Weikum, G. Combining linguistic and statistical analysis to extract relations from web documents. in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* 712–717 (Association for Computing Machinery, 2006).

19. Kulkarni, S. R., Lugosi, G. & Venkatesh, S. S. Learning pattern classification-a survey. *IEEE Trans. Inf. Theory* **44**, 2178–2206 (1998).

20. Anthony. Probabilistic analysis of learning in artificial neural networks: The PAC model and its variants. *Neural Computing Surveys*.

21. Haussler, D., Kearns, M., Seung, H. S. & Tishby, N. Rigorous learning curve bounds from statistical mechanics. *Mach. Learn.* **25**, 195–236 (1996).

22. Boucheron, S., Bousquet, O. & Lugosi, G. Theory of Classification: a Survey of Some Recent Advances. *ESAIM Probab. Stat.* **9**, 323–375 (2005).

23. Simon, H. U. General Bounds on the Number of Examples Needed for Learning Probabilistic Concepts. *J. Comput. System Sci.* **52**, 239–254 (1996).

24. Aslam, J. A. & Decatur, S. E. On the sample complexity of noise-tolerant learning. *Inf. Process. Lett.* **57**, 189–195 (1996).

25. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).

26. Ratner, A., De Sa, C., Wu, S., Selsam, D. & Ré, C. Data Programming: Creating Large Training Sets, Quickly. *Adv. Neural Inf. Process. Syst.* **29**, 3567–3575 (2016).

27. Bach, S. H. *et al.* Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. *Proc. ACM SIGMOD Int. Conf. Manag. Data* **2019**, 362–375 (2019).

28. Varma, P. & Ré, C. Snuba: Automating Weak Supervision to Label Training Data. *Proceedings VLDB Endowment* **12**, 223–236 (2018).
29. Bringer, E., Israeli, A., Shoham, Y., Ratner, A. & Ré, C. Osprey: Weak Supervision of Imbalanced Extraction Problems without Code. in *Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning* 1–11 (Association for Computing Machinery, 2019).
30. Zhang, J. *et al.* WRENCH: A Comprehensive Benchmark for Weak Supervision. *arXiv preprint arXiv* (2021).
31. Makar, M., Packer, B., Moldovan, D. & Blalock, D. Causally-motivated shortcut removal using auxiliary labels. *arXiv preprint arXiv* (2021).
32. Chen, D., Yu, Z. & Bowman, S. R. Learning with Noisy Labels by Targeted Relabeling. *arXiv preprint arXiv:2110.08355* (2021).
33. Wang, J. *et al.* Bridge the Gap between Supervised and Unsupervised Learning for Fine-Grained Classification. *arXiv preprint arXiv* (2022).
34. Learning from an Approximate Theory and Noisy Examples. https://www.aaai.org/Library/AAAI/1993/aaai93-070.php.
35. Decatur, S. E. & Gennaro, R. On learning from noisy and incomplete examples. *Proceedings of the eighth annual conference on Computational learning theory - COLT '95* (1995) doi:10.1145/225298.225341.
36. Townsend, L. & Wallace, C. Social media research: A guide to ethics. *University of Aberdeen* **1**, 16 (2016).

37. Chaffey, D. Global social media statistics research summary 2022. *Smart Insights* https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/ (2022).

38. Ahmed, W. Using social media data for research: An overview of tools. *Journal of Communication Technology* **1**, 78–93 (2018).

39. Baruah, T. D. & Others. Effectiveness of Social Media as a tool of communication and its potential for technology enabled connections: A micro-level study. *International journal of scientific and research publications* **2**, 1–10 (2012).

40. Facebook Terms and Conditions. https://www.facebook.com/legal/terms/previous.

41. Twitter: number of users worldwide 2020. *Statista* https://www.statista.com/statistics/303681/twitter-users-worldwide/.

42. Machine, W. The Internet Archive. *Searched for http://www. icann. org/icp/icp-1. htm* (2015).

43. Pfeffer, J., Mooseder, A., Hammer, L., Stritzel, O. & Garcia, D. This Sample seems to be good enough! Assessing Coverage and Temporal Reliability of Twitter's Academic API. *arXiv [cs.SI]* (2022).

44. Studies using Twitter until July 9. *Arxiv.org* https://arxiv.org/search/cs?query=twitter&searchtype=all&abstracts=show&order=-announced_date_first&size=50.

45. Burnap, P. & Williams, M. L. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making: Machine classification of cyber hate speech. *Policy Internet* **7**, 223–242 (2015).

46. Beatty, M. Classification Methods for Hate Speech Diffusion: Detecting the Spread of Hate Speech on Twitter. *Proceedings of the 6th World Congress on Electrical Engineering and Computer Systems and Science* (2020) doi:10.11159/cist20.105.

47. Kharde, V. A. & Sonawane, S. Sentiment Analysis of Twitter Data: A Survey of Techniques. *arXiv [cs.CL]* (2016).

48. Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. J. Sentiment analysis of twitter data. in *Proceedings of the workshop on language in social media (LSM 2011)* 30–38 (2011).

49. Klein, A. Z., Cai, H., Weissenbacher, D., Levine, L. D. & Gonzalez-Hernandez, G. A natural language processing pipeline to advance the use of Twitter data for digital epidemiology of adverse pregnancy outcomes. *Journal of Biomedical Informatics: X* 100076 (2020).

50. Klein, A. Z. & Gonzalez-Hernandez, G. An annotated data set for identifying women reporting adverse pregnancy outcomes on Twitter. *Data Brief* **32**, 106249 (2020).

51. Sarker, A. *et al.* Self-reported COVID-19 symptoms on Twitter: An analysis and a research resource. doi:10.1101/2020.04.16.20067421.

52. Singh, S. M. & Reddy, C. An Analysis of Self-reported Longcovid Symptoms on Twitter. doi:10.1101/2020.08.14.20175059.

53. Zou, L., Lam, N. S. N., Cai, H. & Qiang, Y. Mining Twitter Data for Improved Understanding of Disaster Resilience. *Ann. Assoc. Am. Geogr.* **108**, 1422–1441 (2018).

54. Earle, P. Earthquake Twitter. *Nat. Geosci.* **3**, 221–222 (2010).

55. Alam, F., Ofli, F., Imran, M. & Aupetit, M. A Twitter Tale of Three Hurricanes: Harvey, Irma, and Maria. *arXiv [cs.SI]* (2018).

56. Bruns, A. & Liang, Y. E. Tools and methods for capturing Twitter data during natural disasters. *First Monday* **17**, 1–8 (2012).

57. Jain, P., Zaher, Z. & Mazid, I. Opioids on Twitter: A Content Analysis of Conversations regarding Prescription Drugs on Social Media and Implications for Message Design. *J. Health Commun.* **25**, 74–81 (2020).

58. Jeri-Yabar, A. *et al.* Association between social media use (Twitter, Instagram, Facebook) and depressive symptoms: Are Twitter users at higher risk? *Int. J. Soc. Psychiatry* **65**, 14–19 (2019).

59. Lee, K., Agrawal, A. & Choudhary, A. Real-time disease surveillance using Twitter data: demonstration on flu and cancer. in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* 1474–1477 (Association for Computing Machinery, 2013).

60. Zhang, L., Hall, M. & Bastola, D. Utilizing Twitter data for analysis of chemotherapy. *Int. J. Med. Inform.* **120**, 92–100 (2018).

61. Coppersmith, G., Dredze, M., Harman, C. & Hollingshead, K. From ADHD to SAD: Analyzing the language of mental health on Twitter through self-reported diagnoses. in *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality* 1–10 (2015).

62. Coppersmith, G., Dredze, M. & Harman, C. Quantifying mental health signals in Twitter. in *Proceedings of the workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality* 51–60 (2014).

63. Reece, A. G. *et al.* Forecasting the onset and course of mental illness with Twitter data. *Sci. Rep.* **7**, 13006 (2017).

64. CDCTobaccoFree. Tips from former smokers ®. https://www.cdc.gov/tobacco/campaign/tips/ (2021).

65. truth. https://www.thetruth.com/.

66. Chaudhry, A., Glodé, L. M., Gillman, M. & Miller, R. S. Trends in twitter use by physicians at the american society of clinical oncology annual meeting, 2010 and 2011. *J. Oncol. Pract.* **8**, 173–178 (2012).

67. *tweepy*. (Github).

68. *twarc*. (Github).

69. Hale, S. A. Global connectivity and multilinguals in the Twitter network. *Proceedings of the SIGCHI conference on human* (2014).

70. *python-twitter: A simple Python wrapper for Twitter API v2 ✨🍰✨*. (Github).

71. Ahmed, S. *Real-Time-Twitter-Stream: Stream live tweets across the world in real time*. (Github).

72. *twitter-streaming-api: Easily work with the Twitter Streaming API*. (Github).

73. Tekumalla, R. & Banda, J. M. Social Media Mining Toolkit (SMMT). *Genomics Inform.* **18**, e16 (2020).

74. Tekumalla, R. & Banda, J. M. An Enhanced Approach to Identify and Extract Medication Mentions in Tweets via Weak Supervision. in *Proceedings of the BioCreative VII Challenge Evaluation Workshop* (2021).

75. Littman, J., Wrubel, L. & Kerchner, D. 2016 United States Presidential Election Tweet Ids. (2016) doi:10.7910/DVN/PDI7IN.

76. Summers, E. Eclipse tweet IDs. doi:https://archive.org/details/eclipse-tweets.csv.

77. Diaz, F., Gamon, M., Hofman, J. M., Kıcıman, E. & Rothschild, D. Online and Social Media Data As an Imperfect Continuous Panel Survey. *PLoS One* **11**, e0145406 (2016).
78. Olteanu, A., Varol, O. & Kiciman, E. Distilling the Outcomes of Personal Experiences. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (2017) doi:10.1145/2998181.2998353.
79. Freelon, D. Beyond the hashtags twitter data. http://dfreelon.org/2017/01/03/beyond-the-hashtags-twitter-data/.
80. Littman, J. & Wrubel, L. Climate change tweets ids. (2019) doi:10.7910/DVN/5QCCUU.
81. Summers, E. Trump tweet IDs. doi:https://archive.org/details/trump-tweet-ids.
82. Littman, J. Healthcare Tweet Ids. (2019) doi:10.7910/DVN/IHWGQT.
83. Ruest, N. #WomensMarch tweets January 12-28, 2017. (2017) doi:10.5683/SP/ZEL1Q6.
84. Littman, J., Kerchner, D. & Wrubel, L. U.S. government tweet ids. (2017) doi:10.7910/DVN/2N3HHD.
85. Littman, J. & Kerchner, D. End of term 2016 U.s. government twitter archive. (2017) doi:10.7910/DVN/TQBLWZ.
86. Bergis Jules, Y. O. A. ed S. Nipsey Hussle tweets.
87. Littman, J. Winter Olympics 2018 tweet ids. (2018) doi:10.7910/DVN/YMJPFC.
88. Phillips, M. E. Dallas police shooting Twitter dataset. in (2016). doi:https://digital.library.unt.edu/ark:/67531/metadc991469/.
89. Littman, J., Wrubel, L., Kerchner, D. & Bromberg Gaber, Y. News outlet tweet ids. (2017) doi:10.7910/DVN/2FIFLH.
90. Littman, J. Charlottesville Tweet Ids. (2018) doi:10.7910/DVN/DVLJTO.
91. Zubiaga, A. Twitter event datasets (2012-2016). (2017) doi:10.6084/m9.figshare.5100460.v1.

92. Littman, J. Immigration and travel ban tweet ids. (2018) doi:10.7910/DVN/5CFLLJ.

93. Poghosyan, G. Insight4news Irish news related hashtagged tweet collection. (2019) doi:10.6084/m9.figshare.7932422.v4.

94. Summers, E. BlackLivesMatter tweets 2016. in doi:https://archive.org/details/blacklivesmatter-tweets-2016.txt.

95. Ruest, N. Tweets to Donald Trump (@realDonaldTrump). (2017) doi:10.5683/SP/8BAVQM.

96. Littman, J. Hurricanes Harvey and Irma Tweet ids. (2017) doi:10.7910/DVN/QRKIBW.

97. Wrubel, L. Hurricane Florence. in (Harvard Dataverse, 2019). doi:10.7910/DVN/GSIUXQ.

98. Phillips, M. E. Hurricane Harvey Twitter Dataset. in (2018). doi:ark:/67531/metadc993940.

99. Littman, J. 115th U.S. Congress Tweet Ids. (2017) doi:10.7910/DVN/UIVHQR.

100. Chen, E., Deb, A. & Ferrara, E. 2020 US Presidential Election Tweet IDs Release 1.3. (2021) doi:10.7910/DVN/QYSSVA.

101. Phillips, M. E. Hurricane Florence Twitter Dataset. in (2018). doi:https://digital.library.unt.edu/ark:/67531/metadc1259406/.

102. Summers, E. Puerto Rico tweets. (2017) doi:https://archive.org/details/puertorico-tweets.

103. Zubiaga, A. & Ji, H. Tweet, but verify: epistemic study of information verification on Twitter. *Social Network Analysis and Mining* **4**, 163 (2014).

104. Benjamin Rachunok, Roshanak Nateghi & Douglas McWherter. Hurricane Dorian Tweet IDs. (2019) doi:/10.4231/CPGM-E419.

105. Phillips, M. E. Hurricane Dorian Twitter Dataset. in (2019). doi:https://digital.library.unt.edu/ark:/67531/metadc1706014.

106. Wrubel, L., Littman, J. & Kerchner, D. 2018 U.S. Congressional Election Tweet Ids. (2018) doi:10.7910/DVN/AEZPLU.

107. Paul, M. J. & Dredze, M. Discovering health topics in social media using topic models. *PLoS One* **9**, e103408 (2014).

108. Liu, J., Singhal, T., Blessing, L. T. M., Wood, K. L. & Lim, K. H. EPIC30M: An Epidemics Corpus of Over 30 Million Relevant Tweets. in *2020 IEEE International Conference on Big Data (Big Data)* 1206–1215 (2020).

109. Mitchell, T. M. & Others. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill* **45**, 870–877 (1997).

110. Russell Stuart, J. Artificial Intelligence/Stuart J. Russell, Peter Norvig. *A Modern Approach (Third ed.). Prentice Hall* 649 (2010).

111. Chauhan, N. K. & Singh, K. A Review on Conventional Machine Learning vs Deep Learning. in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)* 347–352 (2018).

112. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

113. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995).

114. Hosmer, D. W., Jr, Lemeshow, S. & Sturdivant, R. X. *Applied Logistic Regression*. (John Wiley & Sons, 2013).

115. Murphy. Naive bayes classifiers. *Univ. B. C. Law Rev.*

116. Webb, G. I., Keogh, E. & Miikkulainen, R. Naïve Bayes. *Encyclopedia of machine learning* **15**, 713–714 (2010).

117. Quinlan, J. R. Induction of decision trees. *Mach. Learn.* **1**, 81–106 (1986).

118. Pavlov, Y. L. *Random Forests*. (Walter de Gruyter GmbH & Co KG, 2019).

119. Deng, L. & Yu, D. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* **7**, 197–387 (2014).

120. Kowsari, K. *et al.* Text Classification Algorithms: A Survey. *Information* **10**, 150 (2019).

121. Hubel, D. H. & Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *J. Physiol.* **160**, 106–154 (1962).

122. Hu, B., Lu, Z., Li, H. & Chen, Q. Convolutional Neural Network Architectures for Matching Natural Language Sentences. in *Advances in Neural Information Processing Systems 27* (eds. Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. & Weinberger, K. Q.) 2042–2050 (Curran Associates, Inc., 2014).

123. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).

124. Lawrence, S., Giles, C. L., Tsoi, A. C. & Back, A. D. Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Netw.* **8**, 98–113 (1997).

125. Li, Q. *et al.* Medical image classification with convolutional neural network. in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)* 844–848 (2014).

126. Li, C., Zhan, G. & Li, Z. News Text Classification Based on Improved Bi-LSTM-CNN. in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)* 890–893 (2018).

127. Wang, S., Huang, M., Deng, Z. & Others. Densely connected CNN with multi-scale feature attention for text classification. in *IJCAI* 4468–4474 (2018).

128. Du, J., Gui, L., Xu, R. & He, Y. A Convolutional Attention Model for Text Classification. in *Natural Language Processing and Chinese Computing* 183–195 (Springer International Publishing, 2018).

129. Li, Y. & Yang, T. Word Embedding for Understanding Natural Language: A Survey. in *Guide to Big Data Applications* (ed. Srinivasan, S.) 83–104 (Springer International Publishing, 2018).

130. Lavertu, A. & Altman, R. B. RedMed: Extending drug lexicons for social media applications. *J. Biomed. Inform.* **99**, 103307 (2019).

131. Pennington, J., Socher, R. & Manning, C. D. Glove: Global vectors for word representation. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* 1532–1543 (2014).

132. Godin, F. Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing. (PhD thesis, PhD Thesis, Ghent University, Belgium, 2019. 35, 2019).

133. Vaswani, A. *et al.* Attention is all you need. in *Advances in neural information processing systems* 5998–6008 (2017).

134. Rajapakse, T. *simpletransformers*. (Github, 2019). doi:https://github.com/ThilinaRajapakse/simpletransformers.

135. Wolf, T. *et al.* HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv e-prints* arXiv:1910.03771 (2019).

136. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2623–2631 (Association for Computing Machinery, 2019).

137. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv [cs.CL]* (2018).

138. Zhu, Y. *et al.* Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. in *Proceedings of the IEEE international conference on computer vision* 19–27 (2015).

139. Lee, J. *et al.* BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **36**, 1234–1240 (2020).

140. Liu, Y. *et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv [cs.CL]* (2019).

141. Nagel, S. Cc-news. *http: //web.archive.org/save/http: //commoncrawl.org/2016/10/newsdataset-available* http:

142. Gokaslan, A. & Cohen, V. Openwebtext corpus. (2019).

143. Trinh, T. H. & Le, Q. V. A Simple Method for Commonsense Reasoning. *arXiv [cs.AI]* (2018).

144. Nguyen, G. Disaster_tweet_bert. *Hugging Face* https://huggingface.co/garynguyen1174/disaster_tweet_bert.

145. Nguyen, D. Q., Vu, T. & Nguyen, A. T. BERTweet: A pre-trained language model for English Tweets. *arXiv [cs.CL]* (2020).

146. O’Connor, K. *et al.* Pharmacovigilance on twitter? Mining tweets for adverse drug reactions. *AMIA Annu. Symp. Proc.* **2014**, 924–933 (2014).

147. Leaman, R. *et al.* Towards Internet-Age Pharmacovigilance: Extracting Adverse Drug Reactions from User Posts in Health-Related Social Networks. (2010).

148. DailyStrength: Online Support Groups and Forums. https://www.dailystrength.org/.

149. Sarker, A. & Gonzalez, G. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *J. Biomed. Inform.* **53**, 196–207 (2015).

150. Sarker, A. & Gonzalez, G. A corpus for mining drug-related knowledge from Twitter chatter: Language models and their utilities. *Data Brief* **10**, 122–131 (2017).

151. Health Language Processing Lab @ Penn IBI. https://healthlanguageprocessing.org/.

152. Lindquist, M. The need for definitions in pharmacovigilance. *Drug Saf.* **30**, 825–830 (2007).

153. National Library of Medicine (US). UMLS® Reference Manual [Internet] : 2, Metathesaurus. *National Library of Medicine (US)* https://www.ncbi.nlm.nih.gov/books/NBK9684/ (2009).

154. National Library of Medicine. RxNorm [Internet]. *National Library of Medicine (US)* http://www.nlm.nih.gov/research/umls/rxnorm/ (2008).

155. Tekumalla, R., Asl, J. R. & Banda, J. M. Mining Archive. org's Twitter Stream Grab for Pharmacovigilance Research Gold. in *Proceedings of the International AAAI Conference on Web and Social Media* vol. 14 909–917 (2020).

156. Aiello, L. M. *et al.* Sensing Trending Topics in Twitter. *IEEE Trans. Multimedia* **15**, 1268–1282 (2013).

157. Klein, A., Sarker, A., Rouhizadeh, M., O'Connor, K. & Gonzalez, G. Detecting personal medication intake in Twitter: an annotated corpus and baseline classification system. in *BioNLP 2017* 136–142 (2017).

158. Weissenbacher, D., Rawal, S., Magge, A. & Gonzalez-Hernandez, G. Addressing Extreme Imbalance for Detecting Medications Mentioned in Twitter User Timelines. doi:10.1101/2021.02.09.21251453.

159. Hughes, A. L. & Palen, L. Twitter adoption and use in mass convergence and emergency events. *Int. J. Emergency Manage.* **6**, 248–260 (2009).

160. Imran, M., Mitra, P. & Castillo, C. Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. *arXiv [cs.CL]* (2016).

161. Imran, M., Elbassuoni, S., Castillo, C., Diaz, F. & Meier, P. Practical extraction of disaster-relevant information from social media. in *Proceedings of the 22nd International Conference on World Wide Web* 1021–1024 (Association for Computing Machinery, 2013).

162. Truong, B., Caragea, C., Squicciarini, A. & Tapia, A. H. Identifying valuable information from twitter during natural disasters. *Proc. Am. Soc. Inf. Sci. Technol.* **51**, 1–4 (2014).

163. Ofli, F. *et al.* Combining Human Computing and Machine Learning to Make Sense of Big (Aerial) Data for Disaster Response. *Big Data* **4**, 47–59 (2016).

164. Resch, B., Usländer, F. & Havas, C. Combining machine-learning topic models and spatiotemporal analysis of social media data for disaster footprint and damage assessment. *Cartogr. Geogr. Inf. Sci.* **45**, 362–376 (2018).

165. Burel, G. & Alani, H. Crisis Event Extraction Service (CREES) - Automatic Detection and Classification of Crisis-related Content on Social Media. in (2018).

166. Bontcheva, K. *et al.* Twitie: An open-source information extraction pipeline for microblog text. in *Proceedings of the international conference recent advances in natural language processing RANLP 2013* 83–90 (2013).

167. Nguyen, D. *et al.* Robust Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks. *ICWSM* **11**, 632–635 (2017).

168. Madichetty, S. & M, Sridevi. Improved Classification of Crisis-Related Data on Twitter using Contextual Representations. *Procedia Comput. Sci.* **167**, 962–968 (2020).

169. Earle, P. S., Bowden, D. C. & Guy, M. Twitter earthquake detection: earthquake monitoring in a social world. *Ann. Geophys.* **54**, (2011).

