2020

Finetuning Pre-Trained Language Models for Sentiment Classification of COVID19 Tweets

Arjun Dussa
Technological University Dublin

Finetuning Pre-trained language models for sentiment classification of COVID19 tweets

Arjun Dussa

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Data Analytics)

September 2020

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: ***Arjun Dussa***

Date: ***1st September 2020***

ABSTRACT

It is a common practice in today's world for the public to use different micro-blogging and social networking platforms, predominantly Twitter, to share opinions, ideas, news, and information about many things in life. Twitter is also becoming a popular channel for information sharing during pandemic outbreaks and disaster events. The world has been suffering from economic crises ever since COVID-19 cases started to increase rapidly since January 2020. The virus has killed more than 800 thousand people ever since the discovery as per the statistics from Worldometer [1] which is the authorized tracking website. So many researchers around the globe are researching into this new virus from different perspectives. One such area is analysing micro-blogging sites like twitter to understand public sentiments.

Traditional sentiment analysis methods require complex feature engineering. Many embedding representations have come these days but, their context-independent nature limits their representative power in rich context, due to which performance gets degraded in NLP tasks. Transfer learning has gained the popularity and pretrained language models like BERT(bi-directional Encoder Representations from Transformers) and XLNet which is a Generalised autoregressive model have started overtaking traditional machine learning and deep learning models like Random Forests, Naïve Bayes, Convolutional Neural Networks etc. Despite the great performance results by pretrained language models, it has been observed that finetuning a large pretrained model on downstream task with less training instances is prone to degrade the performance of the model. This research is based on a regularization technique called Mixout proposed by Lee (Lee, 2020). Mixout stochastically mixes the parameters of vanilla network and dropout network. This work is to understand the performance variations of finetuning BERT and XLNet base models on COVID-19 tweets by using Mixout regularization for sentiment classification.

Key words: *sentiment analysis, pretrained language models, mixout, COVID-19, transfer learning, finetuning, BERT, XLNet, Twitter*

[1] https://www.worldometers.info/coronavirus/

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my project supervisor **Jack O' Neill,** for his guidance, encouragement, constructive suggestions, recommendations **and** support throughout the dissertation process. You are an amazing guide, an excellent teacher and a good person!

I would also like to thank **Dr. Luca Longo**, M.Sc. thesis coordinator, for his useful inputs in the formulation and design of research proposal.

Finally, love and best regards to my family and friends for being with me throughout my tough times and for providing me all the strength and courage to endure this journey.

TABLE OF CONTENTS

Contents

TABLE OF FIGURES

TABLE OF TABLES

LIST OF ACRONYMS

BERT	Bi-directional Encoder Representation of Transformers
CRISP-DM	Cross-Industry Process for Data Mining
NLP	Natural Language Processing
SVM	Support Vector Machine
CNN	Convolutional Neural Network
VADER	Valence Aware Dictionary for Sentiment Reasoning

1. INTRODUCTION

With a big user base of more than 160 million daily active users, Twitter has become one of most pervasive medium for micro-blogging and social networking today. Twitter is gaining popularity as a rich source for research for various social science and data science problems. There are successful implementations as a data source for Text analytics, sentiment and opinion mining, text classification, topic modelling etc. The use of such user-generated content is no longer limited to classical social media research and analysis but also has been effectively tried and tested in various different domains emerging these days, such as, disease tracking, modelling in epidemics, generating insights into the personalities of customers, news analytics, polls, predicting stocks and so on. The use of Twitter as a resource for extracting useful information during epidemic events is a challenging task, owing to the issues related with data quality and reliability of the posted content; it facilitates the preparation and planning of relief operations for outbreak tracking and management.

Jordan and his fellow researchers have helped with a review of research into how twitter is helping for outbreak tracking and surveillance purpose (Jordan et al., 2019). Ji et al., have published a paper about sentiment analysis of monitoring public health concerns using twitter sentiment classification with different techniques (Ji et al., 2013). Processing of social media messages during time and safety- critical situations help to reduce the risk of contamination during disease outbreaks, providing donations and volunteering services, coordinating media responses and arranging well- timed help to the people in affected areas. Analysing twitter feeds during these difficult times is easier and faster than other sources of information because of the real-time rapid transmission. Over the past few years, crisis response using social media information has gained so much popularity and an active area of research. Even twitter has created a new endpoint for easy access of COVID-19 outbreak tweets for easy analysis purpose.

1.1 Background

There is a long history signifying the use of Internet and Web technologies to gather and disseminate disease related information. During such events to facilitate stakeholders and disease control bodies, for planning and preparation of disease response. The Web has created unprecedented resources for tracking threats to public health. Ginsberg et al., relied exclusively on search engines to approach this problem, in which users could input queries in reference to issues they were concerned about. Their thread of research led to the realization that an aggregation of large numbers of queries might show patterns that are useful for the early detection of epidemics. Twitter, a micro-blog service provider shows several advantages over search engines for disease surveillance. It is up-to-date and there are more than 340 million tweets posted by 500 million Twitter users per day [2]. Most tweets are public, and the Twitter API enables researchers to retrieve the tweets as well as related information, such as geographical location and hyperlinks included. As a result, it has become a mainstream practice for the affected population and other concerned people to increasingly use social media platforms to post textual information as well as other useful multimedia content (images and videos) to express their emotions.

Corona Virus Disease or COVID-19 is a new virus disease that originated in Wuhan, China. The virus has now spread across the world and now almost all the countries are battling against this virus by trying their best to curb the spread as much as possible. The World Health Organization has declared it as a Pandemic and is leaving no stone unturned to control which is awaiting a vaccine to cure it. (El Zowalaty & Järhult, 2020)

Sentiment Analysis is also known as *opinion mining* or *emotional Artificial Intelligence* is based on the usage of Natural language Processing (NLP), text mining, computational linguistics to evaluate and examine the emotional states and subjective information. Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral" (A. & Sonawane, 2016). Over the years, people have posted their opinions, thoughts, or attitudes on social media platforms.

[2] https://en.wikipedia.org/wiki/Twitter

Twitter has enormous corpus of data. Analyzing these texts provide lots of useful information which can be applied in different domains.

Recent advancements in computational power has given opportunity to create many deep learning models and transformer based models which can capture most of the feature information from texts. By using transfer learning technique, a trained neural network can be used to fine-tune based on the specific task at hand. In this experiment, XLNet and BERT models used for sentiment classification of COVID-19 tweets. XLNet and BERT are pretrained language models based on transformers which pretrained on large unlabeled corpus.

In natural language processing, it has been observed recently that generalization could be greatly improved by finetuning a large-scale language model pretrained on a large unlabelled corpus. However, it has been observed that finetuning sometimes fails when there are less training instances (Lee, 2020). When finetuning a language model, dropout has been used as a regularization technique. The aim of this experiment is to apply a regularization technique called Mixout to both XLNet and BERT base models with and without sufficient training instances to understand whether finetuning works better with dropout or Mixout. It is basically a mix of Vanilla network and dropout network. (Lee, 2020). Mixout stochastically mixes the parameters of the two models' Vanilla network and dropout network. Vanilla network is the base network without any dropping of neurons. Dropout drops the neurons by certain percentage specified. *A dropout value of 0.5 indicates that 50% of neurons in the network will be temporarily removed.*

1.2 Research Project/problem

The main focus of this work is defined by the research question:

"To what extent finetuning Transformer based deep learning models like XLNet and BERT with Mixout can provide better accuracy results when compared to finetuning with Dropout when there are less training instances in a Multiclass sentiment classification using Twitter tweets on COVID-19?"

Research Sub-Question A - Is there any difference in classification performance of COVID-19 related tweets when finetuned with BERT and XLNet with dropout in a multiclass problem?

Research Sub-Question B - Does using mixout regularization technique to finetune BERT and XLNet improves classification performance when compared to Dropout regularization with enough training instances?

Research Sub-Question C - Does using mixout strategy instead of dropout regularization improves performance of multiclass classification when there are less training instances?

Research Sub-Question D - Which classifier performs best in terms of accuracy, precision, recall and f1-score for classifying COVID-19 tweets in both cases of training instances mentioned above?

Transfer learning has been widely used for the tasks in Natural language processing. Despite its success and wide adoption, finetuning a large pretrained model on a downstream task is prone to degenerate performance when there are less training instances. When finetuning a big, pretrained language model, dropout has been used as a regularization technique to prevent co-adaptation of neurons (Vaswani et al., 2017).

Co-adaptation means different hidden units in neural networks have highly correlated behaviour. This causes overfitting problem. Overfitting occurs when a function is fit too closely with some data points. In this experiment, a regularization technique called Mixout is used which is a combination of Vanilla network and Dropout network.

The aim of this research is to develop models and answer all the sub-questions mentioned above.

1.3 Research Objectives

The aim of the research is to do a multiclass sentiment analysis of collected Twitter tweets by finetuning pretrained language models such as BERT and XLNet with two different regularization techniques. The main objective includes finetuning the models with mixout regularization. The concept inspired from research conducted by (Lee, 2020) on Bert Large model for various datasets. Researcher has introduced this new

regularization strategy to improve the finetuning results of pretrained language models when there are less training instances. The paper says that when there are less training examples, mixout works better for large pretrained models. As part of this thesis, the same concept is applied on BERT and XLNet models with less training examples to understand the performance difference by comparing the results with models implemented by dropout regularization. In this regard, a null hypothesis is constructed suggesting no improvement in classification performance by applying mixout regularization on both the models. This is the hypothesis to be tested in this work. To be clearer, the aim is to determine whether mixout improves the classification performance of the mentioned models with less training examples and doesn't impact the performance of the same models with enough training examples.

Null Hypothesis: If Mixout regularization is used when there are less training instances to finetune pre-trained language models such as BERT and XLNet base models to address sentiment classification problem of twitter tweets on COVID-19, they cannot statistically outperform finetuning the same models with Dropout regularization on classification accuracy.

Alternate Hypothesis: If Mixout regularization is used when there are less training instances to finetune pre-trained language models such as BERT and XLNet base models to address sentiment classification problem of twitter tweets on COVID-19, they can statistically outperform finetuning the same models with Dropout regularization on classification accuracy.

The research objectives corresponding to each research sub-question are as described:

Research Objective A- Data Analysis to understand the sentiment variation for the data period.

Research Objective B- Perform finetuning of Bert and XLNet using dropout and mixout regularizations for complete data after pre-processing.

Research Objective C- Under sample the data and perform finetuning of Bert and XLNet using dropout and mixout for the reduced data sample.

Research Objective D- Compare and evaluate the performance of different models developed in objective B, C objective wise with precision, recall, F1 score and accuracy.

The resulting experimental tasks undertaken to achieve the research objectives are:

1. Extract and prepare COVID-19 dataset from Twitter for selected industry domains.

2. Assign the polarities for the extracted tweets after pre-processing.

3. Generate sentiment-based features using model tokenizers for BERT and XLNet and finetune the models.

4. Train and test the classification performance of both the models with Dropout and Mixout regularizations.

5. Observe the performance of BERT and XLNet classifiers on original data using performance metrics defined.

6. Under sample the data to reduce training examples and finetune the same models with dropout and mixout regularization techniques.

7. Train and test the models on under sampled data and observe the performance in terms of accuracy, precision, recall and f1-score.

8. Measure, analyse, compare and report the results of all the classification models performance in terms of dropout and mixout.

1.4 Research Methodologies

The research conducted in this project is *secondary* as it relies on the concept of mixout paper published by (Lee, 2020). Data required to fulfil the objective is extracted from social media network called Twitter by conducting some preliminary research about domains targeted, hashtags and account handles. According to the domains chosen, industry hashtags are taken to filter the data after extraction. The research is *quantitative* as it deals with statistical, mathematical and numerical analysis of data using objective measures.

The current research project involves multiclass sentiment text classification task where the text is labelled initially, and models developed to classify the tweet texts into *Positive, Negative* and *Neutral* categories. This is an attempt to examine the concept of *mixout regularization* technique on transformer-based models BERT and XLNet.

As the performance accuracies of different machine learning classifiers will be compared against each other using two different regularization techniques, the obtained results are verifiable by observation rather than purely by logic or theory. This research is *empirical* in nature as it focuses on testing the feasibility of the suggested solution using empirical evidence. This research follows a *deductive* approach as it starts with a proposed theory, progresses to a hypothesis and ends with a rejection or acceptance of the hypothesized solution.

The research methodology broadly follows Cross-Industry Process for Data Mining (CRISP-DM) which is a well-known methodology. In this context, CRISP-DMs *Business Understanding* phase can be considered similar to the *Literature Review* covered in Chapter 2. The *Data Understanding*, *Data Preparation* and *Data Modelling* phases of CRISP-DM are covered in Chapter 3 under *Design and Methodology*. Chapter 4 covers *Results, Model Evaluation and Analysis* which is *Model Evaluation* in CRISP-DM. Lastly, the end of the CRISP-DM cycle, Deployment phase corresponds to the *Discussions and Conclusions* which are outlined in Chapter 5.

1.5 Scope and Limitations

The scope of this research is strictly limited to the examination of changes in text classification performance of finetuning Pretrained language models *BERT base* and *XLNet base* using Dropout and Mixout regularization techniques with original and under sampled dataset. Dataset under sampling is done by using RandomUnderSampler from Random Sampler package in python. While doing this, 3000 instances for each class are selected to reduce the training examples. Finetuning models with complete data prepared is to verify if there is any performance impact in the classification of tweets when finetuned BERT and XLNet with enough training examples when dropout regularization is used. Finetuning models with under

sampled data is to check whether there is any improve in performance of the classifiers when finetuned the models with mixout regularization with less training examples. The performance of the classifiers is evaluated in terms of *Precision, Recall, F1-Score* and *% Accuracy* of Correctly Classified Tweets.

Although extensive study has been conducted to extract the data, there are chances to miss important tweets as we have limited the data per day to 1K tweets to cover the maximum covid period. No attempt is made to tune hyperparameter values as it is suggested by model developers to use the same hyperparameter values for finetuning. BERT and XLNet models are taken because of the growing popularity and the results it has produced on various NLP tasks such as document ranking, sentiment classification, language generation etc. It should be noted that finetuning of the models is performed on twitter data collected and labelled using polarity scores generated by *Vader Analyzer*. Performance of the Vader scores are verified by taking random sample from the cleaned dataset, labelled them manually as Positive, Negative, Neutral and verified with results achieved by Vader. Although verification is done by taking random sample, there is no way to guarantee the quality of results generated as there is only one person included in labelling process. The accuracy of the results obtained thus may depend on the quality of the results achieved during labelling.

1.6 Document Outline

There are four chapters remaining in this report. Below presented an outline of the content covered in each chapter ordered by the chapter number:

Chapter 2- Literature Review: This chapter provides a comprehensive literature review of previously conducted researches on social media during outbreaks, Sentiment analysis approaches, Sentiment Analysis using social media data, transfer learning, finetuning pretrained language models for sentiment analysis, performance metrics for evaluating deep learning models and gaps in the research.

Chapter 3- Design and Methodology: This chapter provides insight into the experiment that was conducted, in order to test the hypothesis and eliminate the gaps defined in

Chapter 2. It underpins an inclusive clarification to the design process of the experiment and methods to evaluate the performance of the proposed technique and compare the developed models.

Chapter 4- Results, Evaluation and Discussion: The results of the experiment are presented here, and the performance of different models with regularizations applied are evaluated and compared. Design flaws that led to inaccurate results and possible improvements that may guide to build a better model will be discussed.

Chapter 5- Conclusion: In this chapter, the results, observations and insights gathered throughout this investigation is summarized, further research that can be carried out as a potential extension to this thesis is presented.

2. LITERATURE REVIEW

Sentiment Analysis of social media channels such as Twitter are an active form of communication channels during pandemic events, natural disasters and daily news. Research suggests that a thorough analysis of social media content could turn out tremendously useful to predict sentiments and panic during outbreaks and the psychological effects on people. This could help government bodies to take necessary actions to prevent further spread of the negative emotions.

Extracting useful information from social media messages involves various processing stages like filtering, parsing, ranking, classifying, summarizing etc, depending upon the nature of the task. Using this textual information posted as tweets have certain challenges, which includes information gathering and classification. This is because of the limited number of words the platform has defined for posting, irregular structure and presence of additional noise. This causes significant drop in the performance of the classification models due to different slangs, misspellings, hashtags, abbreviations, URLs, sarcasm, improper language usage, emojis and emoticons (Dubey, 2020). Machine learning has evolved to handle most of the issues in text processing in natural language. There are different state-of-the-art machine learning techniques including supervised, semi-supervised and un-supervised techniques.