170. Tekumalla, R. & Banda, J. M. TweetDIS: A large Twitter dataset for natural disasters built using weak supervision. *arXiv [cs.CL]* (2022).

171. CrowdFlower. CrowdFlower. *CrowdFlower* https://visit.figure-eight.com/People-Powered-Data-Enrichment_T.

172. Brownstein, J. S., Freifeld, C. C., Reis, B. Y. & Mandl, K. D. Surveillance Sans Frontières: Internet-Based Emerging Infectious Disease Intelligence and the HealthMap Project. *PLoS Medicine* vol. 5 e151 (2008).

173. Mykhalovskiy, E. & Weir, L. The Global Public Health Intelligence Network and Early Warning Outbreak Detection. *Canadian Journal of Public Health* vol. 97 42–44 (2006).

174. Johnson, H. A. *et al.* Analysis of Web access logs for surveillance of influenza. *Stud. Health Technol. Inform.* **107**, 1202–1206 (2004).

175. Ginsberg, J. *et al.* Detecting influenza epidemics using search engine query data. *Nature* **457**, 1012–1014 (2009).

176. Banda, J. M. *et al.* A Large-Scale COVID-19 Twitter Chatter Dataset for Open Scientific Research—An International Collaboration. *Epidemiologia* vol. 2 315–324 (2021).

177. Pawelek, K. A., Oeldorf-Hirsch, A. & Rong, L. Modeling the impact of twitter on influenza epidemics. *Math. Biosci. Eng.* **11**, 1337–1356 (2014).

178. Khatua, A., Khatua, A. & Cambria, E. A tale of two epidemics: Contextual Word2Vec for classifying twitter streams during outbreaks. *Inf. Process. Manag.* **56**, 247–257 (2019).

179. Lampos, V., De Bie, T. & Cristianini, N. Flu Detector - Tracking Epidemics on Twitter. in *Machine Learning and Knowledge Discovery in Databases* 599–602 (Springer Berlin Heidelberg, 2010).

180. Aramaki, E., Maskawa, S. & Morita, M. Twitter catches the flu: detecting influenza epidemics using Twitter. in *Proceedings of the 2011 Conference on empirical methods in natural language processing* 1568–1576 (2011).

181. Chen, E., Lerman, K. & Ferrara, E. Covid-19: The first public coronavirus twitter dataset. (2020).

182. Lamsal, R. Design and analysis of a large-scale COVID-19 tweets dataset. *Appl Intell (Dordr)* **51**, 2790–2804 (2021).

183. Culotta, A. Towards detecting influenza epidemics by analyzing Twitter messages. in *Proceedings of the First Workshop on Social Media Analytics* 115–122 (Association for Computing Machinery, 2010).

184. Molaei, S., Khansari, M., Veisi, H. & Salehi, M. Predicting the spread of influenza epidemics by analyzing twitter messages. *Health Technol.* **9**, 517–532 (2019).

185. Missier, P. *et al.* Tracking Dengue Epidemics Using Twitter Content Classification and Topic Modelling. in *Current Trends in Web Engineering* 80–92 (Springer International Publishing, 2016).

186. Szomszor, M., Kostkova, P. & Louis, C. S. Twitter Informatics: Tracking and Understanding Public Reaction during the 2009 Swine Flu Pandemic. in *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* vol. 1 320–323 (2011).

187. Szomszor, M., Kostkova, P. & de Quincey, E. #swineflu: Twitter predicts swine flu outbreak in 2009. in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* 18–26 (Springer Berlin Heidelberg, 2011).

188. Chew, C. & Eysenbach, G. Pandemics in the age of Twitter: content analysis of Tweets during the 2009 H1N1 outbreak. *PLoS One* **5**, e14118 (2010).

189. Stevens, R. *et al.* Association Between HIV-Related Tweets and HIV Incidence in the United States: Infodemiology Study. *J. Med. Internet Res.* **22**, e17196 (2020).

190. Bannon, J. A. *et al.* From virus to viral: Content analysis of HIV-related Twitter messages among young men in the U.S. (2022) doi:10.31235/osf.io/jtyr3.

191. Tekumalla, R. & Banda, J. M. Identifying epidemic related Tweets using noisy learning. in *Proceedings of LatinX in Natural Language Processing Research Workshop at NAACL 2022*.

192. Paul, M. J. & Dredze, M. A model for mining public health topics from Twitter. *Health* **11**, 1 (2012).

193. Alkouz, B., Al Aghbari, Z., Al-Garadi, M. A. & Sarker, A. Deepluenza: Deep learning for influenza detection from Twitter. *Expert Syst. Appl.* **198**, 116845 (2022).

194. Wikipedia contributors. Google Flu Trends. *Wikipedia, The Free Encyclopedia* https://en.wikipedia.org/w/index.php?title=Google_Flu_Trends&oldid=1091792681 (2022).

195. West, J. H. *et al.* Temporal variability of problem drinking on Twitter. *Open J. Prev. Med.* **02**, 43–48 (2012).

196. Heaivilin, N., Gerbert, B., Page, J. E. & Gibbs, J. L. Public health surveillance of dental pain via Twitter. *J. Dent. Res.* **90**, 1047–1051 (2011).

197. Love, B., Himelboim, I., Holton, A. & Stewart, K. Twitter as a source of vaccination information: content drivers and what they are saying. *Am. J. Infect. Control* **41**, 568–570 (2013).

198. Thackeray, R., Burton, S. H., Giraud-Carrier, C., Rollins, S. & Draper, C. R. Using Twitter for breast cancer prevention: an analysis of breast cancer awareness month. *BMC Cancer* **13**, 508 (2013).

199. McClellan, Ali, Mutter & Kroutil. Using social media to monitor mental health discussions− evidence from Twitter. *J. Asiat. Soc. Bangladesh Humanit.*

200. Harris, J. K., Moreland-Russell, S., Tabak, R. G., Ruhr, L. R. & Maier, R. C. Communication about childhood obesity on Twitter. *Am. J. Public Health* **104**, e62–9 (2014).

201. Sinnenberg, L. *et al.* Twitter as a Tool for Health Research: A Systematic Review. *Am. J. Public Health* **107**, e1–e8 (2017).

202. Prieto, V. M., Matos, S., Álvarez, M., Cacheda, F. & Oliveira, J. L. Twitter: a good place to detect health conditions. *PLoS One* **9**, e86191 (2014).

203. Klein, A. Z., O'Connor, K., Levine, L. D. & Gonzalez-Hernandez, G. Using Twitter data for cohort studies of drug safety in pregnancy: A proof-of-concept with beta-blockers. *bioRxiv* (2022) doi:10.1101/2022.02.23.22271408.

204. Klein, A. Z., O'Connor, K. & Gonzalez-Hernandez, G. Toward Using Twitter Data to Monitor COVID-19 Vaccine Safety in Pregnancy: Proof-of-Concept Study of Cohort Identification (Preprint). doi:10.2196/preprints.33792.

205. De Choudhury, Gamon & Counts. Predicting depression via social media. *weblogs and social media*.

206. De Choudhury, M., Kiciman, E., Dredze, M., Coppersmith, G. & Kumar, M. Discovering Shifts to Suicidal Ideation from Mental Health Content in Social Media. *Proc SIGCHI Conf Hum Factor Comput Syst* **2016**, 2098–2110 (2016).

207. Balani, S. & De Choudhury, M. Detecting and Characterizing Mental Health Related Self-Disclosure in Social Media. in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* 1373–1378 (Association for Computing Machinery, 2015).

208. Ramya Tekumalla, Zia Baig, Michelle Pan, Luis Alberto Robles Hernandez, Michael Wang, Juan Banda. Characterizing Anti-Asian Rhetoric During The COVID-19 Pandemic: A Sentiment Analysis Case Study on Twitter. in *Workshop Proceedings of the 16th International AAAI Conference on Web and Social Media* (2022). doi:10.36190/2022.81.

APPENDICES

Appendix A: Identifying drug mentions from Twitter

The following are the plots for Precision and Recall for all the training ratios for all the models.