2.1 Social Media during Outbreaks

With the rise of the participatory web and social media (‘‘Web 2.0’’) and resulting proliferation of user-generated content, the public potentially plays a larger role in all stages of knowledge translation, including information generation, filtering, and amplification (Chew & Eysenbach, 2010). Consequently, for public health professionals, it is increasingly important to establish a feedback loop and monitor online public response and perceptions during emergency situations in order to examine the effectiveness of knowledge translation strategies and tailor future communications and educational campaigns. Twitter has become popular since H1N1 outbreak which was the first global pandemic in the social media era. Chew & Eysenbach Used an ‘‘infoveillance’’ approach to report on: 1) the use of the terms ‘‘H1N1’’ versus ‘‘swine flu’’ over time on Twitter, to establish the feasibility of creating metrics to measure the

penetration of new terms and concepts (knowledge translation), 2) an in-depth qualitative analysis of tweet content, expression, and resources, and 3) the feasibility and validation of using Twitter as a real-time content, sentiment, and public attention trend-tracking tool.

The journal published by International society of travel medicine (*The Pandemic of Social Media Panic Travels Faster than the COVID-19 Outbreak*, 2020) talks about how panic spread happened during COVID19 pandemic outbreak. The study says that the impact of media reporting and public sentiments may have a strong influence on the public and private sectors in making decisions on discontinuing certain services including airline services, disproportionate to the true public health need. Analyses of discussions on social media with regard to the epidemic situation geographically and over time can result in real-time maps. Such real-time maps could then be used as a source of information on where to intervene with key communication campaigns.

Chew & Eysenbach published a paper presenting the facts about how social media is trending nowadays to predict and track disease outbreaks. Research also provides information on how media, cell phones and other communication channels have opened up a two-way street in health search, supplying not just a portal for information delivery to the public but also a channel by which people reveal their concerns, locations, and physical movements from one place to another (*A 31*). This study illustrates the fact that this two-way street is transforming disease surveillance through which health officials can respond to disasters and pandemics. But it's also raising hard questions about privacy and about how data streams generated by social-media and cell-phones might be made available for health research by improving surveillance (*A 31*).

Mollema,2015 et al., conducted a research on measles outbreak began in Netherlands in May 2013. This research is about comparing number of messages expressed on twitter and other social media during the measles outbreak by considering number of new articles and reported cases to check public opinion patterns vs disease patterns. Research analysed the content of the messages (i.e., topic) and how the messages were expressed (i.e., sentiment) by using title for determining the topic and sentiment for each data source. If this was not clear or did not match with the summary, then the summary was

used for determining the topic and sentiment. The research has concluded that during the measles outbreak, 3 large peaks in the number of messages with a small width were observed for all 3 types of online media data, which coincided with announcements about the measles outbreak by the RIVM and statements made by well-known politicians.

2.2 Analysing sentiments from Twitter Texts

Sentiment Analysis is the broad task of assigning sentiment-class labels to a given text in consideration with an aim to generate polarity of the opinion expressed by it. The text mostly derives from social media websites, blogs, and product reviews etc. The task of analysing sentiments in each piece of text is also commonly known by the name, opinion mining, and is employed to analyse people's sentiments, attitudes and opinions about different things and entities. There is a constant upsurge in studies related to sentiment analyses due, in part, to the advancement and popularity of machine learning approaches for natural language processing, computational linguistics, information extraction and retrieval as well the ready access to massive and open-utility social media datasets, making sentiment analyses one of the most favoured research domain for social media (A. Kaur, 2019b). Sentiment analysis can be broadly categorized into three main levels on the basis of their depth of operation. These are: *Document Level*, *Sentence Level* and *Entity or Aspect-Level* as mentioned in (Farra et al., 2010; A. Kaur, 2019a; Sharma et al., 2014)

Document Level: The task at this level is classifying sentiments for the entire document. It is important to note that for this type of analysis, the documents should correspond to a single topic, multiple topics can't be accommodated in this case as this level assumes document singularity for its operation.

Sentence Level: This provides a detailed sentence-level analysis for each line in the document. Each sentence is evaluated to determine the polarity of opinion expressed by it ranging from negative to positive. Neutral class may or may not be included for a sentence.

Entity or Aspect Level: Aspect level or entity level deals with each entity that

a sentence talks about. It can be thought of as contextual sentiment analyses as it needs to have an understanding of how many entities a sentence has and what kind of sentiment words (adjectives or adverbs to denote their quality) are being used. A single sentence might have two totally unrelated entities with opposing opinions. As an example considers the sentence: "This book is brilliant but is too lengthy to read". There are two aspects in this case with differing sentiment polarities. Aspect level sentiment analyses are more detailed in approach and thus can be highly reflective of the sentiment expression but is complicated and can vary significantly across domains. Again, the sentiment word "frightening" will be positive for a movie review (horror genre) but when used in context of a product review, say, a car, it totally changes the connotation and meaning. Thus, domain adaptability is one of the main limitations of this finer level sentiment analysis approach.

Sentiment analysis can be performed in a number of ways depending upon the domain, type and nature of text and possible applications. In a review article by (Beigi et al., n.d.), sentiment analysis is classified into two groups - *language processing based* sentiment analysis and *application-oriented* sentiment analysis.

Language Processing Based Sentiment Analysis - This group includes sentiment dictionaries (also called lexicons) to perform the sentiment analysis. It makes use of grammar constructs and rules of language words and semantics to properly classify a sentence into a positive or a negative class. Lexicons can be generated based on a language dictionary or a domain-specific corpus. Dictionary-based approaches are more comprehensive and exhaustive as they involve bootstrapping while corpus-based approach is a bit restrictive and non-transferable to other domain areas. Sentiment lexicons are known to improve the performance of polarity and subjectivity classification for sentences in a given text. (A. Kaur, 2019b)

Application-Oriented Sentiment Analysis - This group deals with the application area where the sentiment analysis is applied. Due to the massive available of online information from social media, several application-oriented sentiment analysis tasks have been performed including classifying movie and product reviews, App reviews, for predicting stock market and customer trends on the basis of their likes and dislikes of

certain items. A wide range of tools are available which perform application-oriented sentiment analyses while machine learning techniques like SVM, Naive Bayes, Maximum Entropy etc. are equivalently popular choices. (Pagolu et al., 2017)

2.2.1 Sentiment Analysis Approaches

There are three main techniques in sentiment analysis. Lexicon approach, Machine learning based approach and Hybrid approach. A brief description is given below.

Lexicon Approach: A dictionary of pre-tagged lexicons is used in this approach. The dictionary can vary across different applications. This works on simple principle: Input text is taken to break it into tokens using a certain token sequence (uni-gram, bi-gram, word-level etc.) and match every token with the contents of the dictionary. Scoring of the token will be done if there is a match found, else generate no score for a given token. In the same way, one can have polarity based lexical analysis. Instead of calculating the sentiment scores, this approach only looks for a match of a token into either of the two classes – positive list and negative list and classifies the incoming token sequence on the basis of the number of matches found in the text. This simple approach has the capability to produce very good quality sentiment classification results. This is one of the earliest approaches to sentiment classification and where it could reach an accuracy of 80% on single phrases using adjectives. (Sadia et al., 2018)

Machine Learning based Approach: This approach could produce high level of accuracy and it has good domain- adaptability. This might be the reason why this technique is favoured. In case of labelled sentiment datasets, the supervised machine learning classifiers are one of the choicest methods to perform sentiment analysis. It is possible to use uni-grams, bi-grams and tr-gram sequences as feature vectors corresponding to single word, two consecutive and three consecutive word phrases respectively. In a case where more adjectives or adverbs are expected, higher order n-grams are useful. Also, the significance of bigrams increases in case of negations and indirect word references. Example, if using a unigram, the sentence 'This is not good' might be classified as positive because of the word 'Good', however, using bigrams, 'not good' is classified as negative sentiment. Most common supervised machine learning techniques employed for sentiment classification include SVM(Support Vector Machine), Naïve Bayes,

Random Forest, etc. Accuracy between 60%-80% is observed for classification using these supervised techniques. The main challenges in designing a classifier in this case depend on the availability of training data, contextual understanding of the word phrase and its surroundings as well as the size of the data corpus. (Caramanis & Barber, 2017.; Elbagir & Yang, 2018a; Li et al., 2020; Shelar & Huang, 2018)

Hybrid Approach: Hybrid approach brings the best of both the previous approaches – lexicon approach and machine learning based approach to enhance the capabilities of the classifiers. These have high accuracy and speed. Take any base classifier like Naïve Bayes, Random Forest, SVM and couple it with lexical component to build a hybrid scheme of sentiment analysis. Several algorithmic approaches have been tried and tested in Twitter to conduct sentiment analyses. A study on comparison of algorithms for twitter sentiment analyses (Caramanis & Barber, 2017) suggest that weighted combination of predictive models yield a higher accuracy than any one method alone.

2.3 Sentiment Analysis of Social Media data

So many techniques for sentiment analysis have been in place. Over the couple of years, Twitter has become the popular source for sentiment classification tasks. Researchers have tried implementing various machine learning deep learning models with different approaches. Here, we will discuss a few researches related to the task undertaken.

In recent years, a lot of work has been done in the field of *Sentiment Analysis* by number of researchers. In fact, work in the field started since the beginning of the century. In its early stage it was intended for binary classification, which assigns opinions or reviews to bipolar classes such as positive or negative. Paper (Turney, 2002) predicts review by the average semantic orientation of a phrase that contains adjective and adverb thus calculating whether the phrase is positive or negative with the use of unsupervised learning algorithm which classifies it as thumbs up or thumbs down review (Elbagir & Yang, 2018a).

Paper (Pagolu et al., 2017) conducted a research about sentiment analysis of twitter data to predict stock market movement. They have used Word2vec and N-gram for analysing the sentiments in tweets and related the stock movement with the company sentiment in

tweets. This is an example of correlation analysis of price and sentiment. The accuracy achieved with Word2vec and N-gram applied to Random Forest classifier is approximately same.

While Alsaeedi conducted research about different approaches followed for sentiment analysis, Dubey implemented lexicon-based approach to categorize sentiments. His study was more focused on representing word count for each country. (Alsaeedi & Khan, 2019; Dubey, 2020)

The research published by Sailunaz Alhaji conducted emotion and sentiment analysis with twitter data with a slight a difference. They have included tweet replies and introduced agreement score, sentiment score and emotion score to analyse. Annotated text as per the emotions and sentiments has been given as input to Naïve Bayes model. Further, text based parameters were merged with user-based parameters to detect influential users which helped to develop a recommender system. (Sailunaz & Alhajj, 2019)

The research conducted by two other papers illustrates topic modelling. The aim of the study was to understand what people are discussing during COVID-19 crisis. They have Implemented LDA (latent Dirichlet Allocation) algorithm for topic modelling.
(Abd-Alrazaq et al., 2020; Medford et al., 2020)

Cai(2013) published a paper on sentiment classification of tweets using very deep convolutional neural networks and Google BERT on Sentiment140 dataset. For very deep CNNs the models were trained using Glove embeddings dataset. The second task was finetuning BERT model for Sentiment140 dataset. Very Deep CNNs developed here with Glove embeddings has got approximately same results as BERT model which is bit surprising. (Cai, 2013). Pota et al., conducted a research on political tweets using deep learning techniques. The research approach was to represent the text by dense vectors comprising sub-word information to better detect word similarities by exploiting both morphology and semantics. CNN model is implemented to do the classification. (Pota et al., 2018)

The paper Ruangkanokmas et al. (2016) implemented deep belief networks using chi-squared based feature selection. As the features not required are filtered from the vocabulary, the efficiency of the networks increased. The experiment claims that this method could achieve higher accuracy results and can speed up training time when compared to other semi-supervised learning algorithms.

The paper (Hao et al., 2011) talks about the research conducted on visual sentiment analysis of twitter data streams. This research was more focussed on handling high-volume twitter data. The paper introduces three novel time-based sentiment analysis techniques. (1) topic-based sentiment analysis that extracts, maps, and measures customer opinions; (2) stream analysis that identifies interesting tweets based on their density, negativity, and influence characteristics; and (3) pixel cell-based sentiment calendars and high density geo maps that visualize large volumes of data in a single view. We applied these techniques to a variety of twitter data, (e.g., movies, amusement parks, and hotels) to show their distribution and patterns, and to identify influential opinions.

Due to the advancement in computational power and high performance results of deep learning models based on transformers, researchers have looked beyond distributed word representations (Glove, Word2vec etc) for effective sentiment analysis with transfer learning technique (Section 2.4) to finetuning pretrained language models such as BERT, XLNET, FastBERT, GPT etc. Distributed word embedding models lack contextual information. Most of such sentiment tasks are into finetuning models for Aspect based sentiment analysis, Target dependent sentiment classification and domain adoptability (Gao et al., 2019; Rietzler et al., 2019; Sun et al., 2019). Aspect specific analysis generally involves, adding a neural network layer or recurrent neural network layer on top of pretrained language model embedding layer. The obtained token representations can be directly fed to the neural network layer to get the softmax probabilities. Domain adaptation generally involves finetuning pretrained models on a dataset related to a different domain and testing on some other domains. Through this generalizability can be improved (Rietzler et al., 2019).

Transfer learning has led researchers to pursue further and develop different techniques using the pretrained language models. Examples are SentiLR, BroXLNet, SentiBERT (Gong et al., 2019; Ke et al., 2019; Yin et al., 2020). SentiLR introduces word level linguistic knowledge including part-of-speech tagging and prior sentiment polarity from SentiWordNet. A paper (Gong et al., 2019) talks about the lack of capturing broad features in sentence level representation. The research proposes a new model which incorporates broad learning system to capture deep contextual features and randomly searching high-level contextual representation in broad spaces. Results achieved using this method did beat state-of-the-art algorithms like BERT, XLNET etc. in sentiment analysis.

2.4 Methodology based on Machine learning Algorithms

Machine learning is considered as a branch of Artificial Intelligence, which enables computers to learn from past experiences without any human need. There are mainly four different categories of Machine learning Algorithms as below:

Supervised Learning: This category requires labelled input data for the model to learn. This is generally used when there are set of input variables and output variable then, the algorithm is used to learn the relationship between the input and output. The task it to find the approximate the mapping function so that the model can predict for a new set of input. Examples are Naïve bayes, Random Forest etc.

Unsupervised Learning: This type of learning is used when there is no defined output variable. The aim is to find the patterns in the data. Example clustering.

Semi-Supervised Learning: This learning is used when there is large amount of input data but only some of the data is labelled.

Reinforcement Learning: This method focusses at using data collected by interacting with environment and then actions will be taken to minimize or maximize the error. This leaning continues until the algorithm explores all the possible range of values.

2.5 Deep Transfer Learning for Natural Language Processing

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. [3]

It has become a popular approach in Deep learning where pretrained models are used as the starting point to finetune the model for the secondary task. Given the compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems. [4]
In deep learning, the form of transfer learning used is called as inductive transfer. The scope of possible models (model bias) is narrowed here in a profitable way using a model fit on a different related task.

Predictive modelling has two common approaches here. A) Develop Model Approach, B) Pre-trained Model Approach. Figure 1 – 2.4.