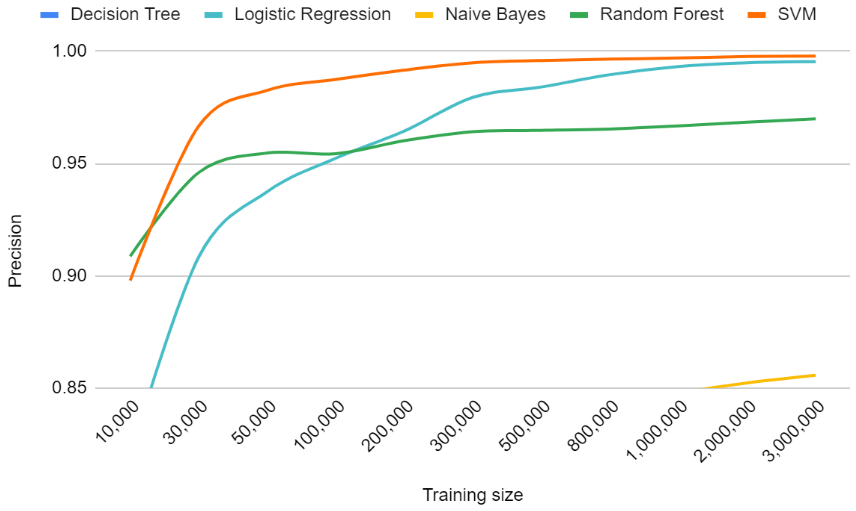


Figure 52. Classical models mean Precision for 1:1 ratio

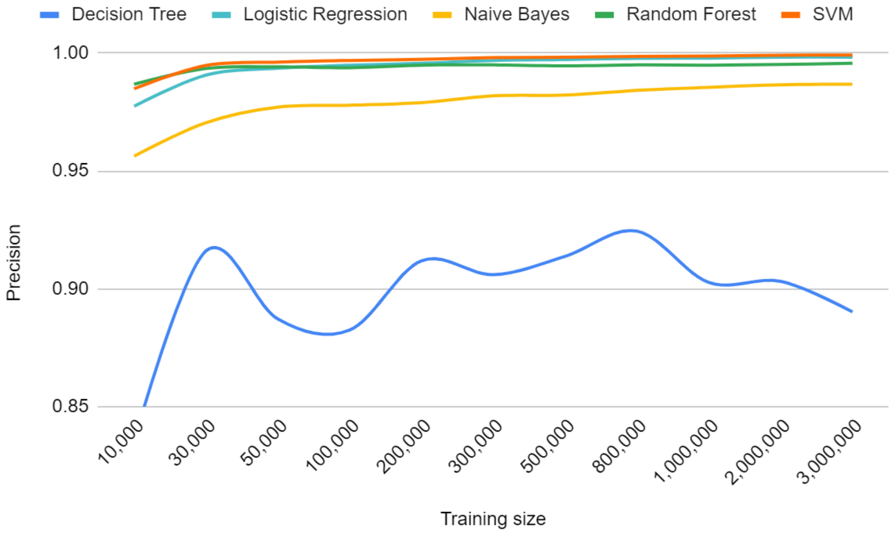


Figure 53. Classical models mean Precision for 1:5 ratio

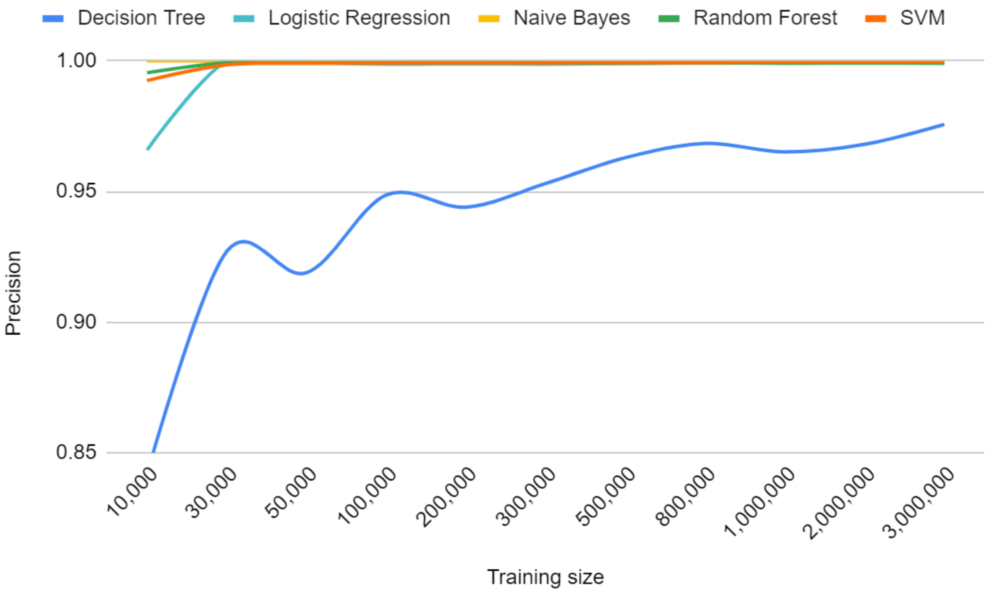


Figure 54. Classical models mean Precision for 1:15 ratio

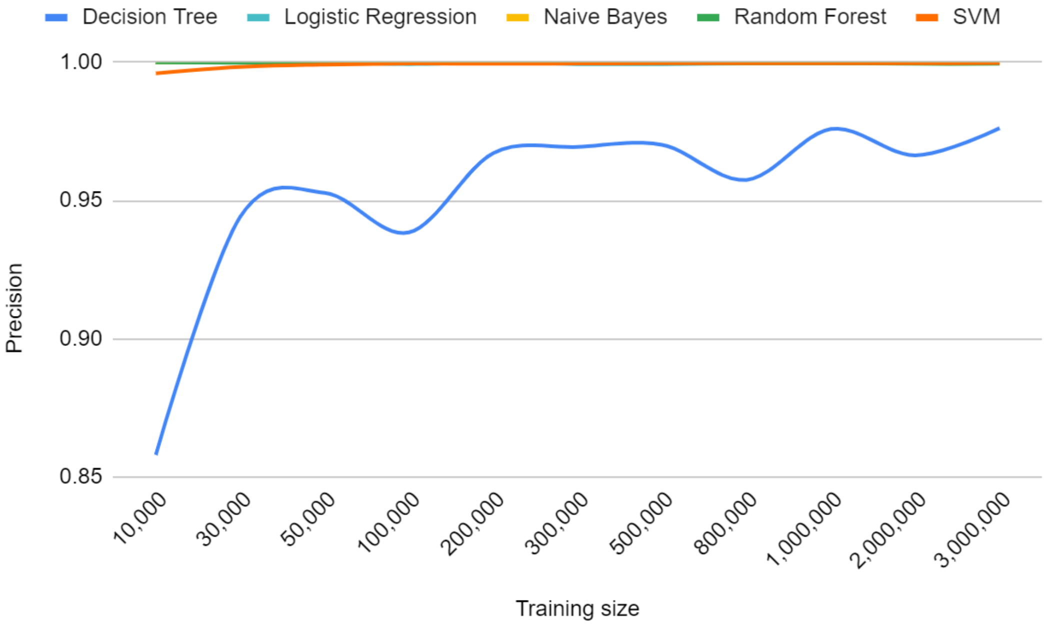


Figure 55. Classical models mean Precision for 1:25 ratio

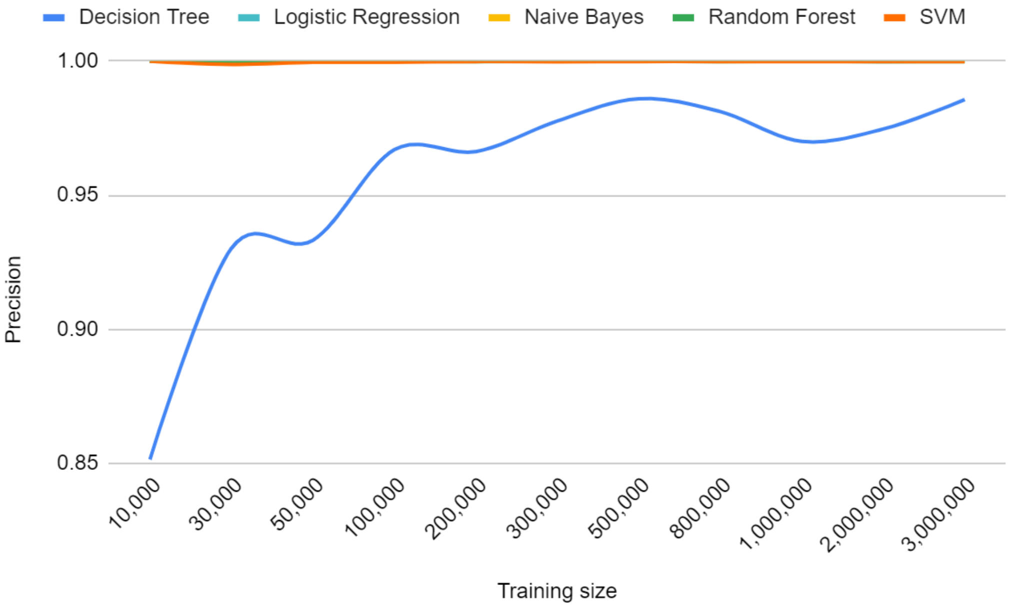


Figure 56. Classical models mean Precision for 1:50 ratio

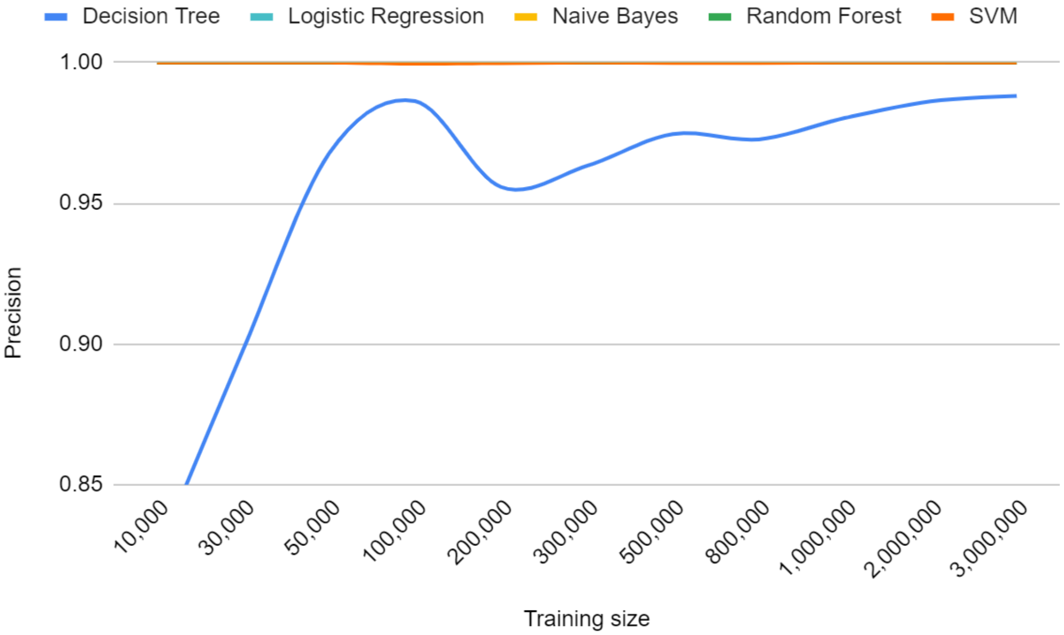


Figure 57. Classical models mean Precision for 1:100 ratio

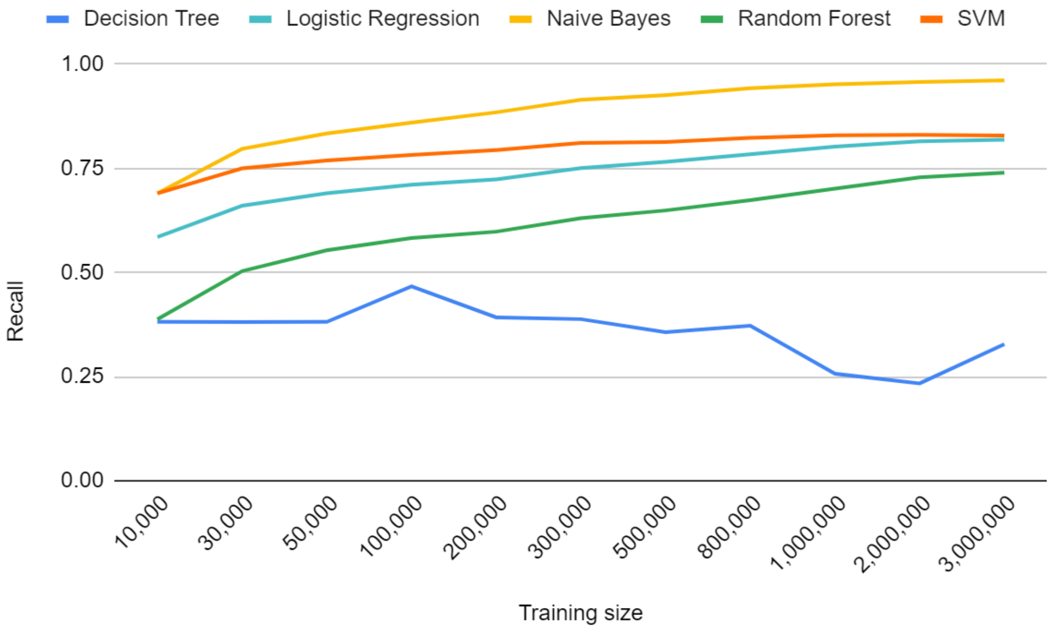


Figure 58. Classical models mean Recall for 1:1 ratio

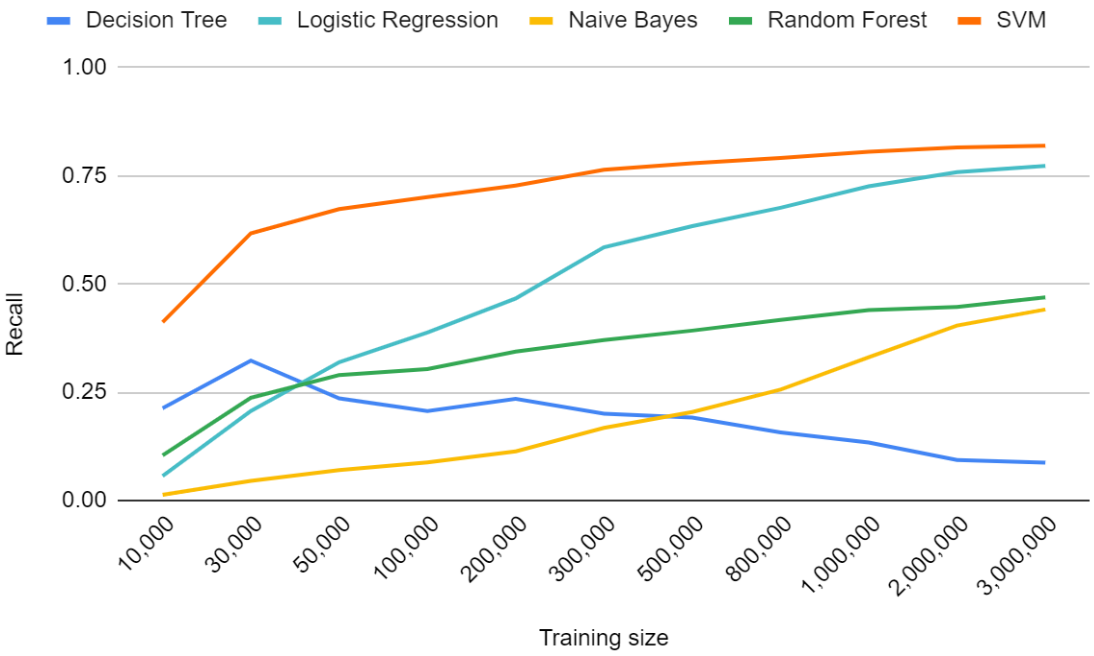


Figure 59. Classical models mean Recall for 1:5 ratio

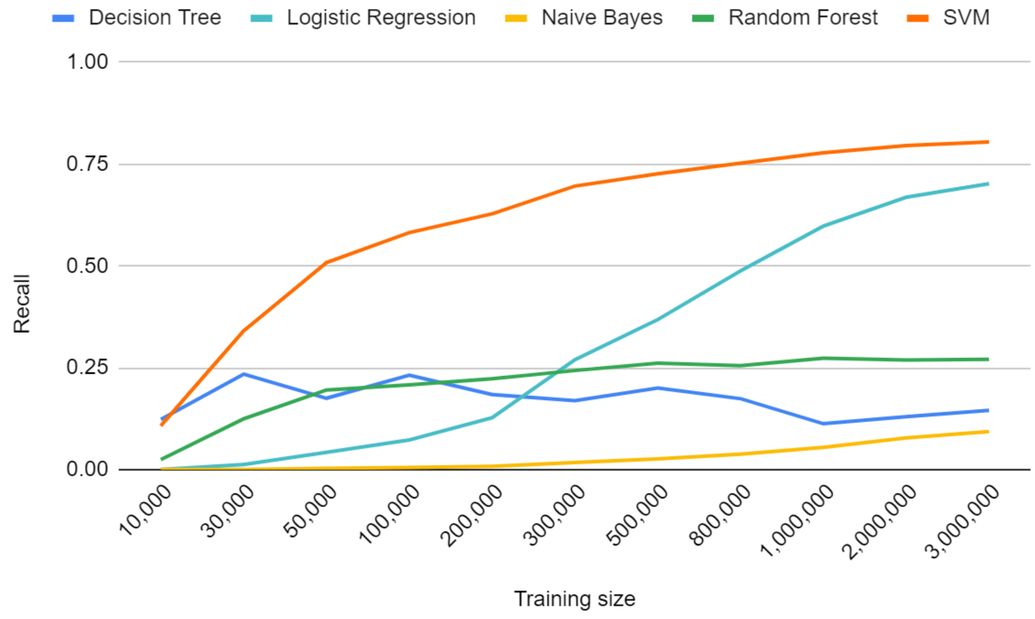


Figure 60. Classical models mean Recall for 1:15 ratio

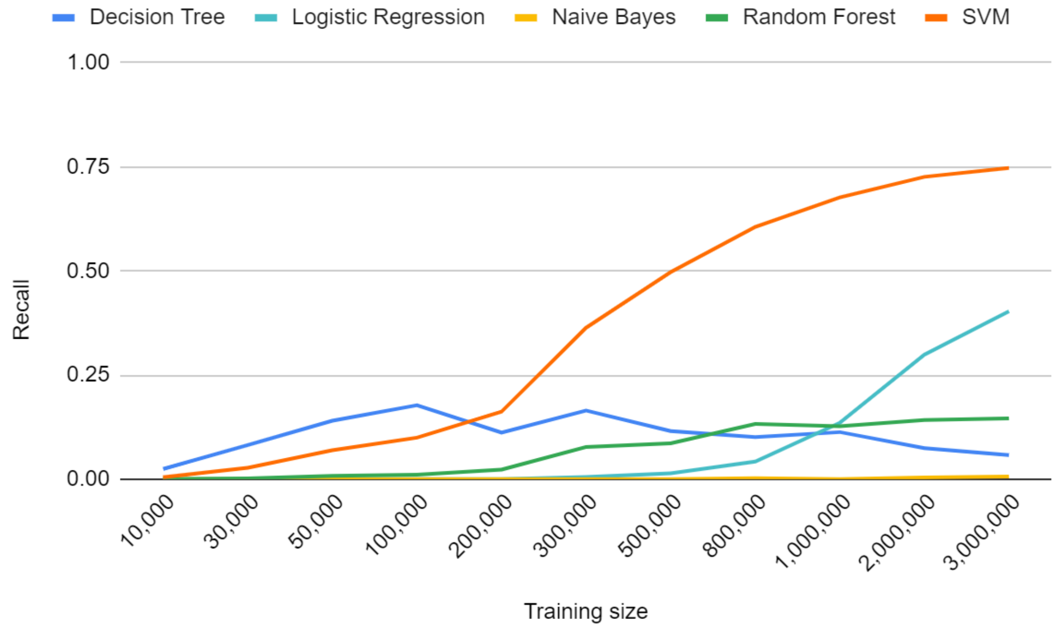


Figure 61. Classical models mean Recall for 1:100 ratio

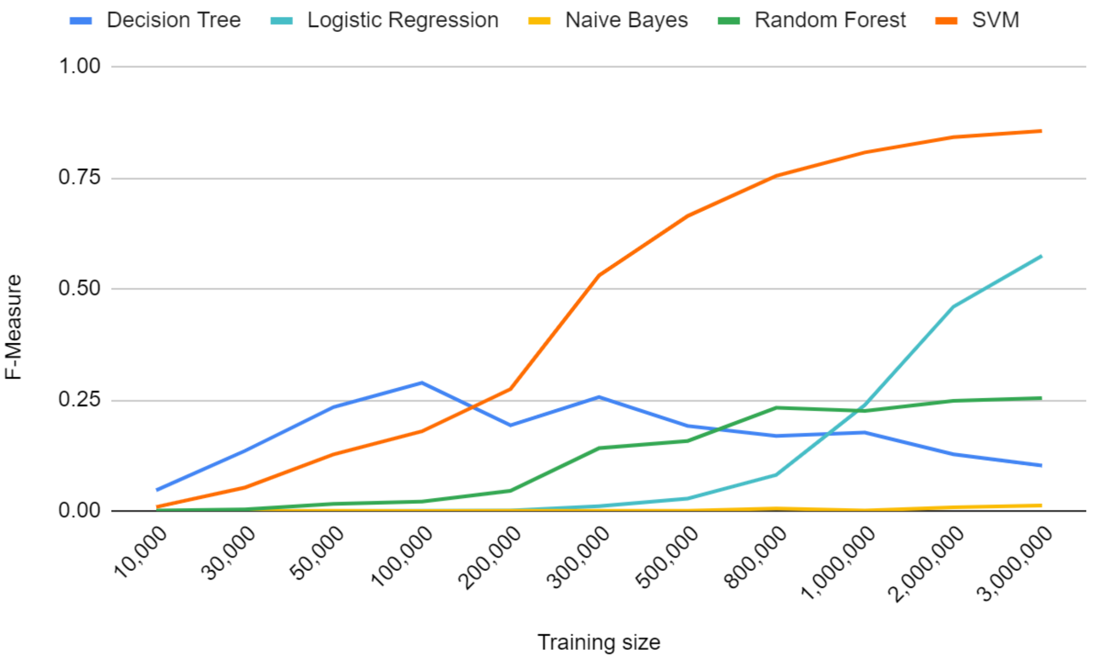


Figure 62. Classical models mean F-Measure for 1:100 ratio

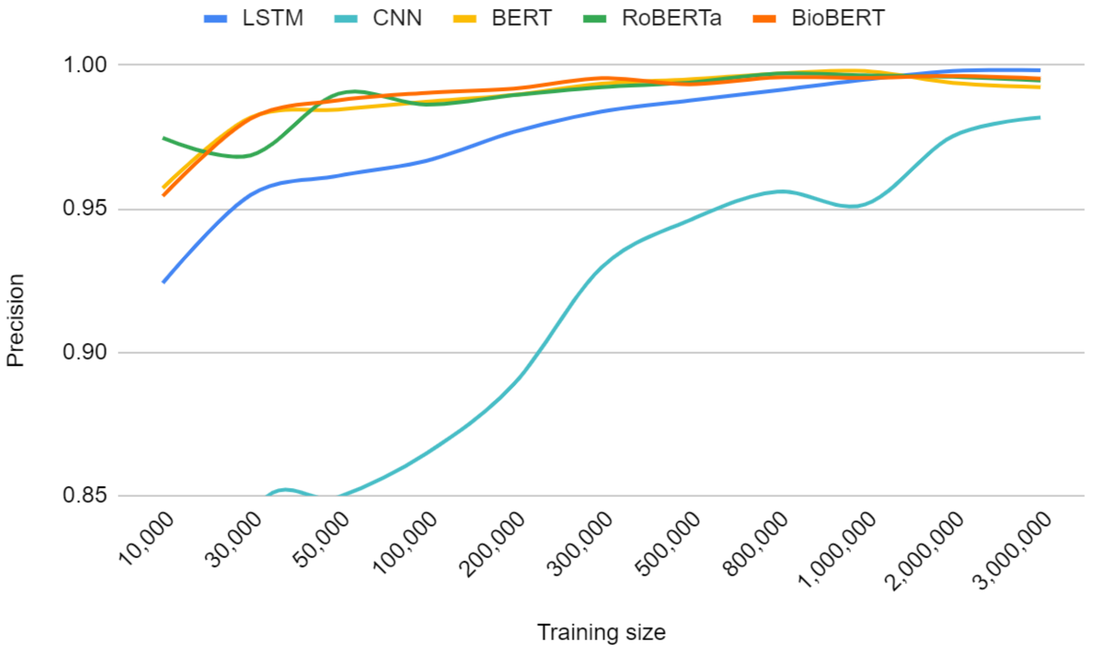


Figure 63. Deep learning models mean Precision for 1:1 ratio

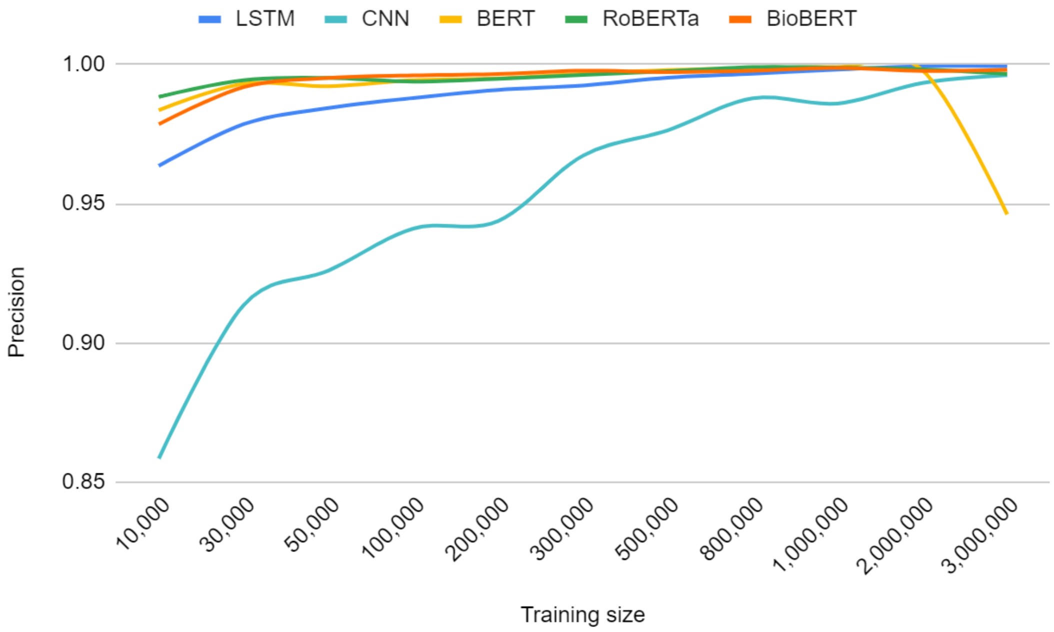


Figure 64. Deep learning models mean Precision for 1:5 ratio

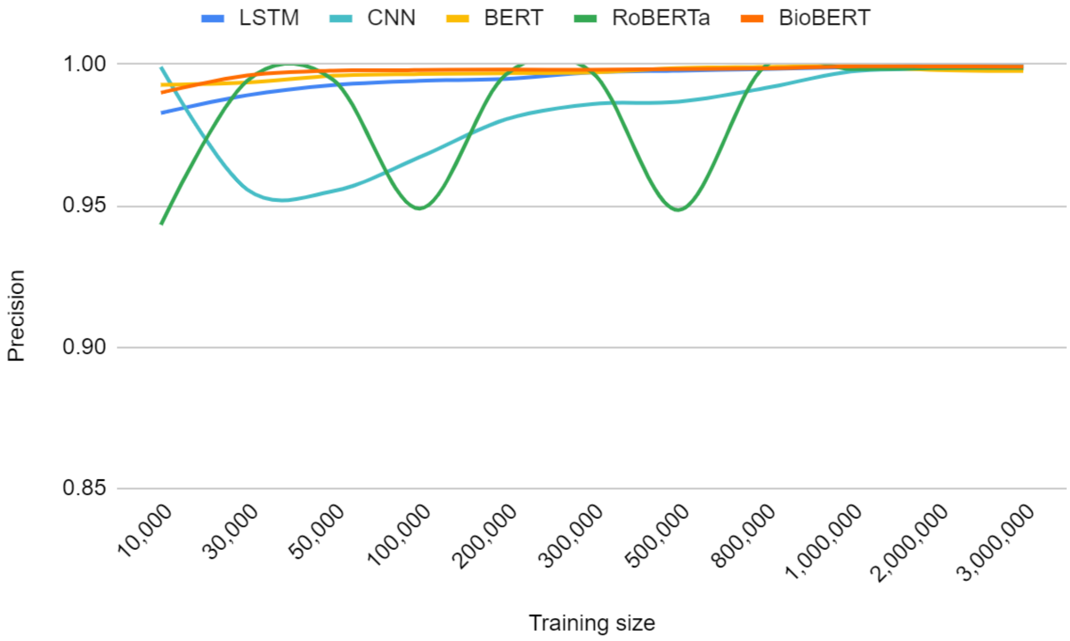


Figure 65. Deep learning models mean Precision for 1:15 ratio

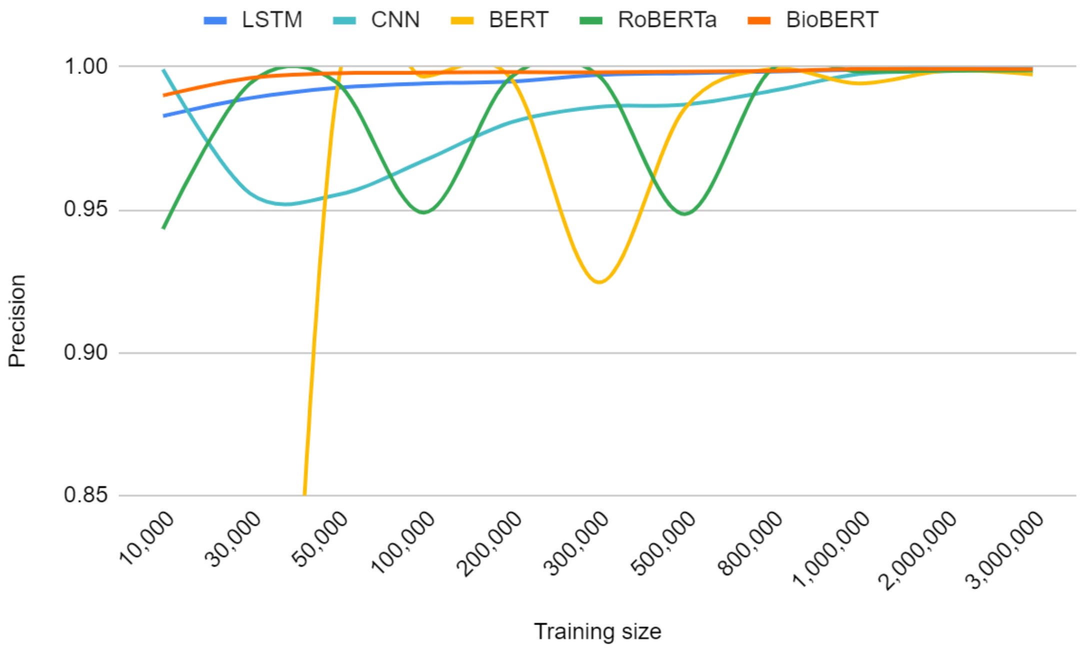


Figure 66. Deep learning models mean Precision for 1:25 ratio

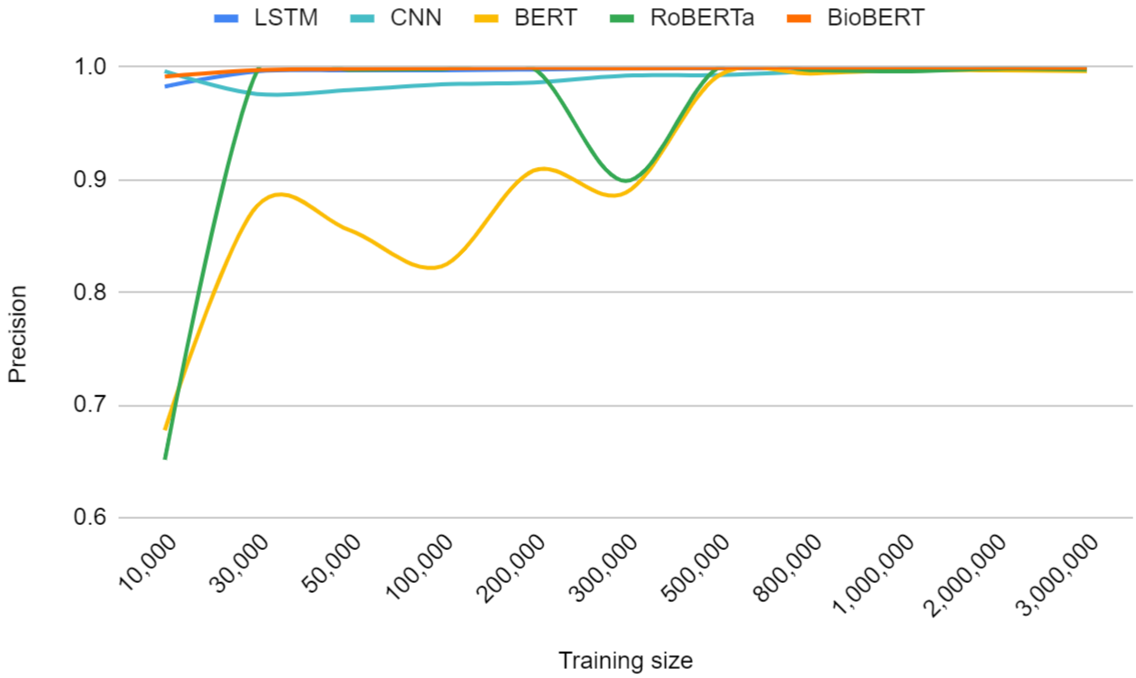


Figure 67. Deep learning models mean Precision for 1:50 ratio

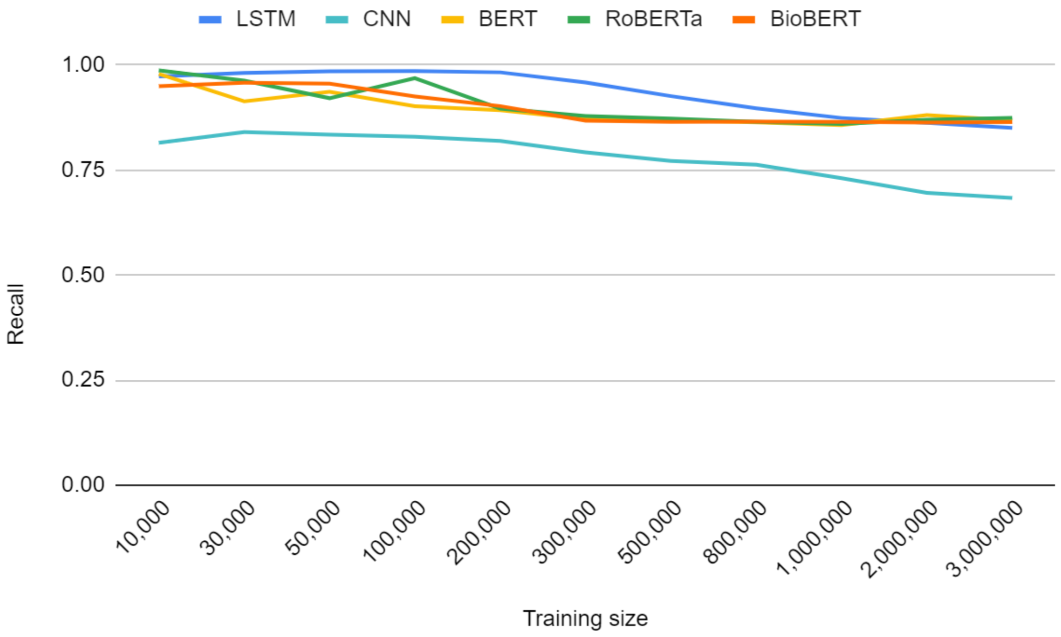


Figure 68. Deep learning models mean Recall for 1:1 ratio

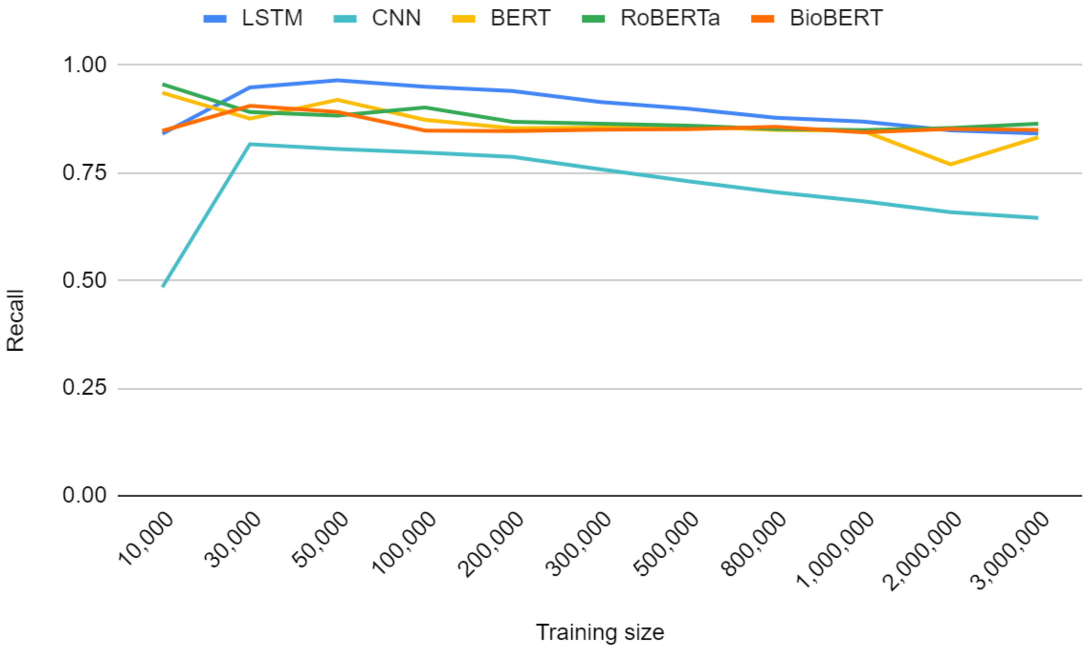


Figure 69. Deep learning models mean Recall for 1:5 ratio

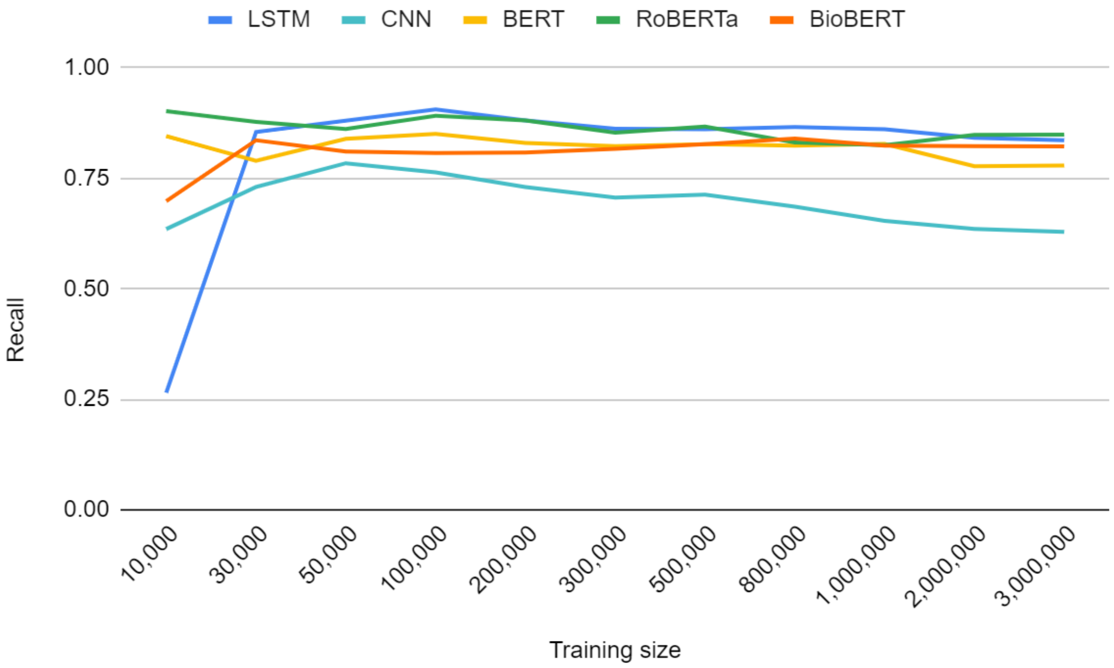


Figure 70. Deep learning models mean Recall for 1:15 ratio

Appendix B: Characterizing different types of natural disasters: hurricanes, earthquakes and floods

The following are the plots for Precision and Recall for all the training ratios for all the models.