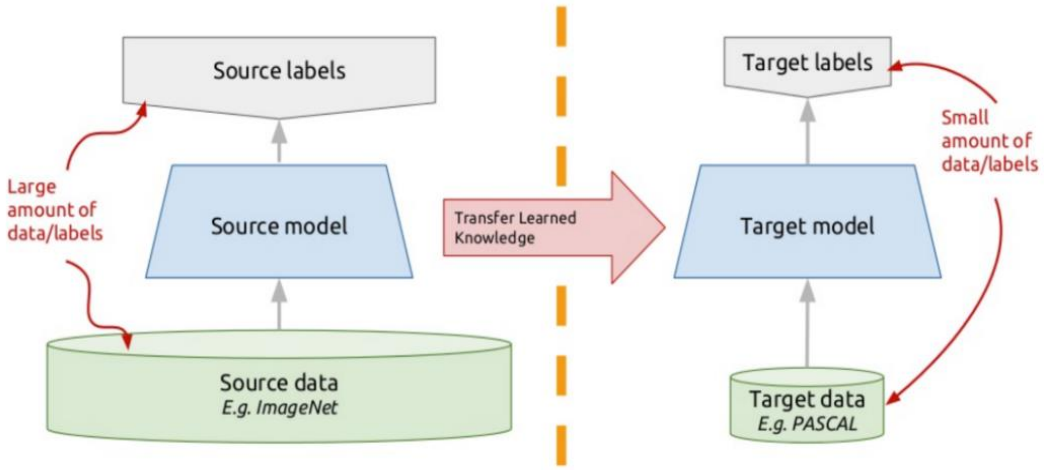

Figure 1- 2.4 Basic flow of Transfer learning [5]

[3] https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[4] https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[5] https://medium.com/the-official-integrate-ai-blog/transfer-learning-explained-7d275c1e34e2

2.5.1 Develop Model Approach

Source Task selection: A predictive modelling problem is selected according to the input data and output needed. Also, there should be some relationship between input and output data.

Develop Source Model: Develop a skilful model for this first task. This model should be better than the naïve model. This is to ensure that some feature learning has been performed.

Model Reuse: The Model fit on the source task now can be used as the basis for a model on the second task of interest. This could sometimes involve all or some parts of the model, depending on the modelling technique used.

Model tuning: Sometimes, the model may need to be adapted or refined on the input-output pair data available for the second task of interest. [2]

2.5.2 Pre-trained Model Approach

Source model selection: Here, a pre-trained source model is selected from the available models. Mostly, Research institutions release these models on large and challenging datasets.

Model Reuse: Then that pre-trained model can be used as the starting point for the second task of interest. Similarly, this may include full or parts of the model.

Tune Model: Depends on the task, source model may need to be adapted or refined for the task of interest. [6]

This Pre-trained model approach is common in deep learning field. Examples of such models are Bert and XLNet, Word2vec, Glove etc. There are so many benefits for using transfer learning. Some of them are higher start, high rate of improvement of skill, better converged skill.

[6] https://machinelearningmastery.com/transfer-learning-for-deep-learning/

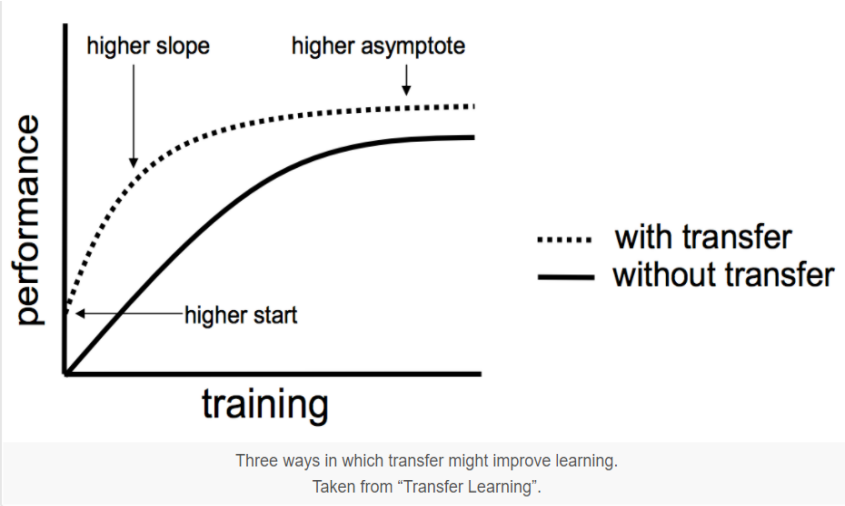

Figure 2 - 2.4 Transfer learning benefits [7]

2.5.3 BERT

BERT (Bidirectional Encoder Representation from Transformers) is a paper (Devlin et al., 2019) published by Google AI Language. This model has received good reviews in the machine learning community by giving state-of-the-art results on a variety of NLP tasks, including sentiment analysis, question answering, natural language inference etc.

BERT makes use of an attention mechanism in transformer that learns contextual relations between words in a text. Transformer has two different mechanisms – an encoder that reads the text input and a decoder that produces the prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary. [8]

First 15% of the words in each sequence are replaced with a [MASK] token before feeding input sequences to BERT. The model then attempts to predict the original value

[7] https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[8] https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

of the masked words, based on the context provided by the other, non-masked words in the sequence. In other terms, the prediction of the output words requires:

1. Adding a classification layer on top of the encoder output.
2. Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
3. Calculating the probability of each word in the vocabulary with softmax function.

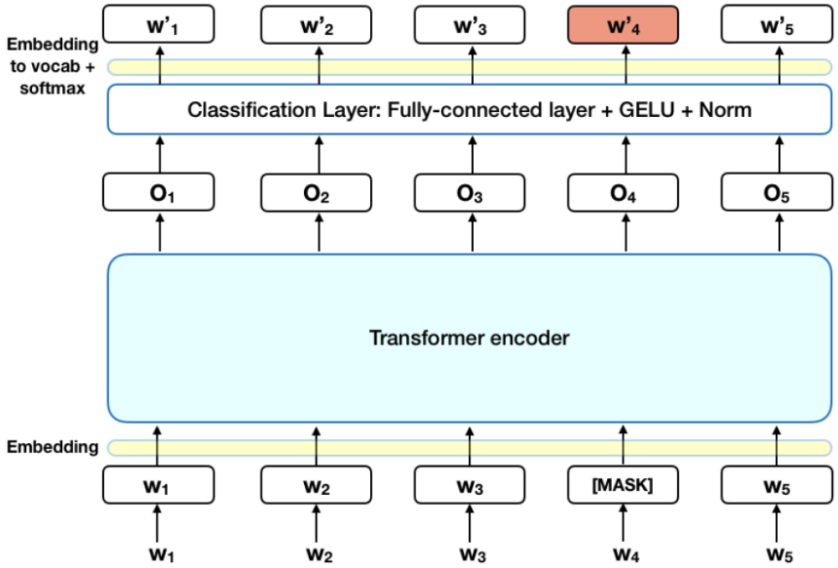

Figure 3 - 2.4.3 BERT model [9]

Softmax is a function that turns X real values into a vector of X real values whos sum is equal to 1. Irrespective of the input type, it transforms them into values between 0 and 1 so as to interpret as probabilities.

Bert loss function considers only the prediction of masked values and ignores the prediction of non-masked words. The model converges more slowly than directional models because of this, a characteristic which is offset by its increased context awareness.

[9] https://machinelearningmastery.com/transfer-learning-for-deep-learning/

BERT has 2 versions: Base and Large comes with cased and uncased [10]. Cased model is trained on english case data. Where as uncased model is trained on lower-case data.
During finetuning for sentiment classsification, a classifier layer is added on top of the transformer output for the [CLS] token. Chaper 3 Section 3.8 Modelling has the details of finetuning performed as part of this work.

2.5.4 XLNet

XLNet is a generalised autoregressive pretraining method. XLNet is Bert like model with some differences. **AR language model** is a kind of model that using the context word to predict the next word. But here the context word is constrained to two directions, either **forward or backward. [11]**

BERT masks the words and assumes that the masked words are independent of each other. It doesn't consider the dependency between the masked words. This is the disadvantage Bert. This is where XLNet comes into picture. XLNet uses permutational language modelling technique. It means, XLNet considers all possible permutations so that it can cover both forward and backward directions.

XLNet makes use of a *permutation operation* during training time that allows context to consist of tokens from both left and right, capturing the bidirectional context, making it a generalized order-aware AR language model. Simply put it, XLNet keeps the original sequence order, uses positional encodings, and relies on a special attention mask in Transformers to achieve the said permutation of the factorization order.
XLNet uses two-stream self-attention mechanism to keep a track of predicted words and consider them in the next token prediction. (Yang et al., 2019)

---

10 https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

11 https://towardsdatascience.com/what-is-xlnet-and-why-it-outperforms-bert-8d8fce710335

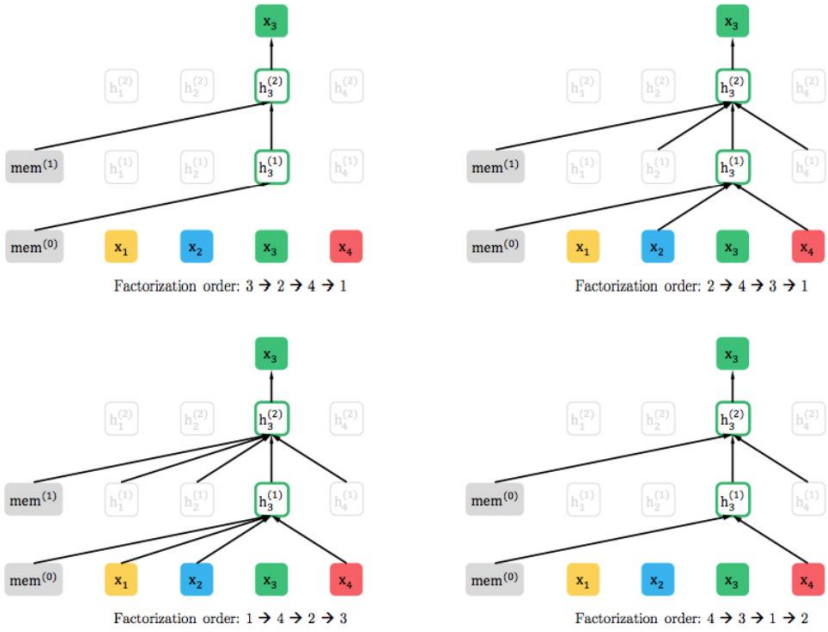

Figure 4 - 2.5.4 XLNet factorization

Similar to the Bert finetuning mentioned in section 2.4.3, a classifier layer is added while finetuning the model either base or large, then output of the last [CLS] token is taken to compute logits. Logit is any function which maps probabilities [0,1] to [-inf, inf]. Softmax is a function that turns a real valued vector into a vector of real values where the sum equals to 1.

For both BERT and XLNet, ADAMW optimizer is recommended by the authors (Devlin et al., 2019; Yang et al., 2019). An *Optimizer* is an algorithm or method used to change the attributes of the neural network such as weights and learning rate to reduce the losses. *Cross Entropy Loss function* measures the performance of a classification model which outputs the probability values between 0 and 1. Cross Entropy Loss increases as the predicted probability diverges from the actual label.

2.6 Mixout- Effective Regularization

Mixout is a regularization strategy proposed in the research (Lee, 2020). The basic idea behind this is, it stochastically mixes the parameters of Vanilla Network and Dropout Network with a probability specified. Vanilla Network is the network without any

dropping of neurons. When the dropout value is specified, number of neurons as per the value (percentage) specified will be temporarily dropped.

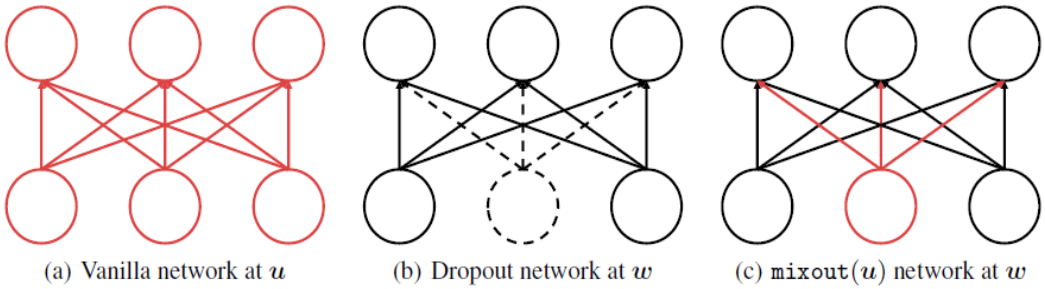

Figure 5 - 2.6 Mixout Network (Lee, 2020)

The process followed by the authors to create a mixout network is as below.
First the parameters of vanilla network were memorized. Then, in the dropout network, they randomly chose an input neuron to be dropped (b) with a probability of value p. It means, all the outgoing parameters of the dropped neuron are eliminated. Then eliminated parameters from network b are replaced with corresponding parameters in Vanilla Network (a). [12]

2.7 Gaps in the Literature

Even though there are multiple implementations of using pretrained language models such as BERT, XLNet, ROBERTA, GPT to finetune for specific task, the research is lacking using different regularization techniques. Most of the research into sentiment analysis has been performed either by machine learning models, distributed word embeddings for better accuracy results, there are still fewer researches into applying transfer learning techniques for various other tasks like pandemic outbreaks and natural disasters. Mostly importantly, there is not even a single research into implementing Mixout regularization for finetuning sentiment analysis except the concept proposed in the paper (Lee, 2020). COVID-19 has endangered human lives for the past 8 months and created economic crisis and unemployment. It is vital for the economic survival of the world to understand how sentiments vary during such crisis situations. The use of

[12] https://github.com/bloodwass/mixout

sentiment analysis during these pandemic outbreaks helps institutions, healthcare and government bodies to take proper policy measures and plan next course of actions. This aim of this work is to do the sentiment classification using Dropout and Mixout regularization techniques to understand the performance difference of finetuning Pretrained language models; BERT and XLNet on COVID-19 tweets related to industries Pharma, Healthcare, Airlines etc.

3. DESIGN AND METHODOLOGY

This chapter discusses the underlying project approach and detailed design aspects of the experiments conducted as a part of this study. This also includes the statistical treatments of the experimental results produced. An overview of the experimental design, specifications of hardware and software used, documentation of the data source and contents is also provided.

3.1 Project Approach

The aim of the current research is grounded in measuring the classification performance of twitter dataset consists of COVID19 tweets related to selected industries by finetuning BERT(Bi-directional Encoder Representations from Transformers), XLNet which is a Generalized Autoregressive pretrained model with two different regularization techniques called Dropout and Mixout.

Dropout is a regularization technique for neural network models proposed by Srivastava et al. In dropout technique, randomly selected are ignored during training. Means, their contribution is removed to the activation of downstream neurons temporarily on the forward pass and weight updates are not applied to the neuron on the backward pass. This is a common regularization strategy being followed to avoid overfitting of the model.

Mixout is a regularization strategy proposed by Lee(Lee, 2020) which works by mixing the parameters of vanilla network with dropout network with some probability value specified. Section 2.6 has detailed explanation of the Mixout network.

The overall project can be divided into four main tasks. Understand the sentiment variation for the selected industries as a whole during COVID19 period from Jan to June, Second; finetune pretrained language models BERT and XLNet with a single classifier layer with Dropout and Mixout techniques, third; under sample the dataset by using RandomUnderSampler to reduce number of training instances and balance the dataset and finetune BERT and XLNet models in the same way with Dropout and Mixout

techniques. Fourth; Compare the performance of the models in each case with regularization change and data size.

The performance differences in the classification performance using dropout and mixout regularization strategy are measured by Accuracy, Precision, Recall and f1-score. These metrics are used to analyse the performance of each model and compare wherever needed to fulfil the overall objective as given in Section 1.3.

- Is there any difference in classification performance of covid19 related tweets when finetuned with BERT and XLNet with dropout in a multiclass problem?
- Does using mixout regularization technique to finetune BERT and XLNet improves classification performance when compared to Dropout regularization with enough training instances?
- Does using mixout strategy instead of dropout regularization improves performance of multiclass classification when there are less training instances?
- Which classifier performs best in terms of accuracy, precision, recall and f1-score for classifying covid19 tweets in both cases of training instances mentioned above?

3.2 Design Aspects

The overall system can be viewed as four-entity process decomposed into BERT and XLNet finetuning with Dropout, BERT and XLNet finetuning with Mixout and repeat the experiment with under sampled data.

The experimentation was undertaken using free **Google Colab Tesla T4 GPU** which has **12GB RAM**.

Using twitter, extracted tweets and then raw tweets are pre-processed and cleaned using python. This includes removing urls, expanding contractions, removing hashtags and account handles, utf8 special characters removal etc. Then cleaned tweets are used to assign sentiment scores by using Vader Analyzer and Textblob. After deciding on sentiment scores using uniform distribution check and manual verification for a few,

performed model generation. It means finetuning of BERT and XLNet with dropout and mixout techniques with the complete data and then using RandomUnderSampler reduced training instances and repeated the experiment for the same models. It is important to note that only 3000 instances for each class are selected to reduce the number of instances in the data and to balance the classes in the target. Section 3.3 covers more about the details of each process. Figure 6 shows the design diagram for the experiment.