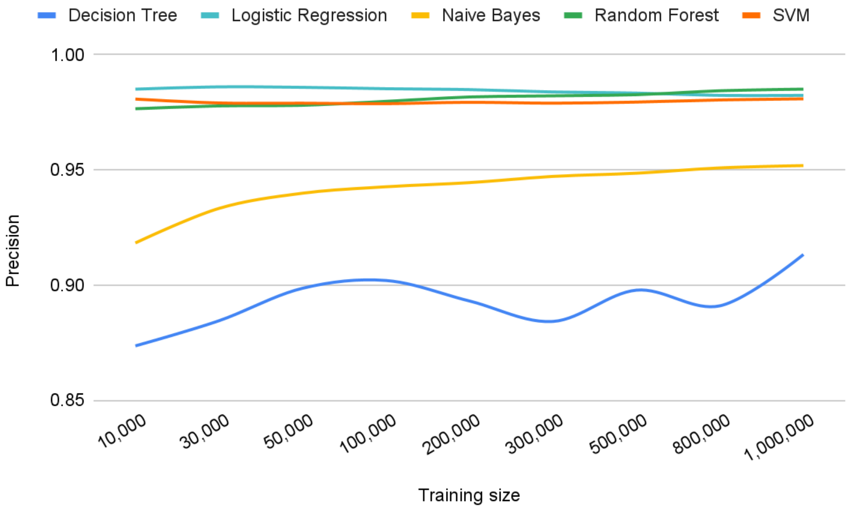


Figure 71. Mean of Precision for 1:1 ratio classical models

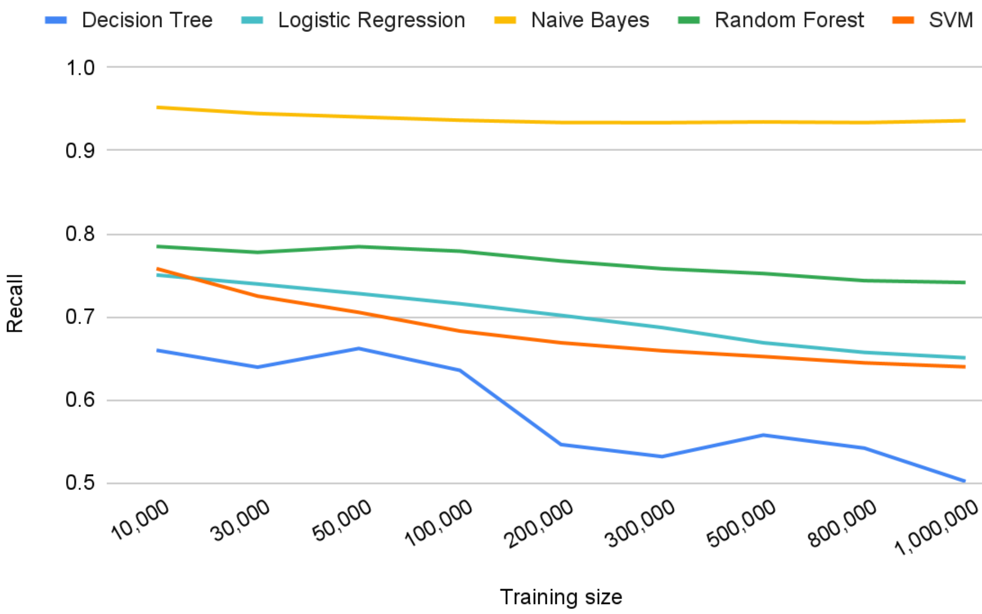


Figure 72. Mean of Recall for 1:1 ratio classical models

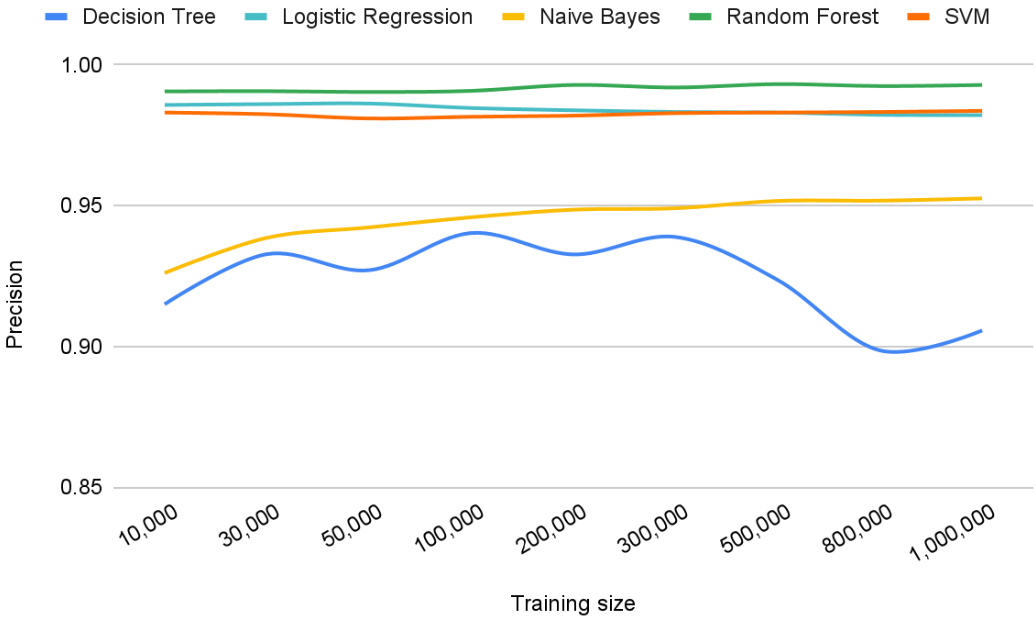


Figure 73. Mean of Precision for 1:5 ratio classical models

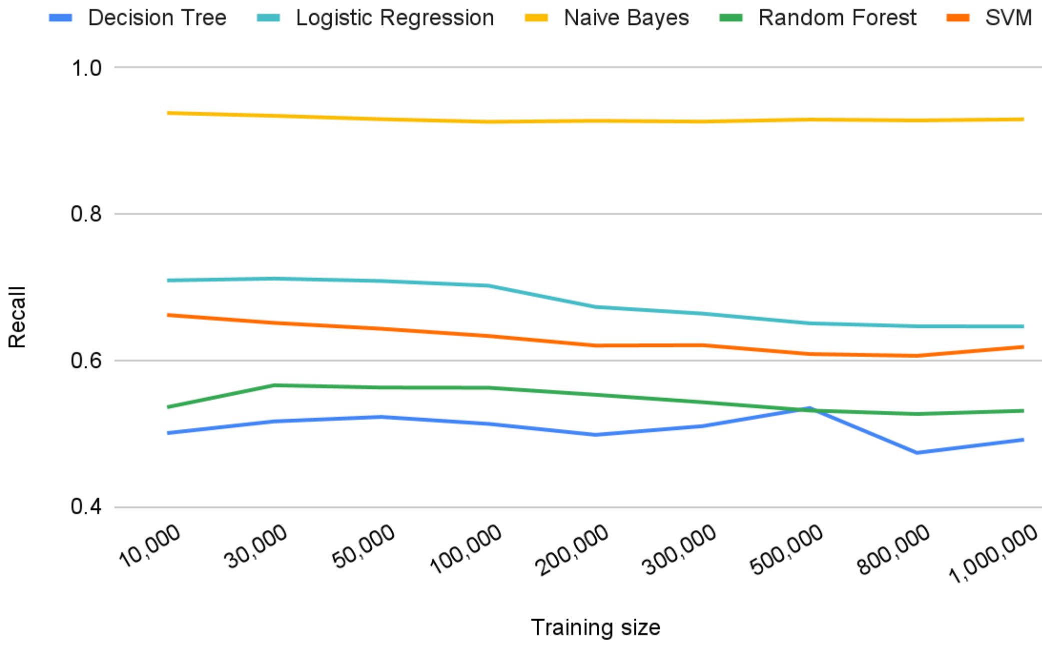


Figure 74. Mean of Recall for 1:5 ratio classical models

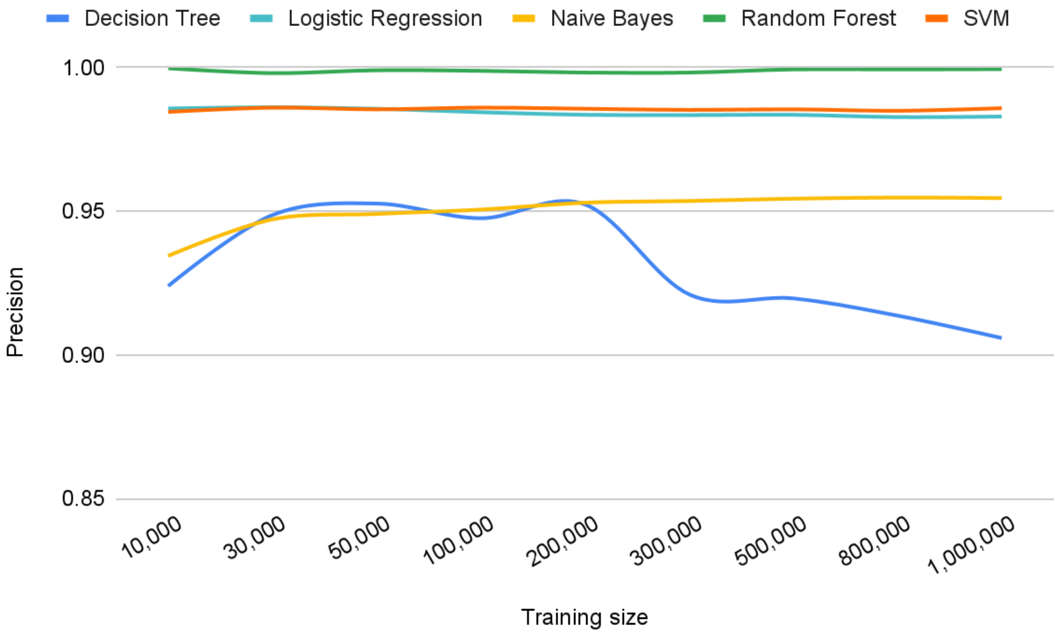


Figure 75. Mean of Precision for 1:15 ratio classical models

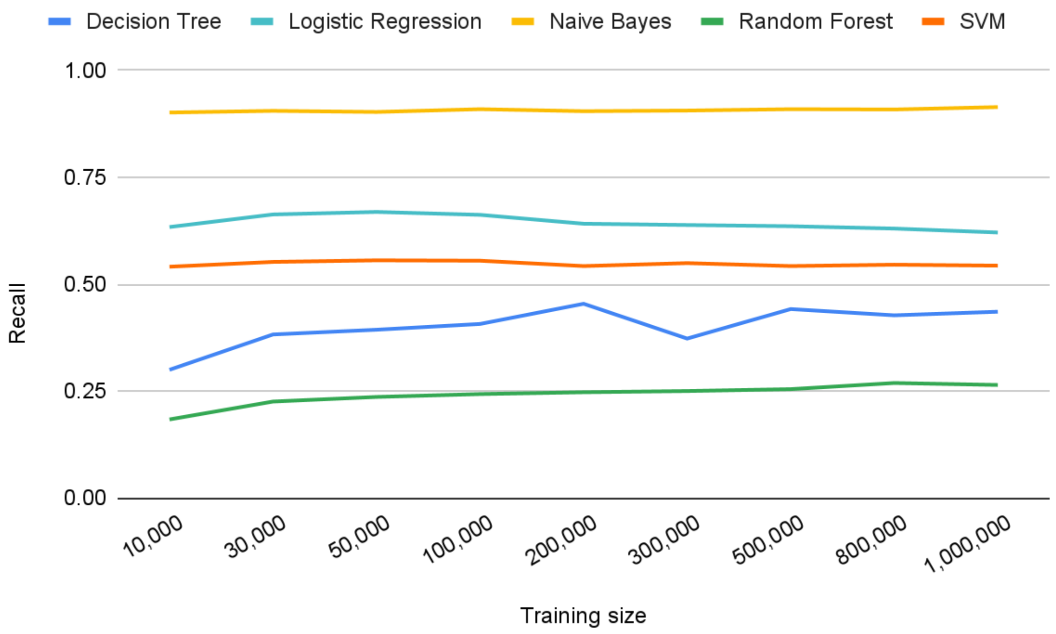


Figure 76. Mean of Recall for 1:15 ratio classical models

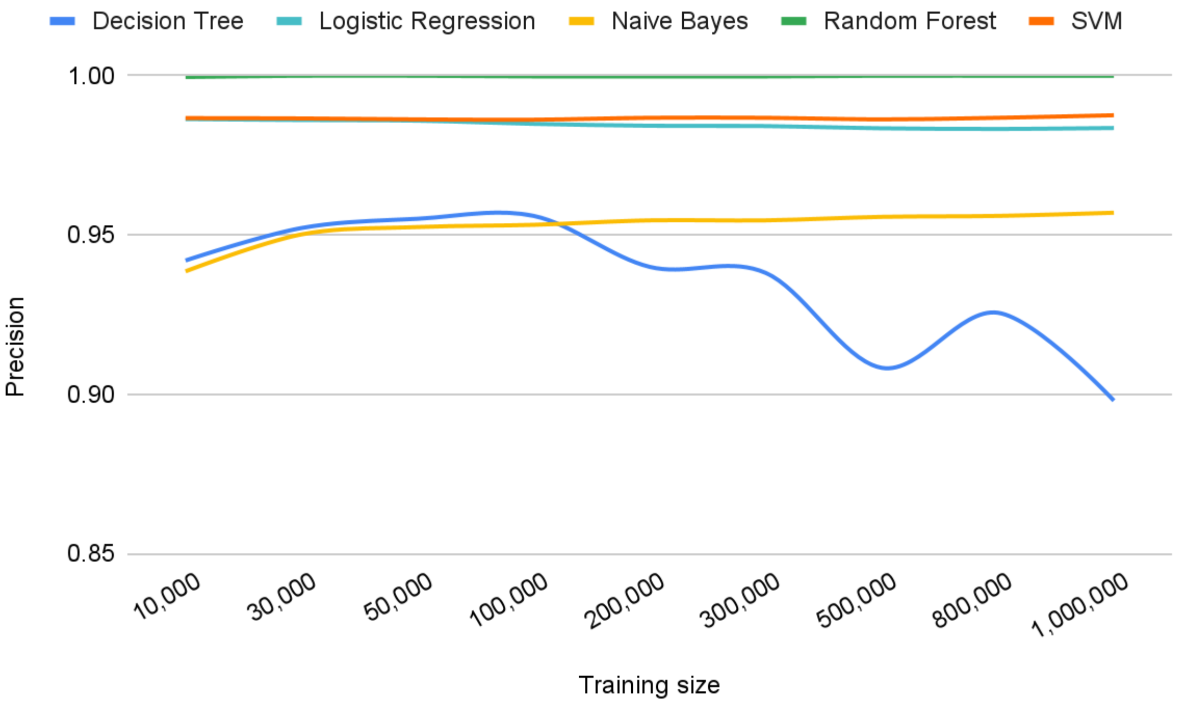


Figure 77. Mean of Precision for 1:25 ratio classical models

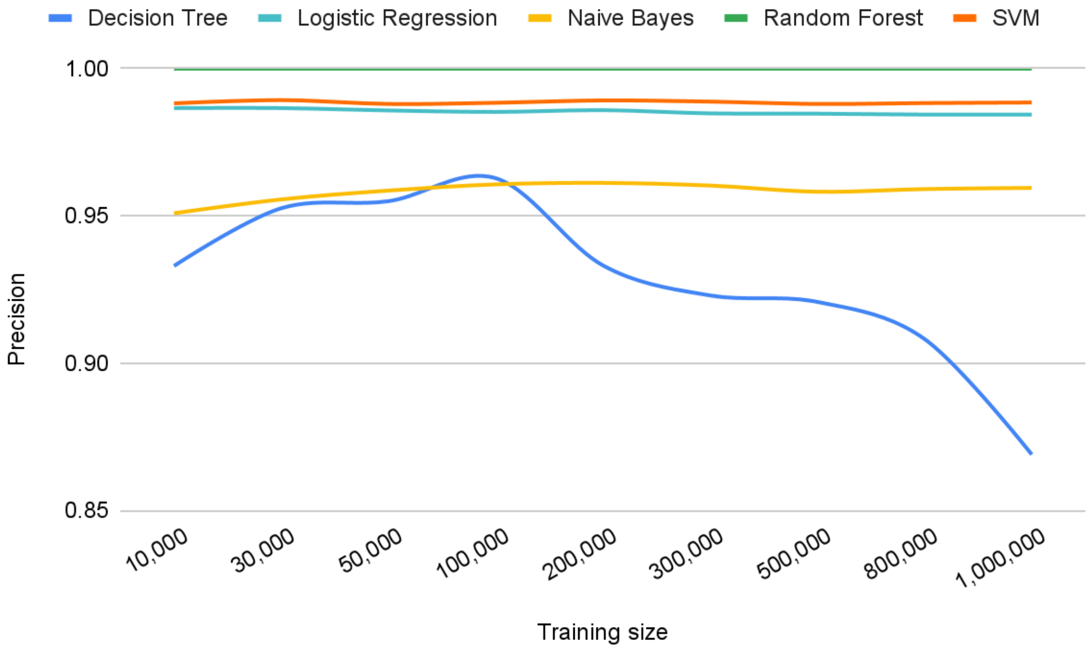


Figure 78. Mean of Precision for 1:50 ratio classical models

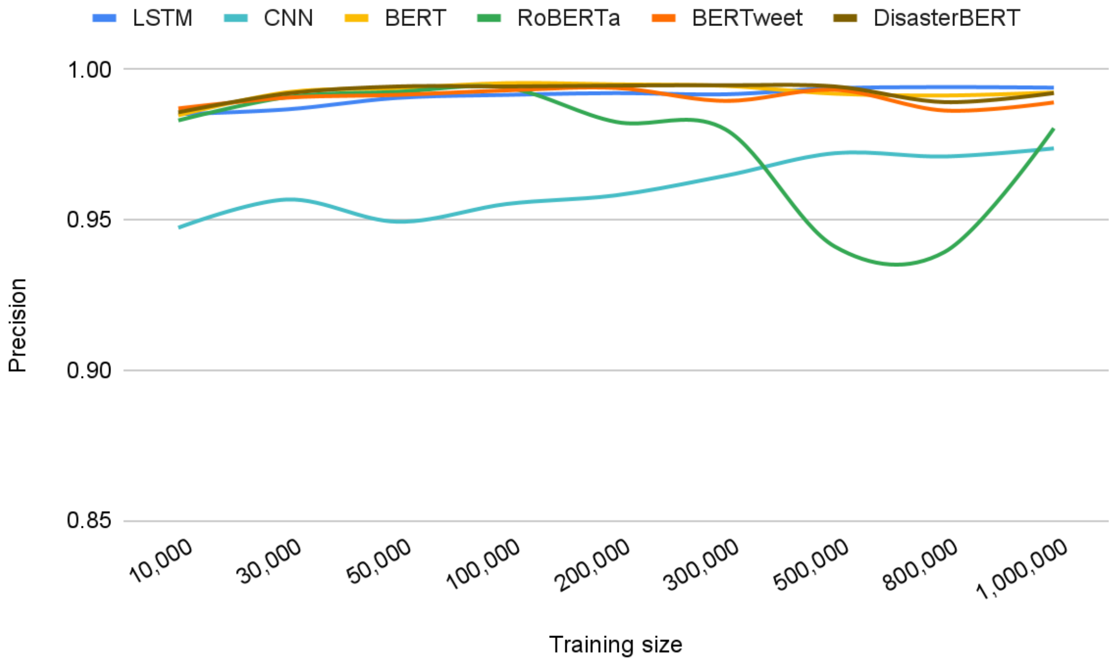


Figure 79. Mean of Precision for 1:1 ratio deep learning models

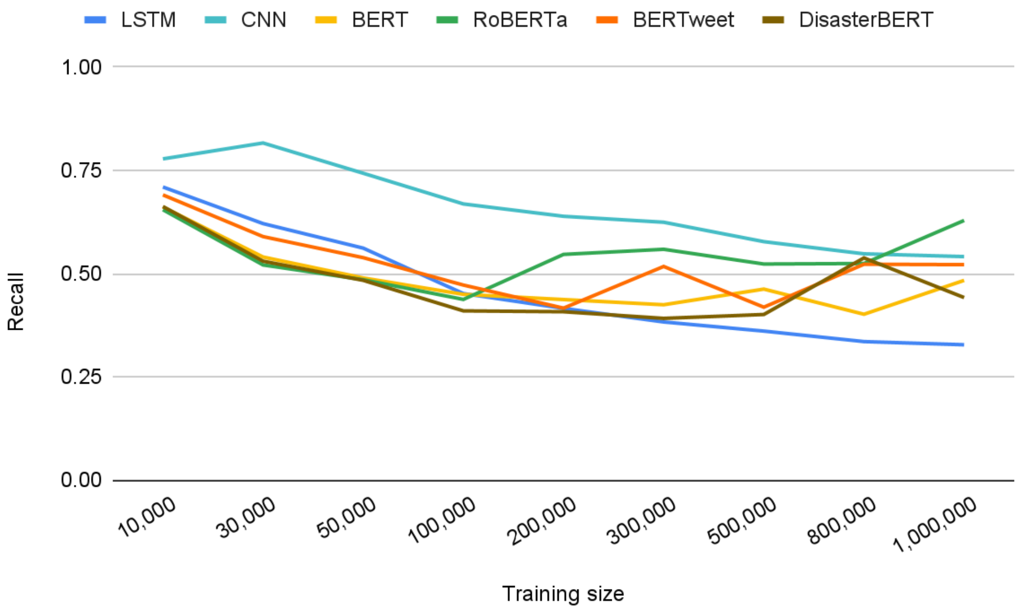


Figure 80. Mean of Recall for 1:1 ratio deep learning models

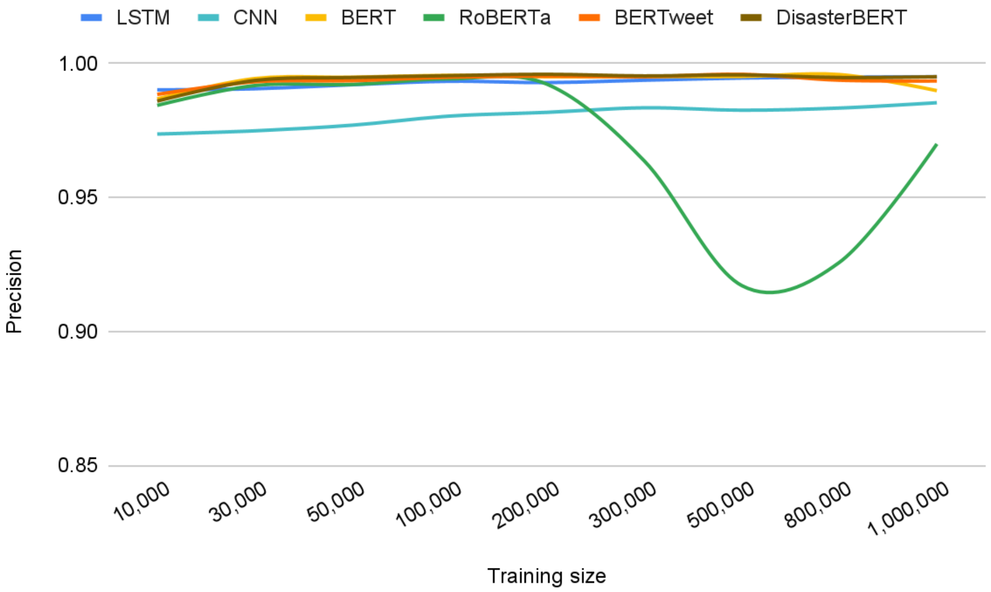


Figure 81. Mean of Precision for 1:5 ratio deep learning models

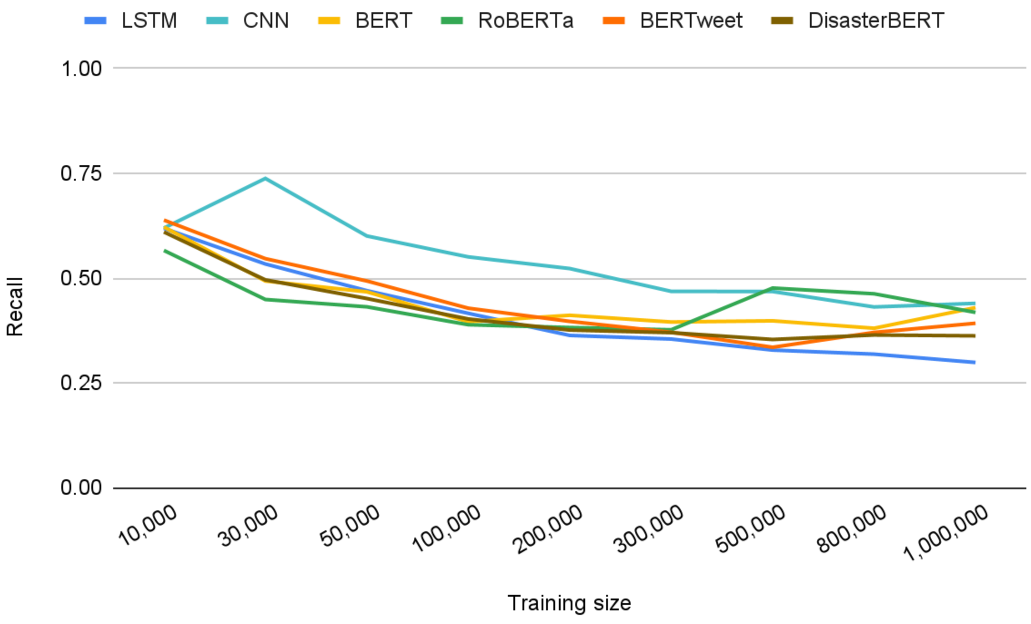


Figure 82. Mean of Recall for 1:5 ratio deep learning models

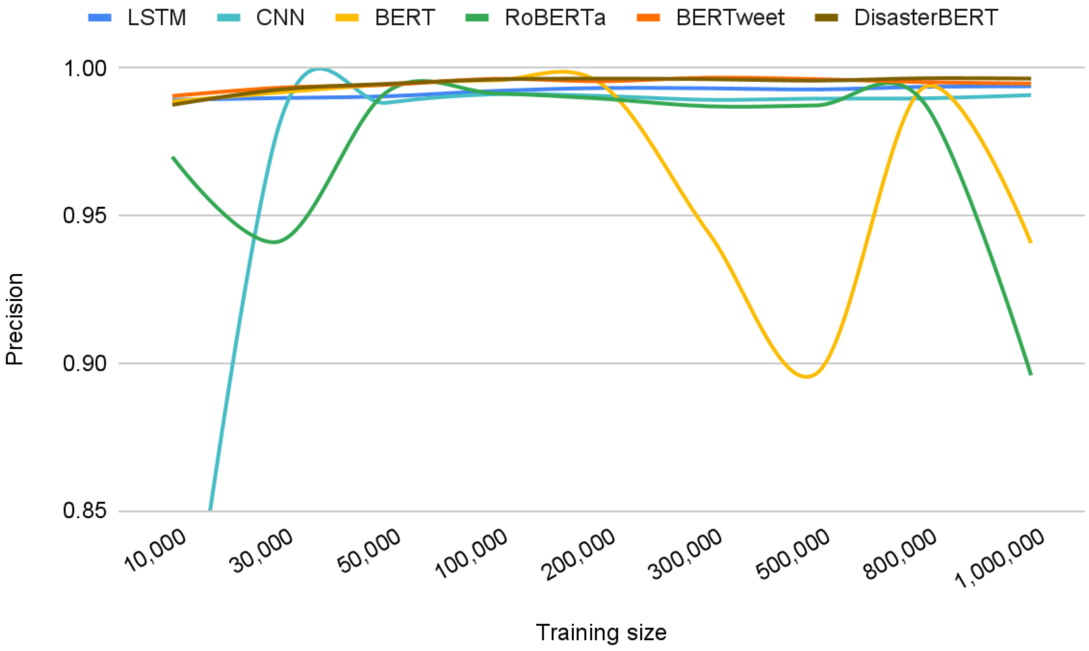


Figure 83. Mean of Precision for 1:15 ratio deep learning models

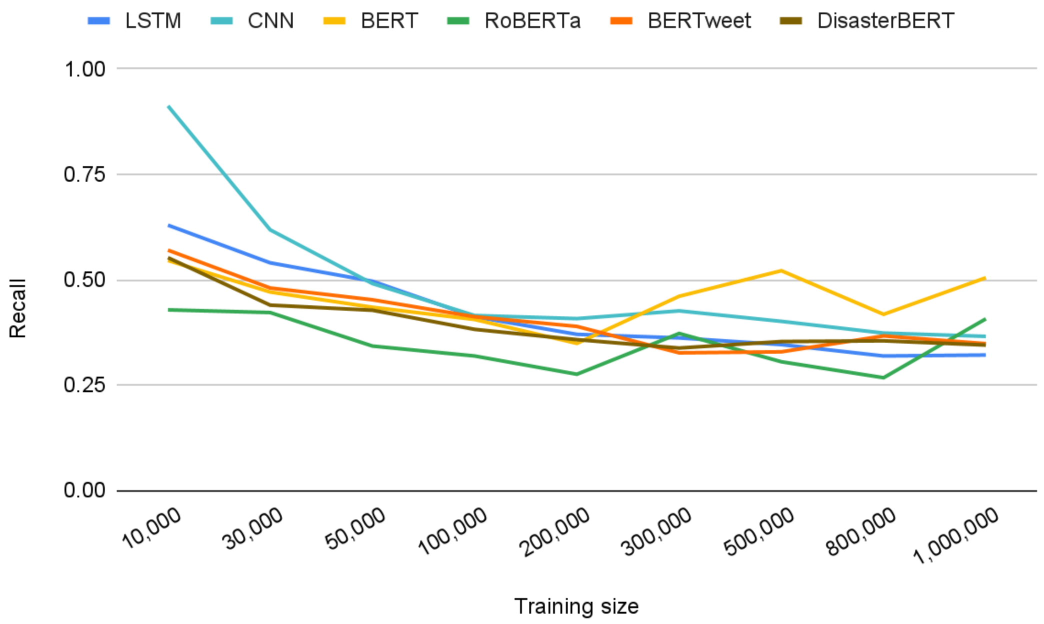


Figure 84. Mean of Recall for 1:15 ratio deep learning models

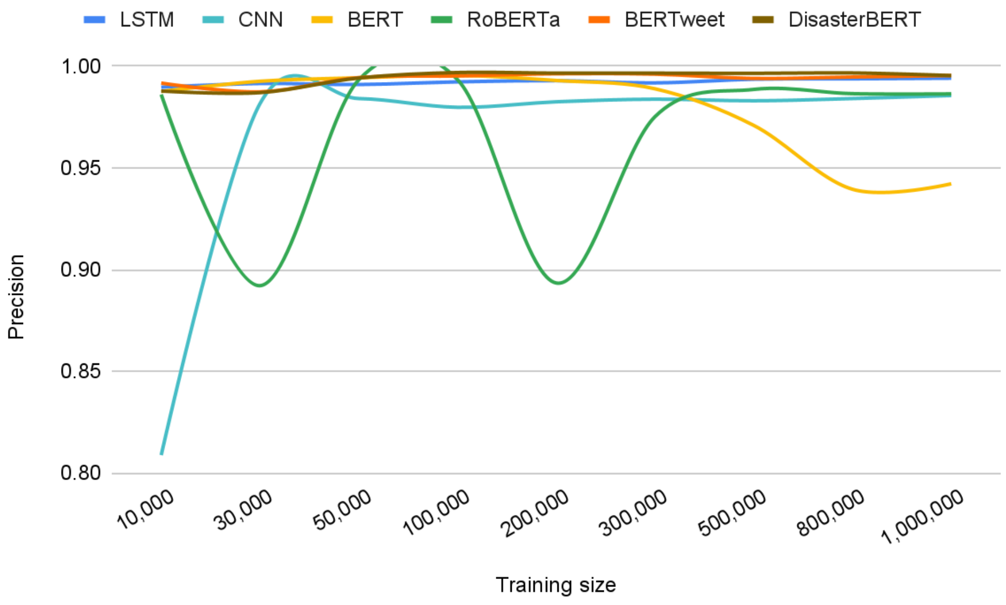


Figure 85. Mean of Precision for 1:25 ratio deep learning models

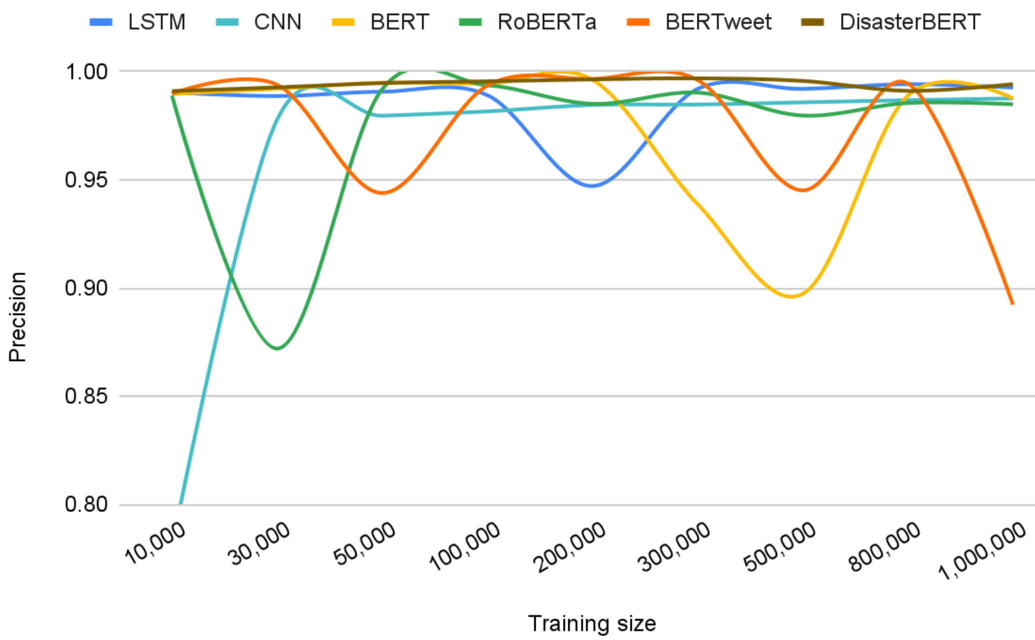


Figure 86. Mean of Precision for 1:50 ratio deep learning models

Appendix C: Detecting epidemic tweets and evaluation of large scale epidemic corpus

Regular expression used for filtering the tweets

```
“(?i:swine\s+flu|swineflu|h1n1|ebola|cholera|influenza|\\bflu\\b|yellow\s+fever|yellowfever|\\bhiv
\\b|\\b#aids\\b|\\#sars\\b|\\b#mers\\b|\\b#flu\\b|\\b#hiv\\b)|\\b#*AIDS\\b|\\bMERS\\b|\\bSARS\\b”
```

The following are the confusion matrices plots for each machine learning model for each experiment