Figure 6 - 3.2 Design diagram

Performance metrics used are *Precision, Recall, F1-score* and *Accuracy* to evaluate model performance in each case. For the models in the 1st case which have used imbalanced data (complete dataset), precision, recall and f1-score are main metrics. Whereas for the models with under sampled data, accuracy is the main measure.

3.3 Detailed Design and Methodology

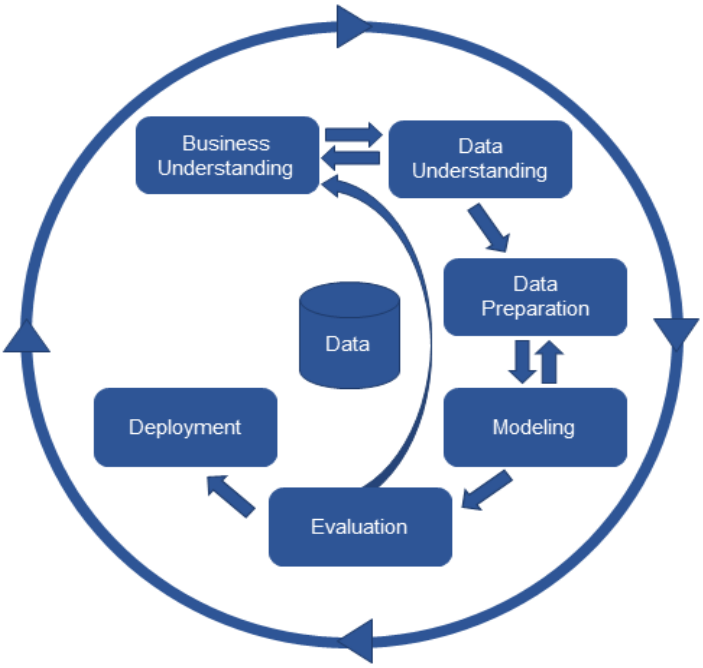

Figure 7 - 3.3 CRISP-DM methodology

This section provides a detailed methodology based on the CRISP-DM (Cross Industry Standard Process for Data Mining) process model as shown in Figure 7 – 3.3. The CRISP-DM process model provides a structured approach to planning and designing a data mining project as well as organizing the experimental set-up.

Chapters 1 & 2 account for the business understanding part. That involves understanding the research objectives and requirements from a business perspective which includes steps such as, refining the research objectives into a specific data mining problem definition and specifying the data mining goals and success criteria. The focus of the current chapter, however, is on devising a preliminary plan to achieve the objectives by outlining a step-by-step action plan for the project as well as initial assessment of the tools and techniques. This is done after reviewing the available data, also called Data Understanding. This involves gathering data, describing, exploring it and most importantly, verifying the data quality. Data preparation covers the cleaning process. Then modelling of the selected models is done followed by evaluating results and providing inputs for the future researches. This concludes by reviewing and reporting results and outputting the deliverable, also called Deployment. The Data

Modelling, Evaluation and Deployment stages are covered in Chapters 4, 5 and 6 respectively of this report.

3.4 Data Description

As part of the research, the dataset used during sentiment classification process plays a very important role, as it can significantly impact the classification performance. According to the review of state-of-art approaches in the field of sentiment classification, the selection of the sentiment classification dataset depends on many factors, the objective of the classification, the domain focus, the data structure and so on. Considering the objective mentioned in Chapter 1 Section 1.3, the dataset is required to be related to corona virus as the objective undertaken is sentiment analysis of COVID-19 tweets. There are no public datasets available online for this task. With the increasing popularity of employing Twitter data for sentiment classification purpose (Bouazizi & Ohtsuki, 2018; Caramanis & Barber, 2017; C. Kaur & Sharma, 2020; Shelar & Huang, 2018), Twitter data is considered in this research.

As the objective is to focus on sentiment analysis of the impacted industries (Pharma, FMCG, Technology, Airlines, Tourism, Stock Market, Tele-Communication) due to corona virus, the data has been extracted from twitter with popular industry specific hashtags. To normalize the tweets, extracted only 1000 tweets for each hashtag. Selection of industries and related hashtags (mentioned below) is based on popularity and research through different websites. Total tweets accumulated with hashtags are 871176.

Hashtags used: *#COVID19, #StayHome, #coronavirus, #pandemic, #lockdown, #COVID-19*

Industries with popular hashtags during COVID-19:

Pharma – #biotech #ehealth #onmedic #healthcare
FMCG – #supermarket #grocery #consumer #beer #sanitizer #facemask
Technology- #tech #science
Airlines - #aviation #flights #airport
Tourism & Hospitality – #travel #hotels #quarantine #transport
Stock Markets – #stock #Stockmarket #investing #finance
Tele-Communication – #communication #Networking #workfromhome

A total of 8 categories were used in this task, as described:

User – username of the user tweeted

Text – Tweet text column

Date – Date of the tweet

Favourites – Favourite count for the tweet

Retweets – Retweet count of the tweet

Mentions – Mention of the other person in the tweet

Hashtags – Hashtags used in the tweet

Location – Location of the tweet

3.5 Polarity Assignment

To perform sentiment analysis, the raw data should be mapped with sentiment scores across tweets. Later on, these sentiment scores are divided into target classes for multiclass sentiment analysis. There are many python libraries to perform this in Natural Language processing. For this experiment, considered two popular libraries called TextBlob and Vader Analyzer.

Textblob is a python library for Natural language processing tasks. Textblob returns polarity and subjectivity of a sentence where Polarity lies between [-1,1], -1 is negative and +1 is positive. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. [13]

Vader is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative/neutral) and intensity (strength) of emotion. NLTK package has this and can be applied directly to unlabelled text data. VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. Then the sentiment score of a text can be obtained by summing up the intensity of each word in the text. [14]. Vader has been found quite useful when dealing with

13 https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524

14 https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

social media texts as it specifically designed to sentiments expressed on social media. [15]

Advantages of Vader:

It performs well on social media texts and generalizes easily to multiple domains. Vader doesn't require training data and produces better sentiment scores on social media data.

After getting the polarities, plotted histograms to check the distribution of sentiment scores for both the NLP libraries. Figure 8 – 3.5 shows that Vader performed well in terms of uniform distribution of sentiments, whereas Texblob scores were extremely biased towards neutral. This also explains that Vader performs better with social media data.

Both the NLP libraries produces scores in the range of -1 to +1 for each tweet. We bucketed sentiments scores on the below criteria after checking a few tweets manually.

- Negative = <-0.2 Polarity score
- Neutral = >-0.2 and <0.2 polarity score
- Positive = >0.2 Polarity score

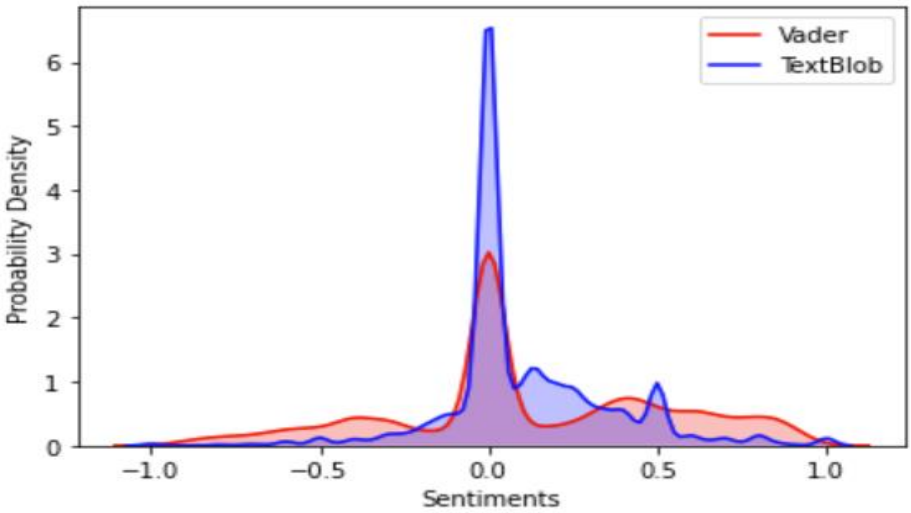

Figure 8 - 3.5 Vader and Textblob sentiment score distribution Graph

15 https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f

Though the scores achieved are good, a manual check is performed by taking 500 random tweets. These tweets were labelled manually, and cross verified the with the Vader sentiments. Vader Score has got 96.8% accuracy where as Textblob has got only 82%. By keeping statistical results in mind, Vader scores have been taken to categorize tweets into *Positive, Negative* and *Neutral* sentiments.