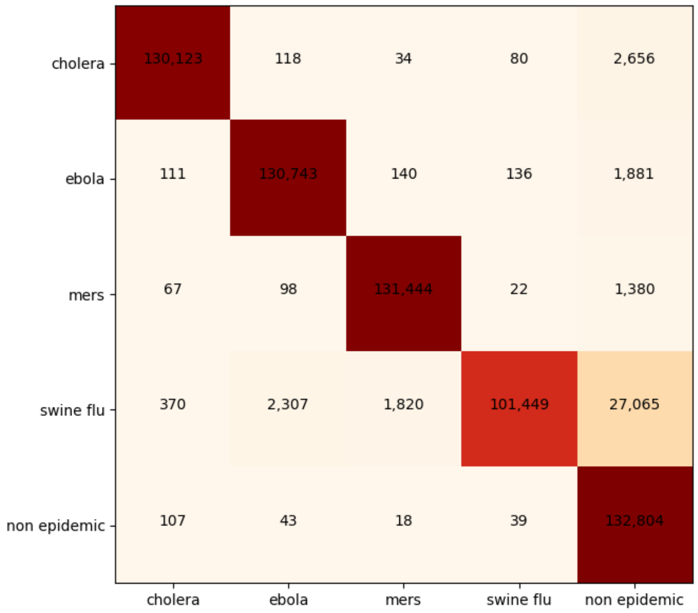


Figure 87. Confusion Matrix for Logistic Regression model for Experiment 1

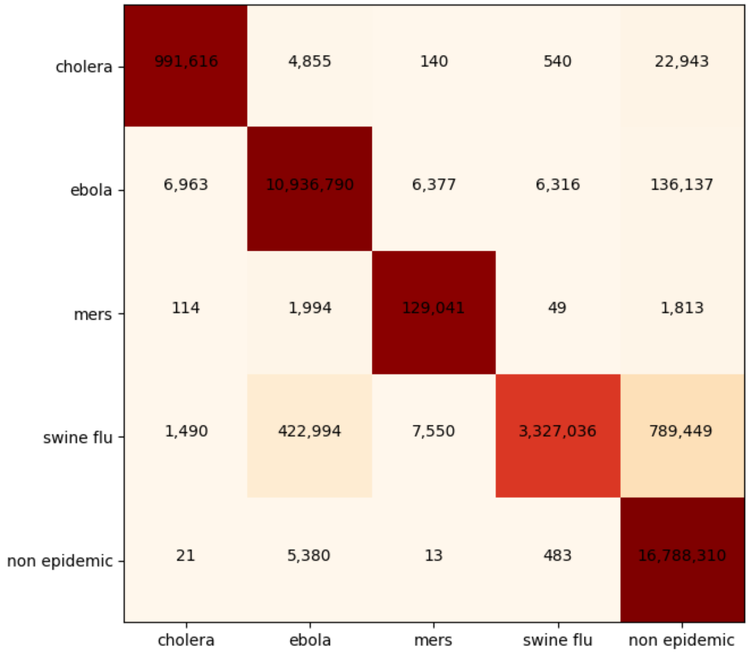


Figure 88. Confusion Matrix for Logistic Regression model for Experiment 2

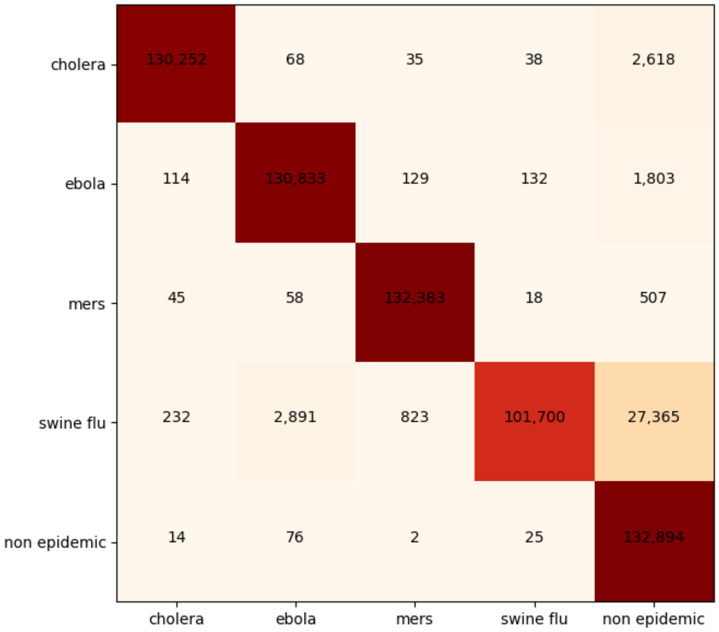


Figure 89. Confusion Matrix for SVM model for Experiment 1

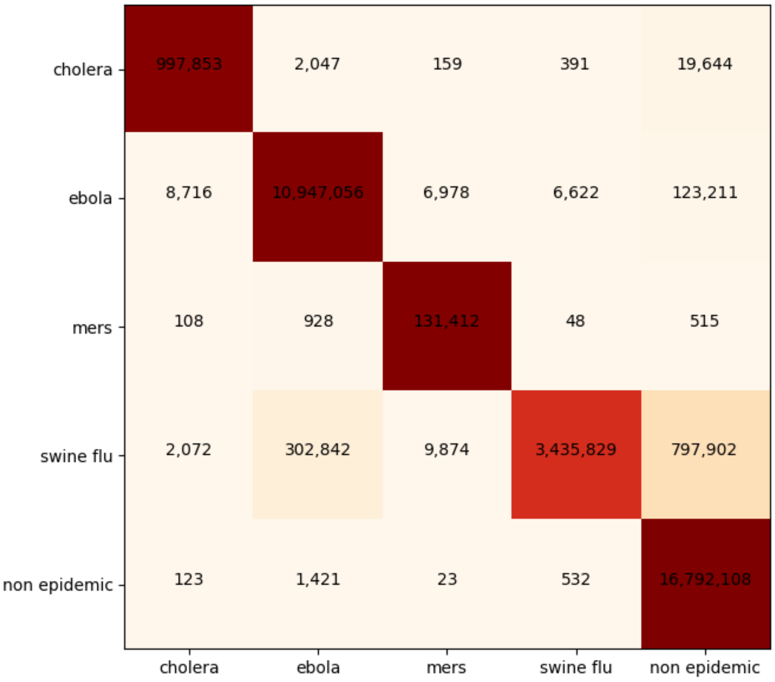


Figure 90. Confusion Matrix for SVM model for Experiment 2

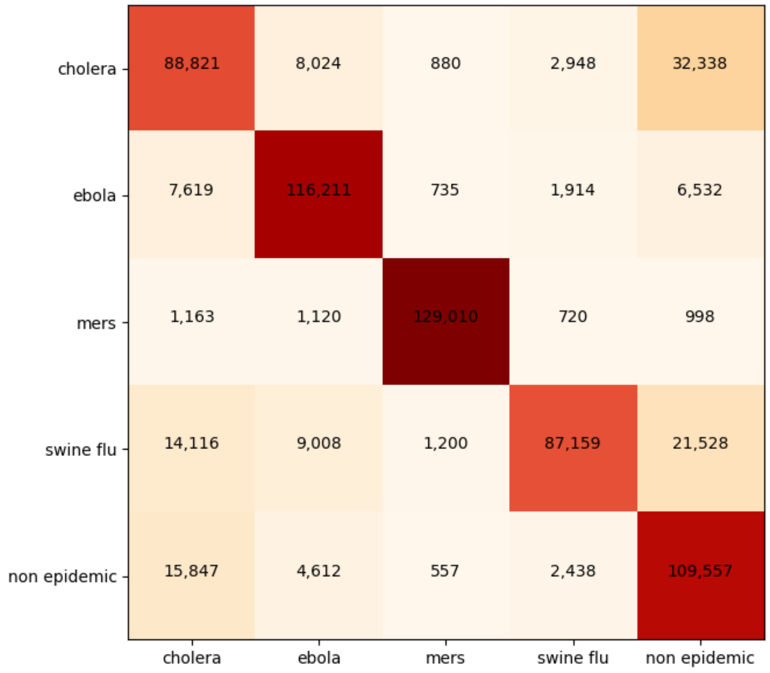


Figure 91 Confusion Matrix for Decision Tree model for Experiment 1

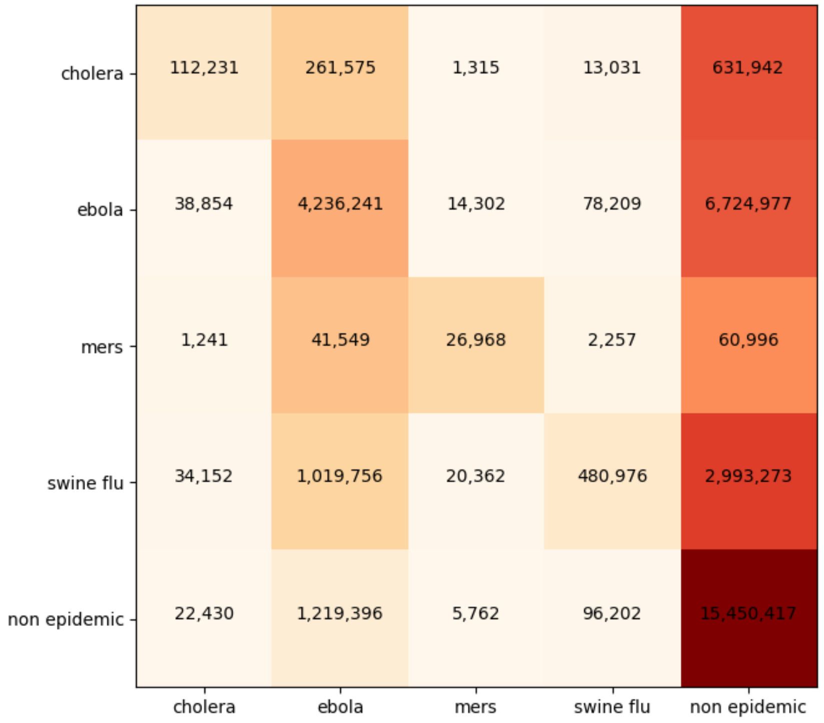


Figure 92. Confusion Matrix for Decision Tree model for Experiment 2

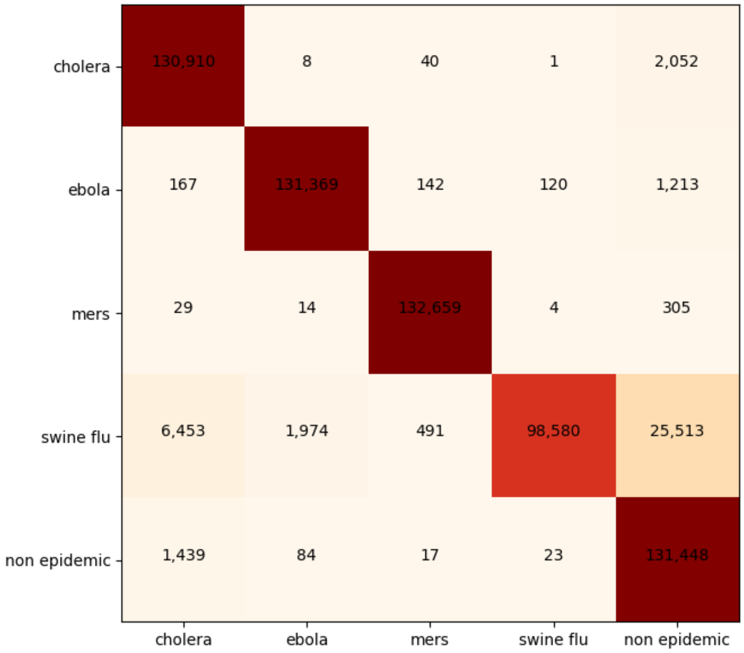


Figure 93. Confusion Matrix for BERTweet model for Experiment 1

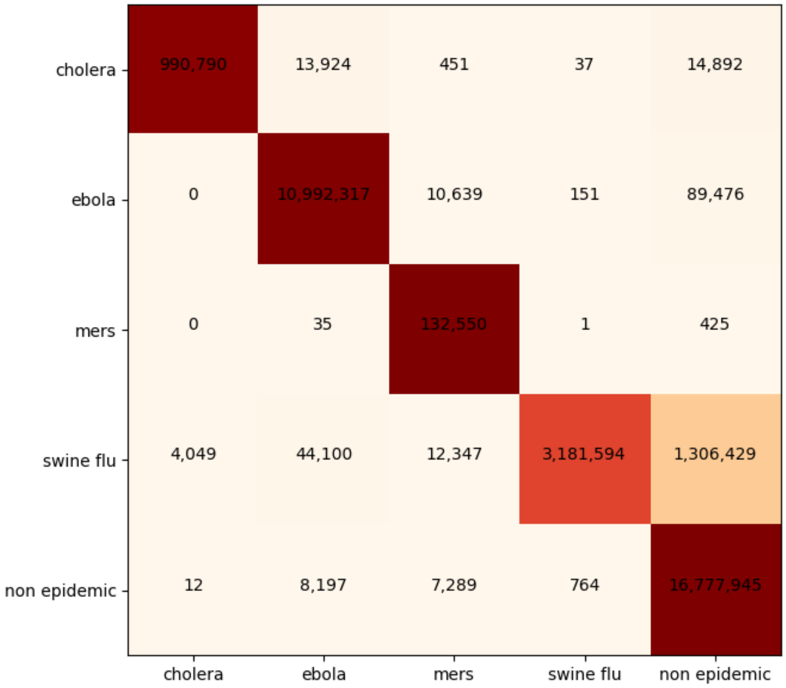


Figure 94. Confusion Matrix for BERT model for Experiment 2

The following are the plots when trained on the "flu" silver standard dataset and tested on one class (flu). The plots represent the mean of the models for 10 experiments in a training size and ratio.

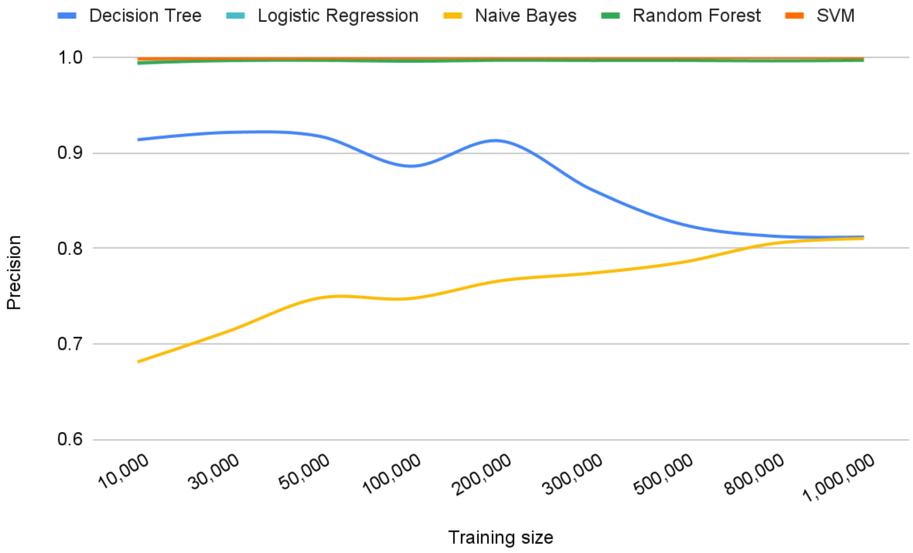


Figure 95. Progression of Precision mean for 1:1 ratio of classical models

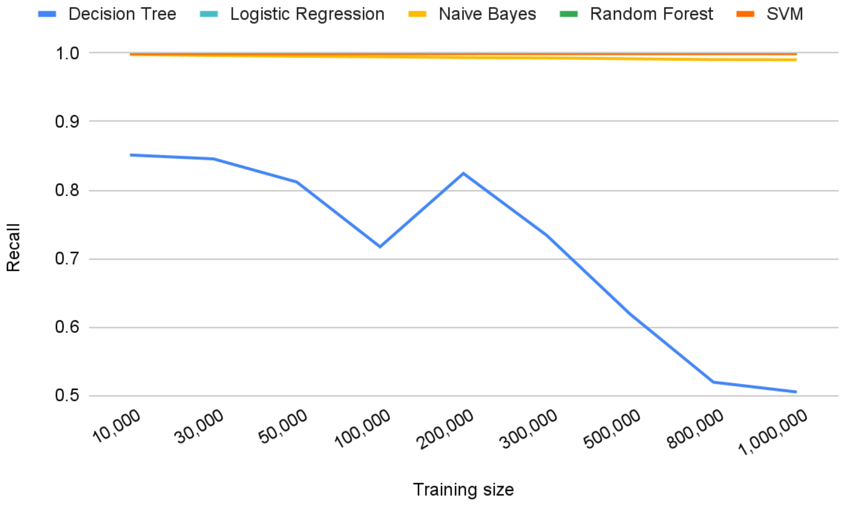


Figure 96. Progression of Recall mean for 1:1 ratio of classical models

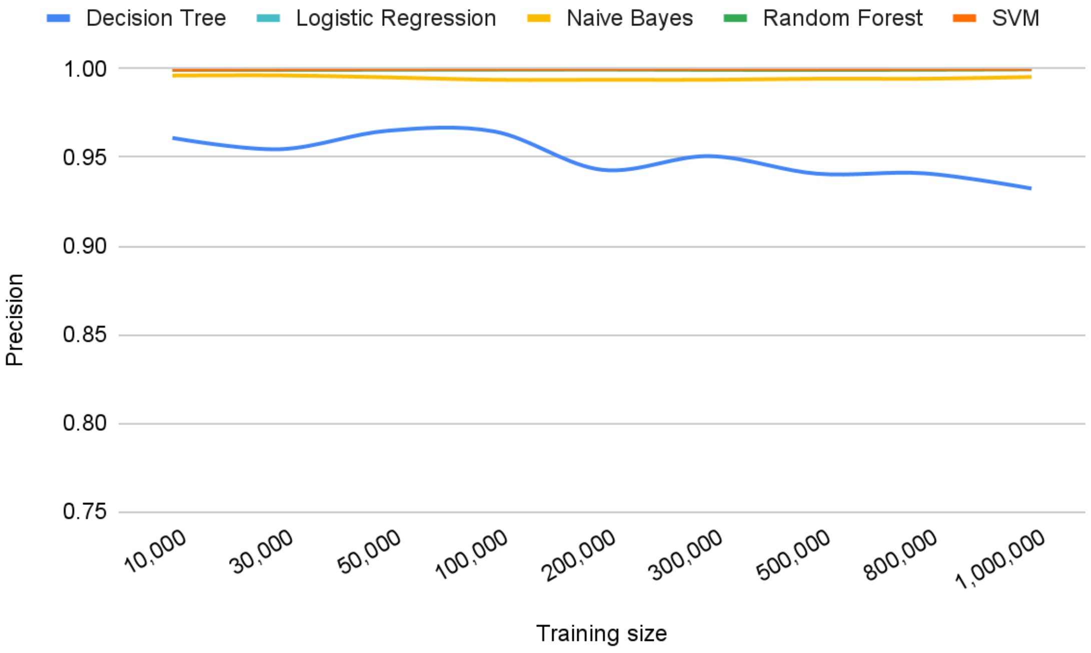


Figure 97. Progression of Precision mean for 1:5 ratio of classical models

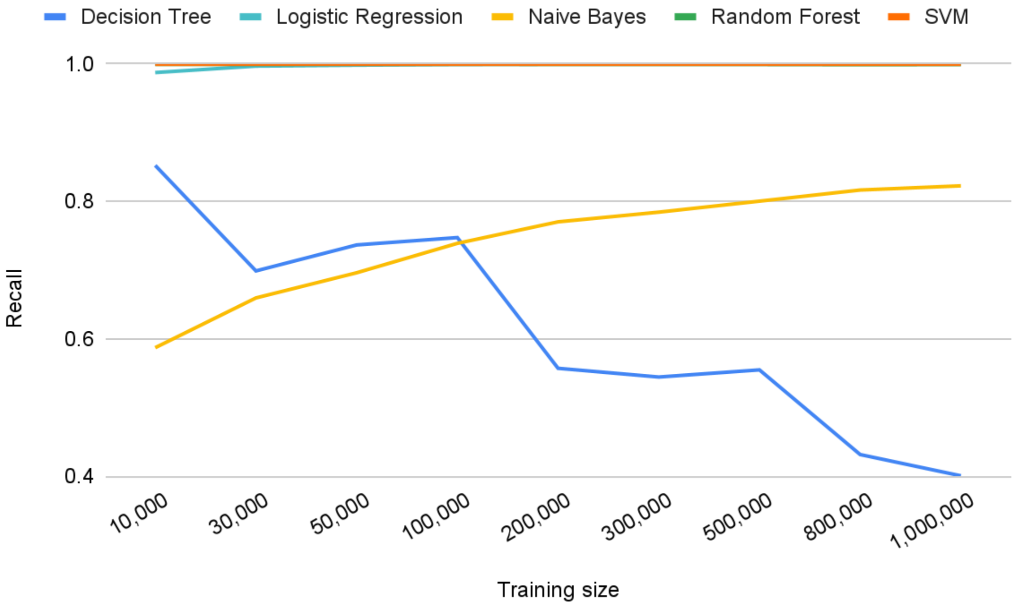


Figure 98. Progression of Recall mean for 1:5 ratio of classical models

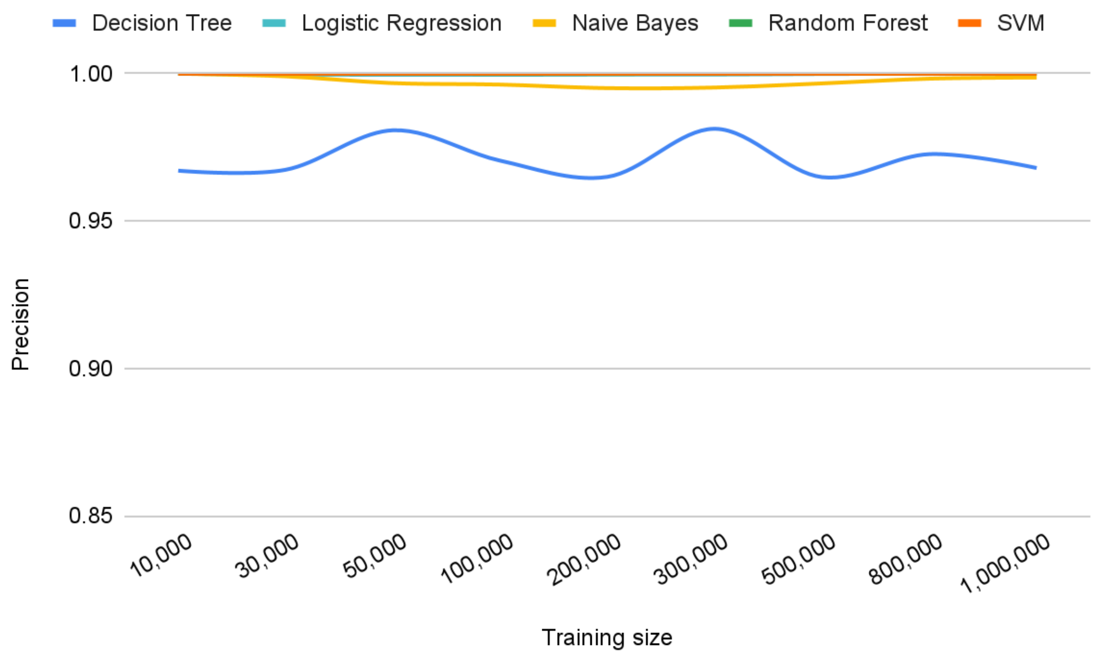


Figure 99. Progression of Precision mean for 1:15 ratio of classical models

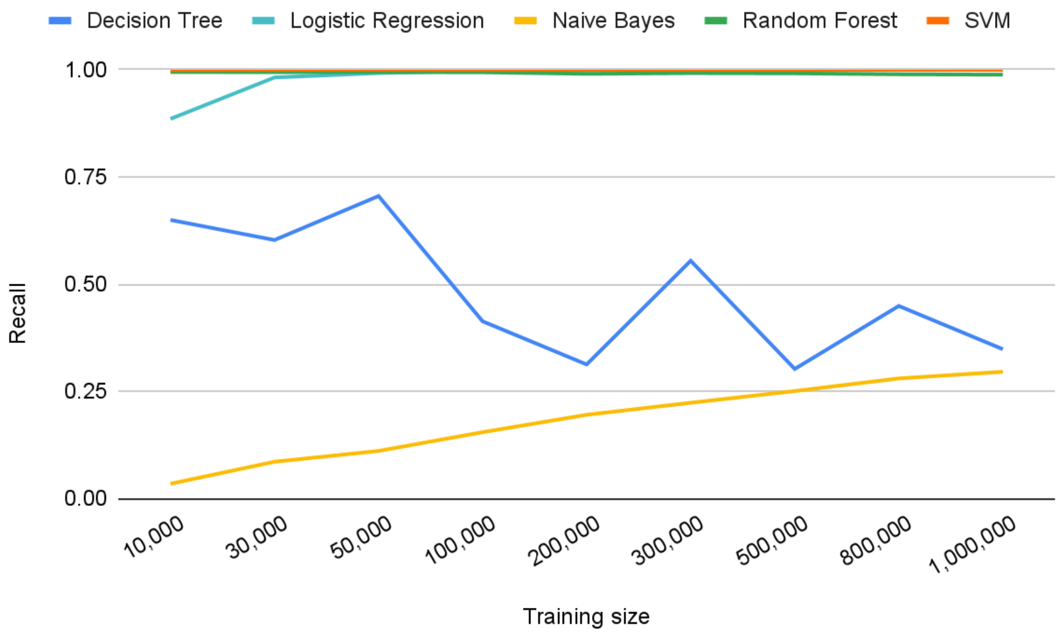


Figure 100. Progression of Recall mean for 1:15 ratio of classical models

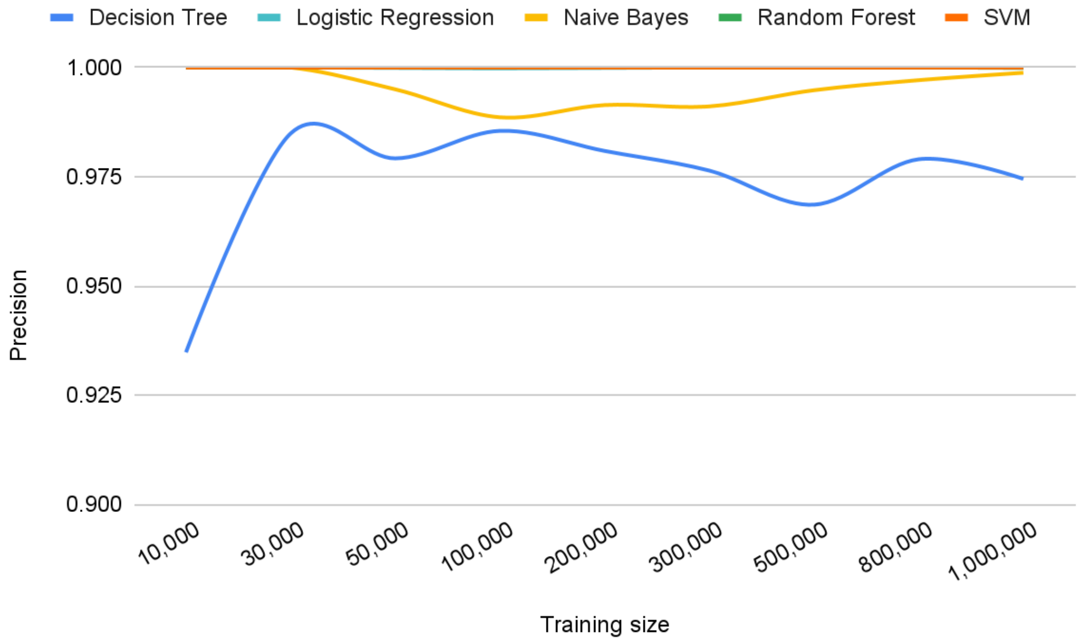


Figure 101. Progression of Precision mean for 1:25 ratio of classical models

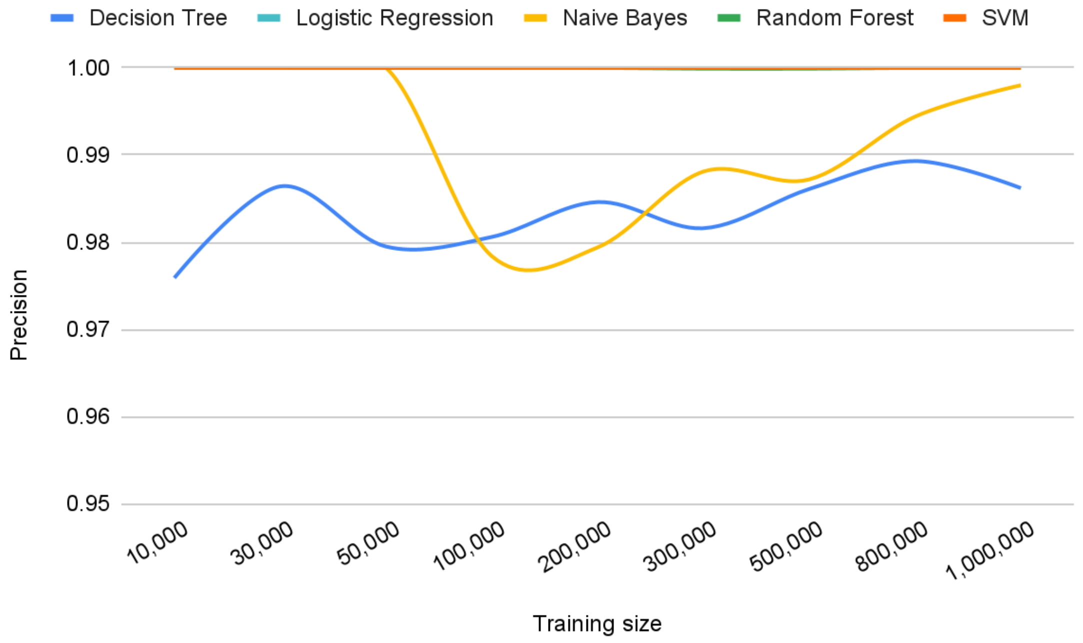


Figure 102. Progression of Precision mean for 1:50 ratio of classical models

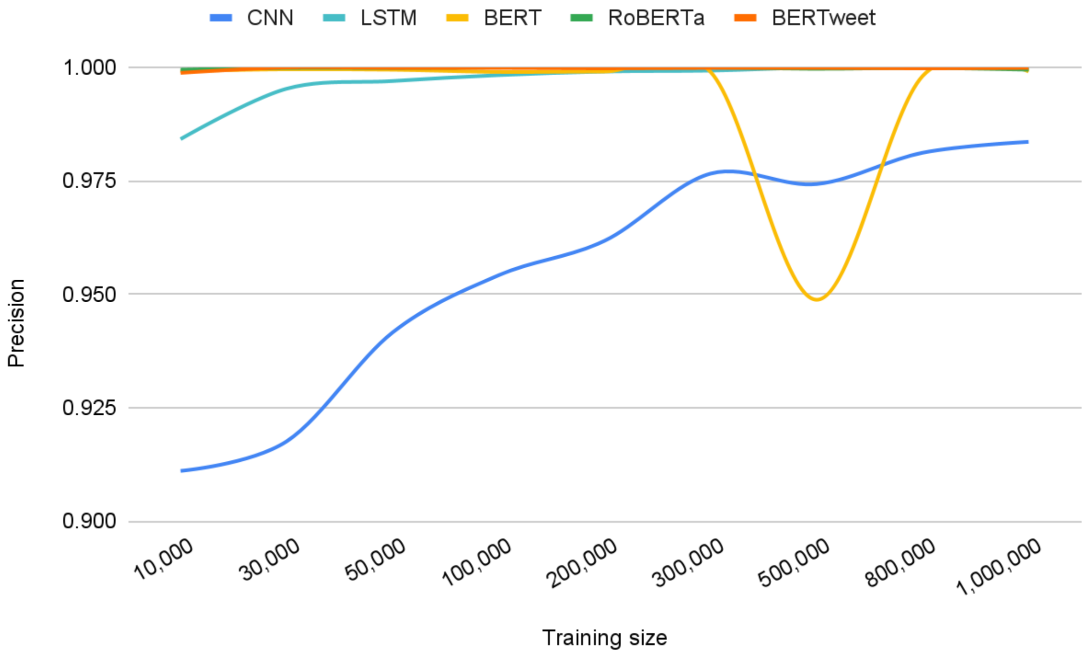


Figure 103. Progression of Precision mean for 1:1 ratio of deep learning models

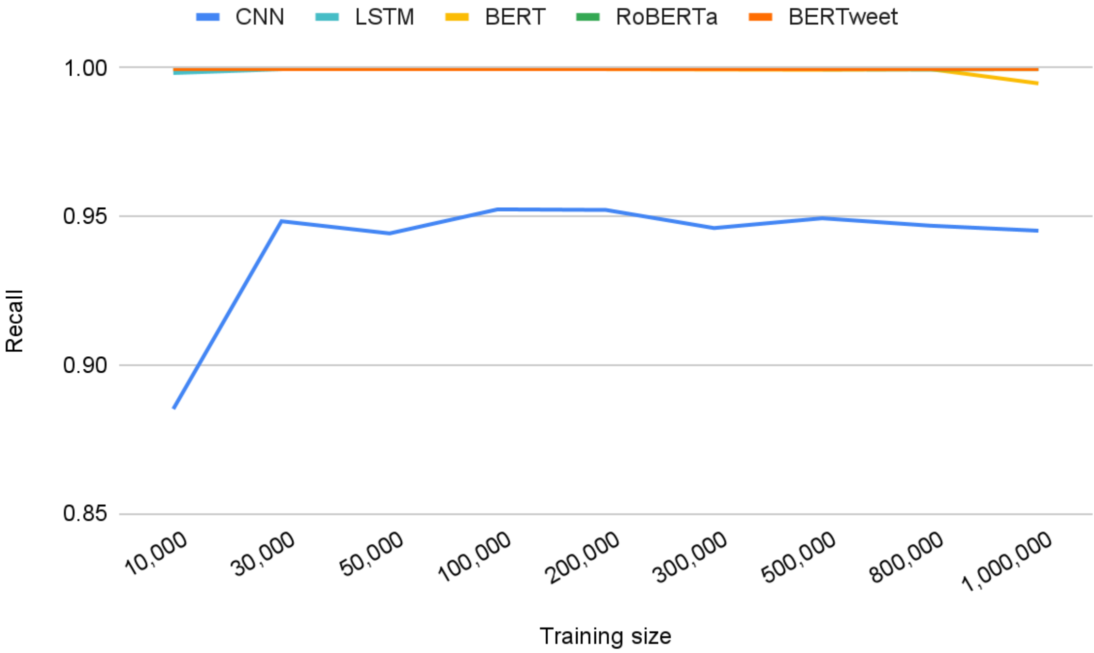


Figure 104. Progression of Recall mean for 1:1 ratio of deep learning models

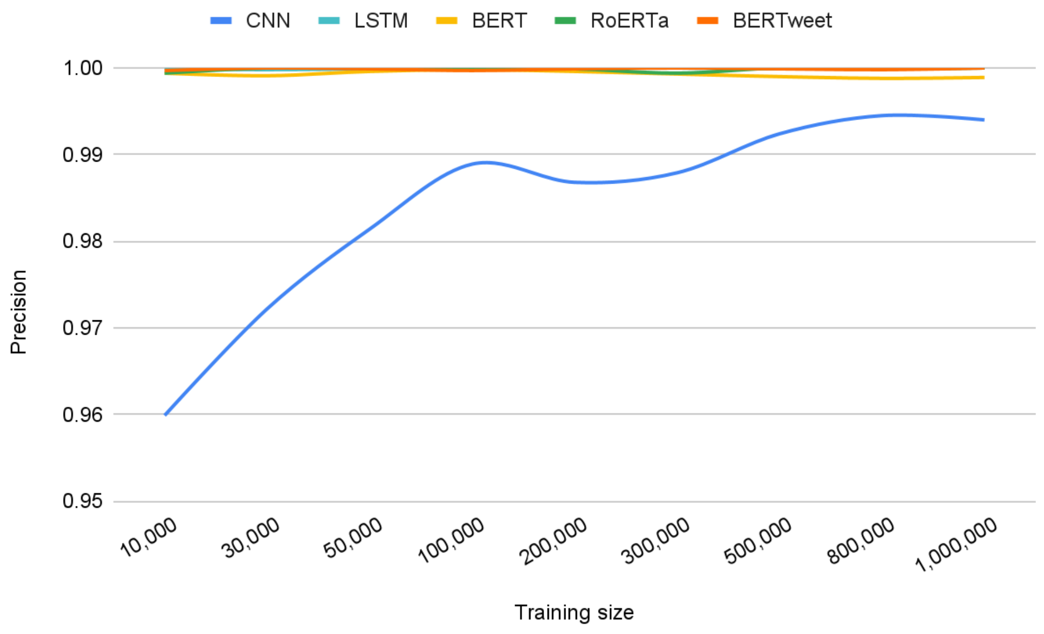


Figure 105. Progression of Precision mean for 1:5 ratio of deep learning models

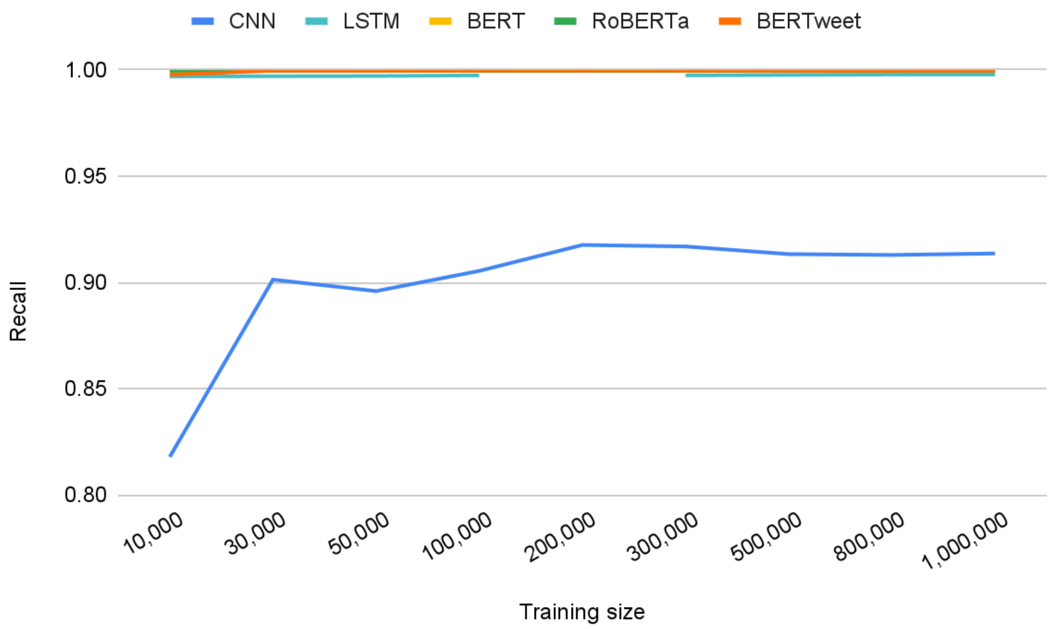


Figure 106. Progression of Recall mean for 1:5 ratio of deep learning models

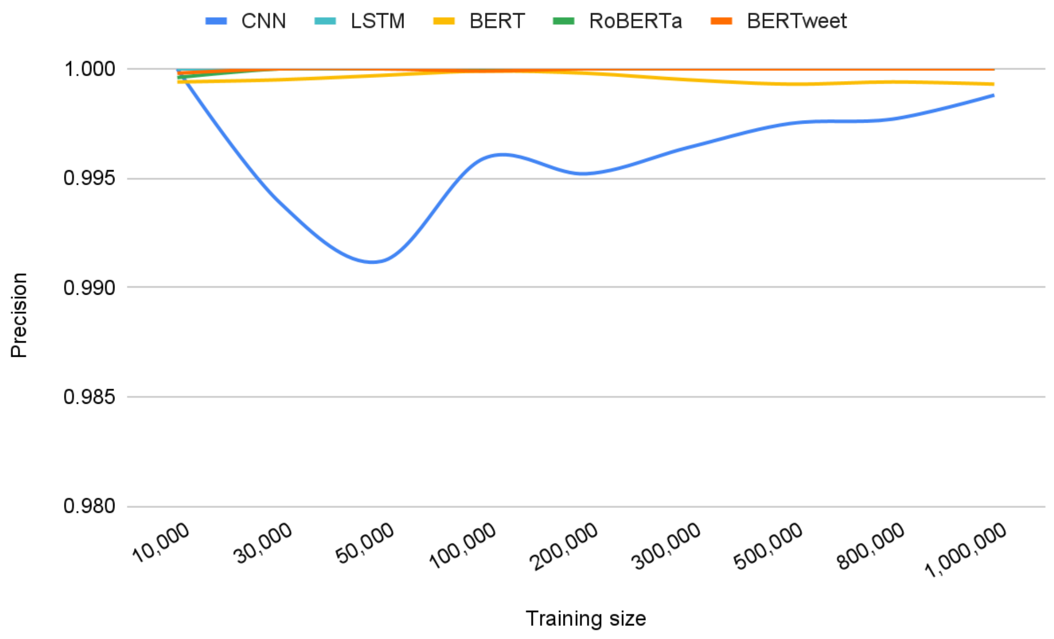


Figure 107. Progression of Precision mean for 1:15 ratio of deep learning models

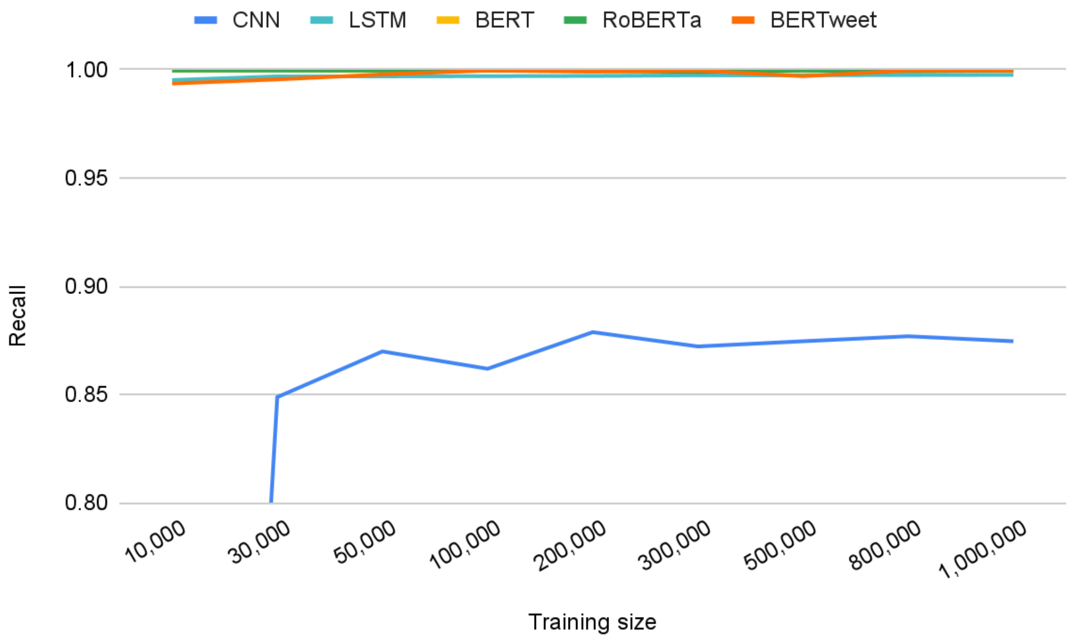


Figure 108. Progression of Recall mean for 1:15 ratio of deep learning models

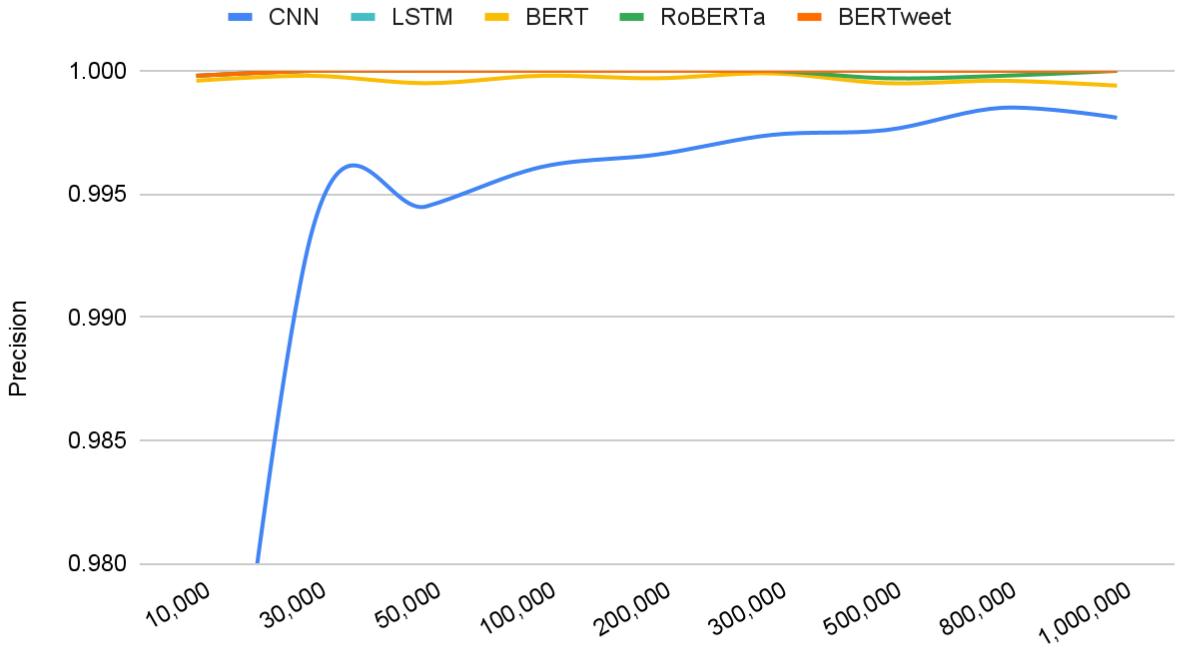


Figure 109. Progression of Precision mean for 1:25 ratio of deep learning models

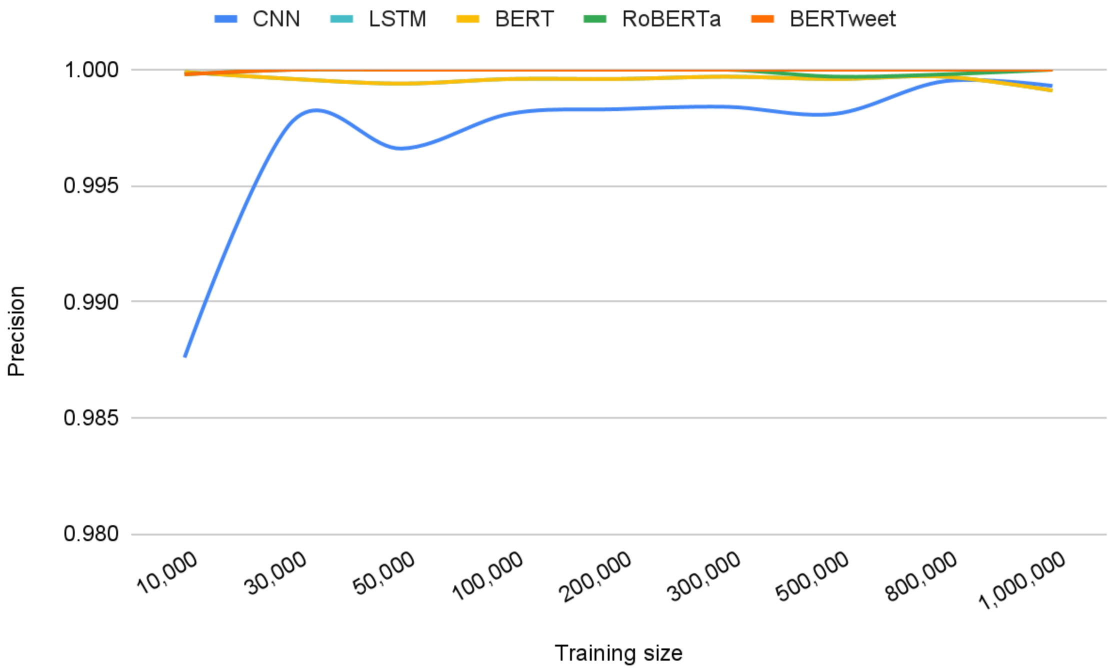


Figure 110. Progression of Precision mean for 1:50 ratio of deep learning models

Appendix D: Characterizing relevant health tweets

The following table presents the number of iterations required for obtaining optimal performance for generating the pseudo gold standard dataset.

Table 31. Number of iterations required to obtain optimal performance.

Class	Sub Class	Precision	Recall	F-Measure	Accuracy	Total no of Iterations
Pregnancy	pregnant	0.7067	0.6897	0.6908	0.6897	2
		0.818	**0.8088**	**0.809**	**0.8088**	
	miscarriage	0.6515	0.65	0.6491	0.65	2
		0.8703	**0.8704**	**0.8701**	**0.8704**	
	abortion	0.8244	0.7073	0.6923	0.7073	2
		0.8617	**0.8444**	**0.8421**	**0.8444**	
Mental Health	anxiety attack	0.5935	0.561	0.5547	0.561	4
		0.7057	0.7073	0.7048	0.7073	
		0.8587	0.8049	0.7961	0.8049	

		0.8997	**0.8723**	**0.8713**	**0.8723**	
	insomnia	0.6703	0.6607	0.6602	0.6607	5
		0.8062	0.6964	0.6562	0.6964	
		0.8764	0.8393	0.8329	0.8393	
		0.8986	0.875	0.8718	0.875	
		0.9107	**0.8929**	**0.8907**	**0.8929**	
	panic attack	0.5142	0.4634	0.4029	0.4634	6
		0.6221	0.6222	0.6218	0.6222	
		0.5551	0.5556	0.5542	0.5556	
		0.7752	0.7111	0.6908	0.7111	
		0.8633	**0.8039**	**0.8001**	**0.8039**	
		0.8966	0.8621	0.8627	0.8621	
	suicidal	0.6515	0.65	0.6491	0.65	3
		0.7604	0.75	0.7475	0.75	
		0.9149	**0.9107**	**0.9101**	**0.9107**	
	depression	0.5752	0.575	0.5747	0.575	2
		0.9348	**0.925**	**0.9246**	**0.925**	
Heart Conditions	acid reflux	**0.9766**	**0.9756**	**0.9755**	**0.9756**	1
	chest pain	0.7073	0.7073	0.7073	0.7073	2
		0.8892	**0.8636**	**0.8571**	**0.8636**	
	heartburn	0.839	0.8049	0.8042	0.8049	3
		0.8537	0.8537	0.8537	0.8537	
		0.9283	**0.9268**	**0.927**	**0.9268**	

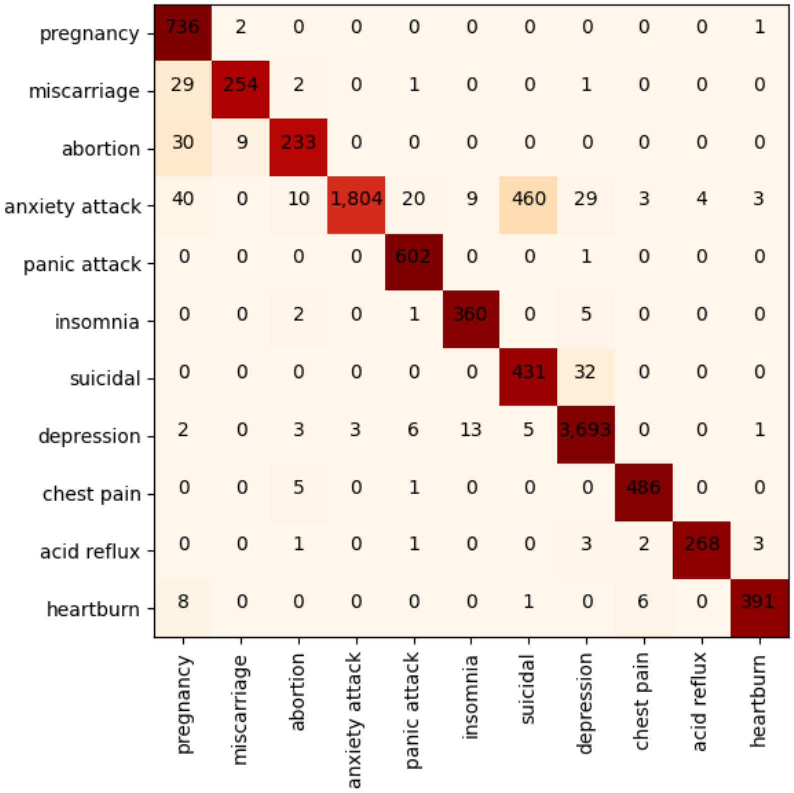
The following table presents the number of labeled tweets in the manually labeled dataset.

Table 32. Total number of manually labeled tweets

Class	**Sub Class**	**Total Tweets**	**Self-reported Tweets**	**Positive Label**	**Negative Label**	**Undecided Label**
Pregnancy	pregnant	256,960	79,696	167	337	246
	miscarriage	9,568	4,465	134	218	191
	abortion	175,520	925	110	113	131

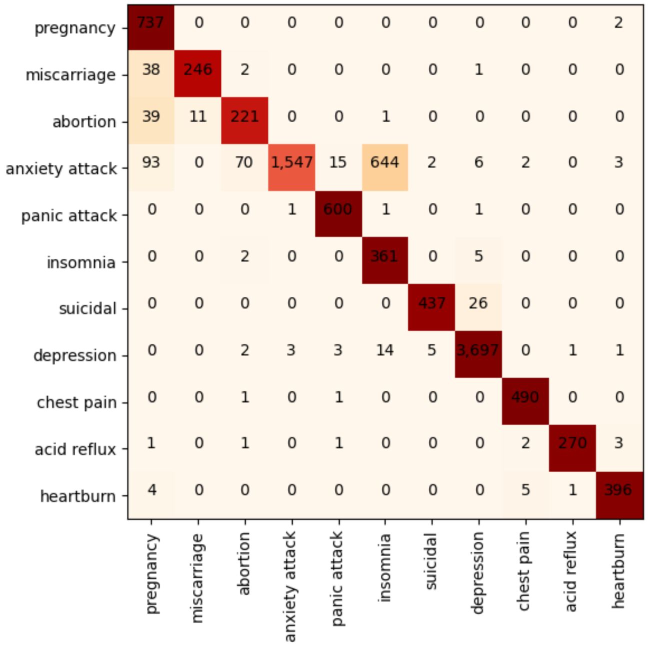
Mental Health	anxiety attack	11,633	5,874	116	130	197
	insomnia	39,086	10,484	137	180	379
	panic attack	29,413	15,754	143	234	155
	suicidal	40,404	12,934	138	150	194
	depression	254,774	115,958	100	113	171
Heart Conditions	acid reflux	5,247	655	101	119	24
	chest pain	7,407	928	110	125	19
	heartburn	8,872	2,123	101	101	61

The following are the confusion matrices for deep learning models for each experiment.



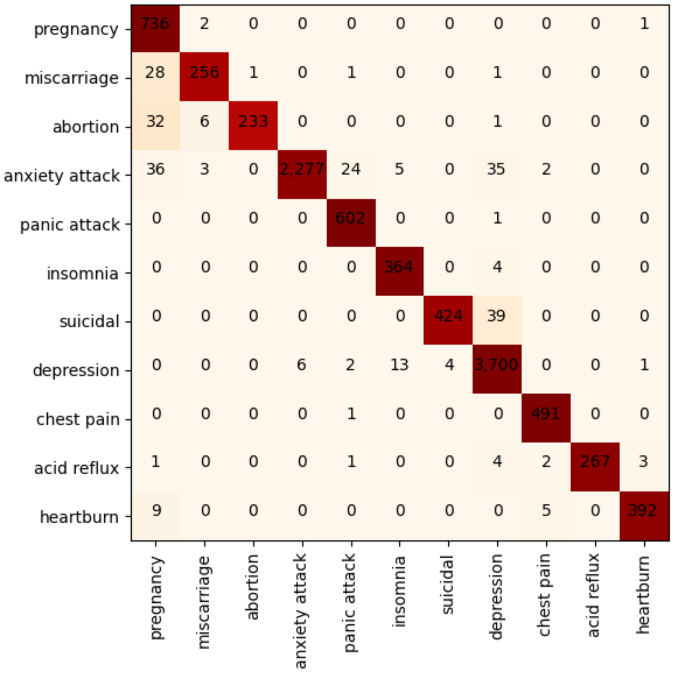
	pregnancy	miscarriage	abortion	anxiety attack	panic attack	insomnia	suicidal	depression	chest pain	acid reflux	heartburn
pregnancy	736	2	0	0	0	0	0	0	0	0	1
miscarriage	29	254	2	0	1	0	0	1	0	0	0
abortion	30	9	233	0	0	0	0	0	0	0	0
anxiety attack	40	0	10	1,804	20	9	460	29	3	4	3
panic attack	0	0	0	0	602	0	0	1	0	0	0
insomnia	0	0	2	0	1	360	0	5	0	0	0
suicidal	0	0	0	0	0	0	431	32	0	0	0
depression	2	0	3	3	6	13	5	3,693	0	0	1
chest pain	0	0	5	0	1	0	0	0	486	0	0
acid reflux	0	0	1	0	1	0	0	3	2	268	3
heartburn	8	0	0	0	0	0	1	0	6	0	391

Figure 111. Confusion Matrix for BERT model for Experiment 1



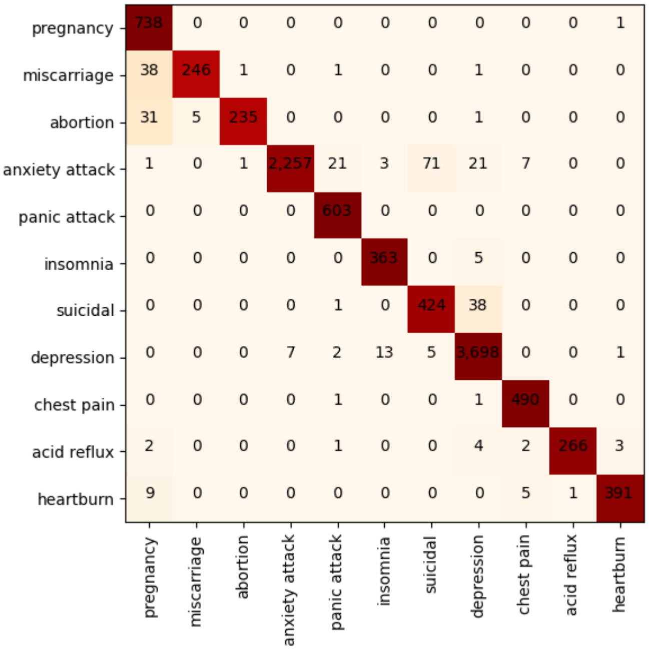
	pregnancy	miscarriage	abortion	anxiety attack	panic attack	insomnia	suicidal	depression	chest pain	acid reflux	heartburn
pregnancy	737	0	0	0	0	0	0	0	0	0	2
miscarriage	38	246	2	0	0	0	0	1	0	0	0
abortion	39	11	221	0	0	1	0	0	0	0	0
anxiety attack	93	0	70	1,547	15	644	2	6	2	0	3
panic attack	0	0	0	1	600	1	0	1	0	0	0
insomnia	0	0	2	0	0	361	0	5	0	0	0
suicidal	0	0	0	0	0	0	437	26	0	0	0
depression	0	0	2	3	3	14	5	3,697	0	1	1
chest pain	0	0	1	0	1	0	0	0	490	0	0
acid reflux	1	0	1	0	1	0	0	0	2	270	3
heartburn	4	0	0	0	0	0	0	0	5	1	396

Figure 112. Confusion Matrix for RoBERTa model for Experiment 1



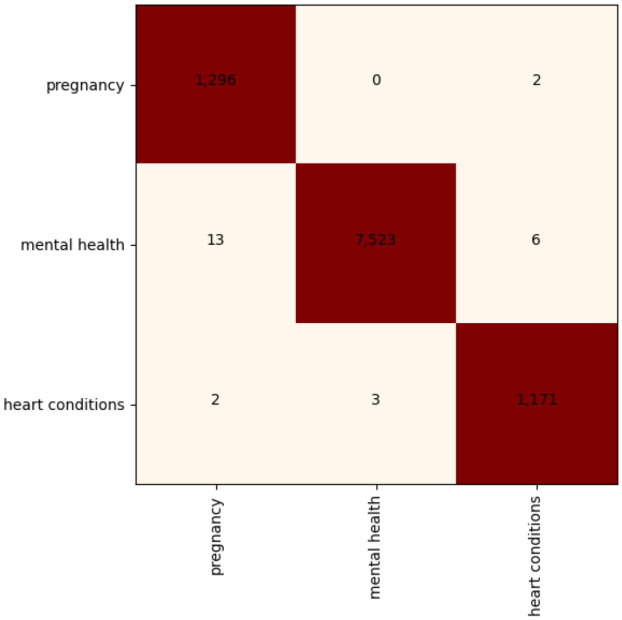
	pregnancy	miscarriage	abortion	anxiety attack	panic attack	insomnia	suicidal	depression	chest pain	acid reflux	heartburn
pregnancy	736	2	0	0	0	0	0	0	0	0	1
miscarriage	28	256	1	0	1	0	0	1	0	0	0
abortion	32	6	233	0	0	0	0	1	0	0	0
anxiety attack	36	3	0	2,277	24	5	0	35	2	0	0
panic attack	0	0	0	0	602	0	0	1	0	0	0
insomnia	0	0	0	0	0	364	0	4	0	0	0
suicidal	0	0	0	0	0	0	424	39	0	0	0
depression	0	0	0	6	2	13	4	3,700	0	0	1
chest pain	0	0	0	0	1	0	0	0	491	0	0
acid reflux	1	0	0	0	1	0	0	4	2	267	3
heartburn	9	0	0	0	0	0	0	0	5	0	392

Figure 113. Confusion Matrix for BERTweet model for Experiment 1



	pregnancy	miscarriage	abortion	anxiety attack	panic attack	insomnia	suicidal	depression	chest pain	acid reflux	heartburn
pregnancy	738	0	0	0	0	0	0	0	0	0	1
miscarriage	38	246	1	0	1	0	0	1	0	0	0
abortion	31	5	235	0	0	0	0	1	0	0	0
anxiety attack	1	0	1	2,257	21	3	71	21	7	0	0
panic attack	0	0	0	0	603	0	0	0	0	0	0
insomnia	0	0	0	0	0	363	0	5	0	0	0
suicidal	0	0	0	0	1	0	424	38	0	0	0
depression	0	0	0	7	2	13	5	3,698	0	0	1
chest pain	0	0	0	0	1	0	0	1	490	0	0
acid reflux	2	0	0	0	1	0	0	4	2	266	3
heartburn	9	0	0	0	0	0	0	0	5	1	391

Figure 114. Confusion Matrix for BERTweet model for Experiment 2



	pregnancy	mental health	heart conditions
pregnancy	1,296	0	2
mental health	13	7,523	6
heart conditions	2	3	1,171

Figure 115. Confusion Matrix for BERT model for Experiment 3

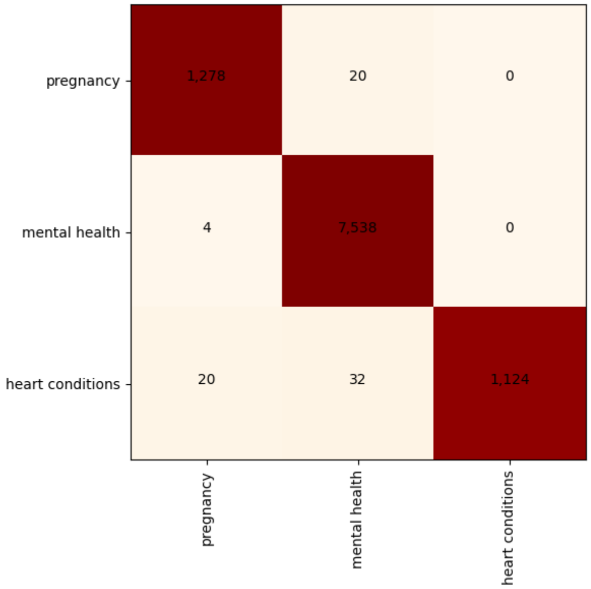


Figure 116. Confusion Matrix for BERT model for Experiment 4

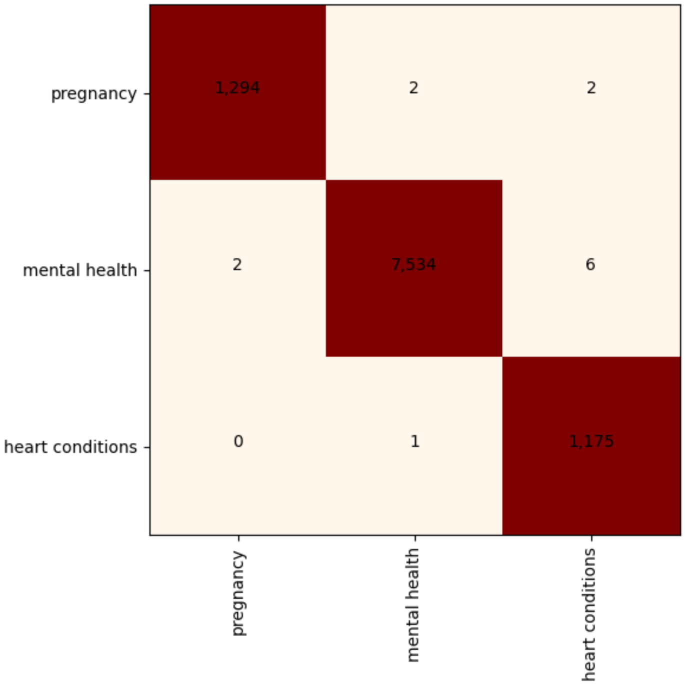


Figure 117. Confusion Matrix for BERTweet model for Experiment 3

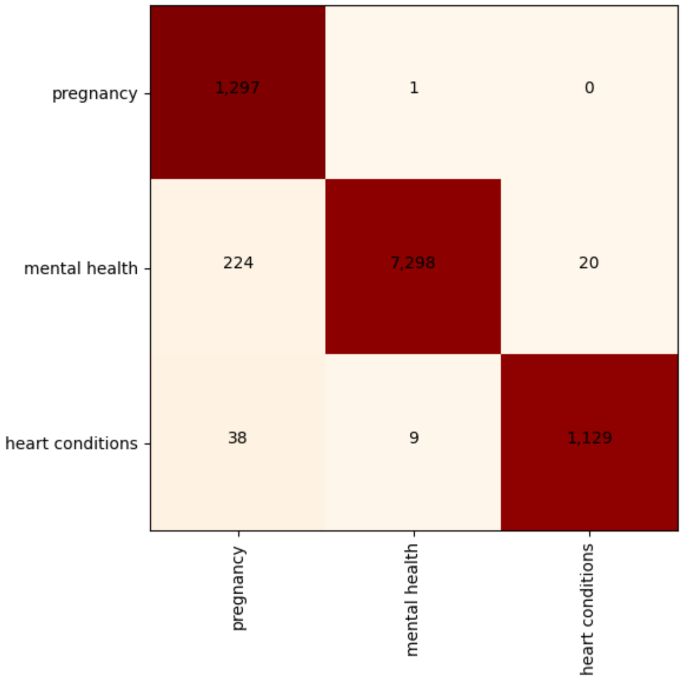


Figure 118. Confusion Matrix for BERTweet model for Experiment 4

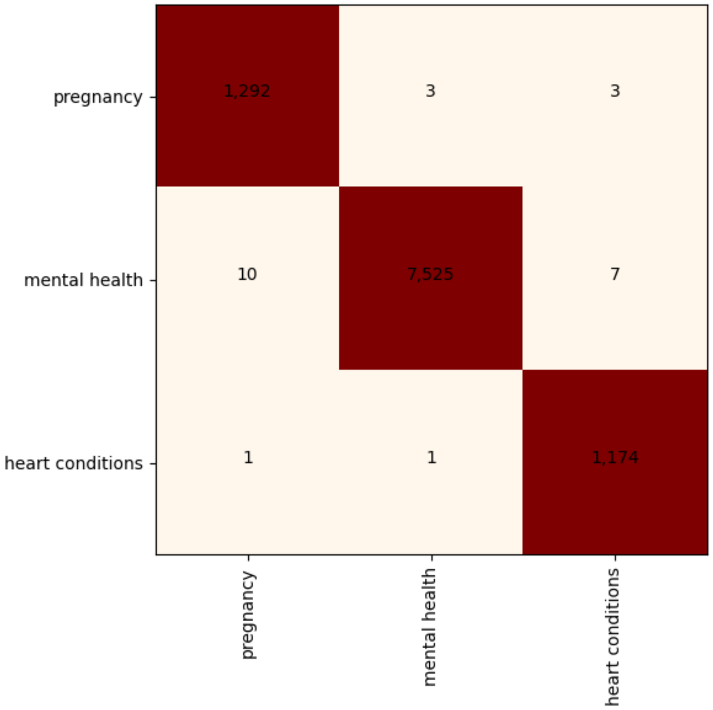


Figure 119. Confusion Matrix for RoBERTa model for Experiment 3

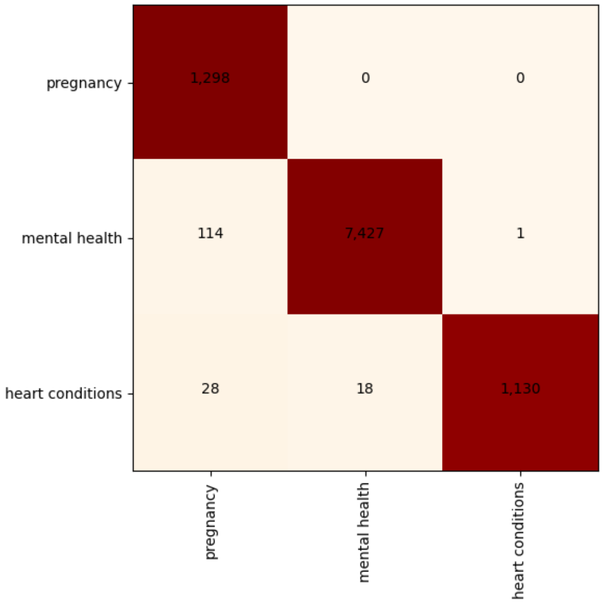


Figure 120. Confusion Matrix for RoBERTa model for Experiment 4