3.6 Data Exploration

It is essential to understand insights in the data before building predictive models. Through data exploration, data insights can be drawn. Below is the simple description of the attributes in the data.

```
0   User            19794 non-null  object
1   Text            19794 non-null  object
2   Date            19794 non-null  object
3   Favorites       19794 non-null  int64
4   Retweets        19794 non-null  int64
5   Mentions        4319 non-null   object
6   HashTags        19782 non-null  object
7   vader_polarity  19794 non-null  float64
8   Num_Sentiment   19794 non-null  int64
```

It appears that there are null entries in “Mentions” and “Hashtags” fields. There is not much use with the “User” field for our analysis as there are so many user tweets in the data. “Date” is further split into “Tweet_Date” and “Tweet_Time”. This could help in identifying number of tweets per day.

Figure 9-3.6 below depicts the date wise distribution of tweets for the top 30 days in the six months period. It is evident from the graph that majority of tweets related to covid19 are from the month of April followed by June which could possibly suggests the peak time for coronavirus. The number of cases has been rising during that month and people were sitting at home expressing their feelings on social media. By looking at the Donut chart Figure 10 – 3.6 below, the number of tweets posted per day is high on Mondays and least on Sundays.

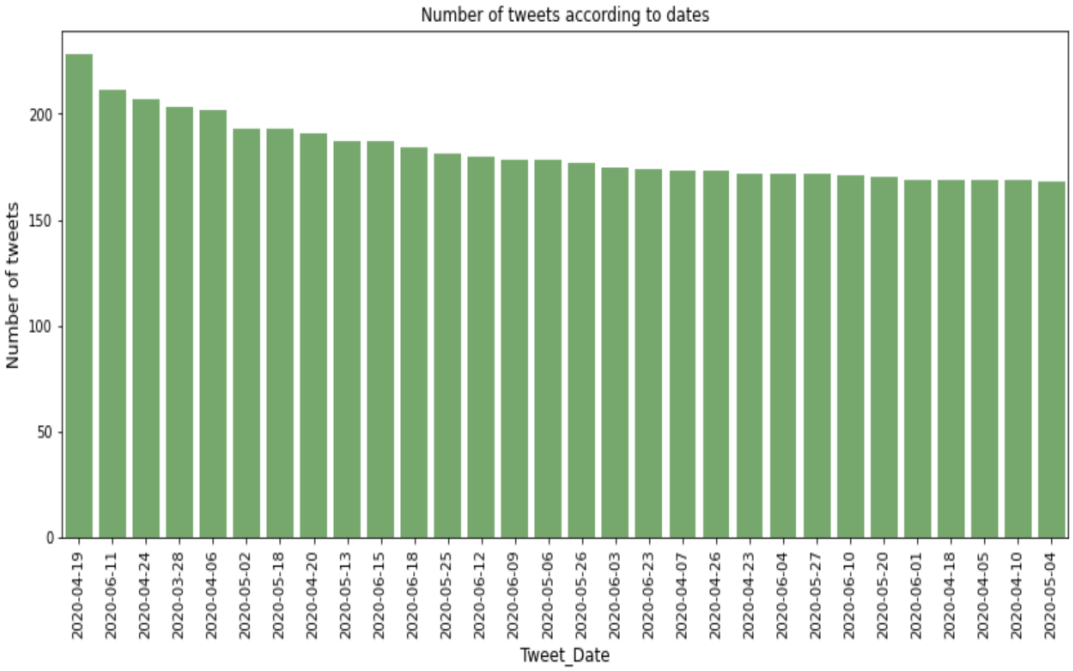

Figure 9 - 3.6 Number of tweets vs Date graph for top 30 days

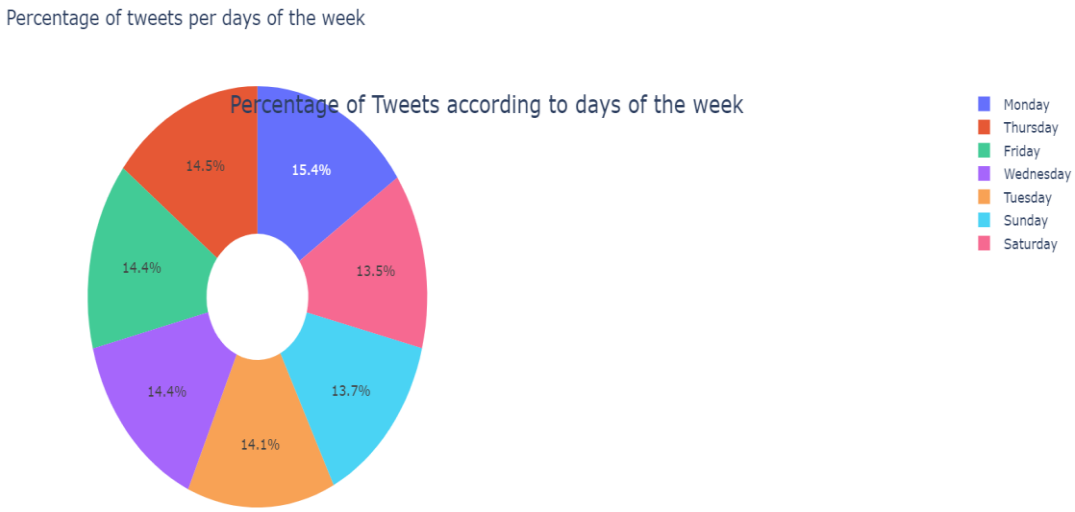

Figure 10 - 3.6 Donut Graph of weekdays vs tweet percentage

To understand the sentiment variation across all the tweets for the entire six months period, sentiment scores plot is taken. Figure 11 – 3.6.

- Sentiments are mostly neutral for the first 2 months from Jan-Feb. Thereafter, there is an immediate spike in positive and negative sentiments for a period of 1 month between Feb end till March Mid.

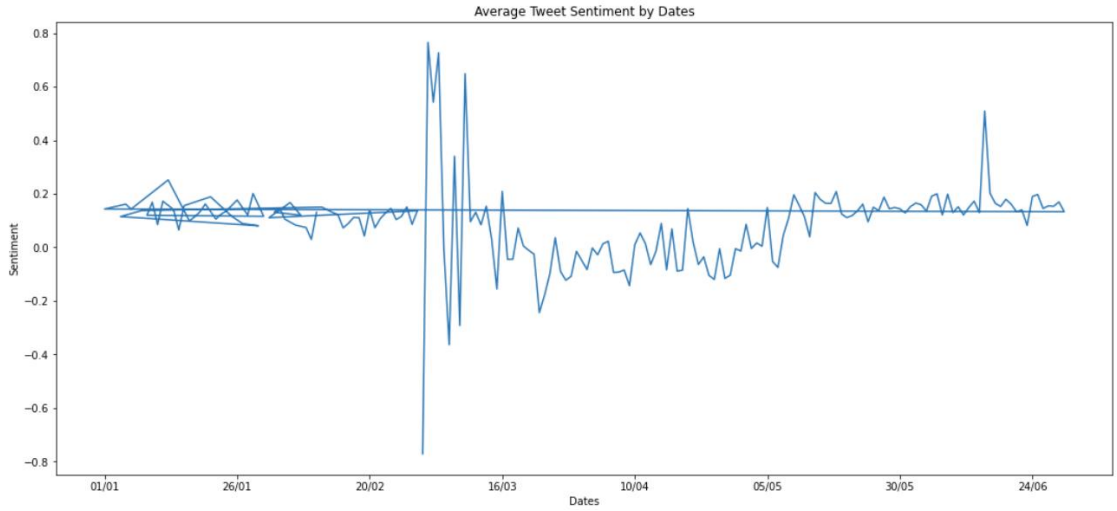

Figure 11 - 3.6 Sentiment scores plot for the entire period of six months

- Most of the negative sentiments appears to be between March mid till May. This could be because of the increased number of cases during that period.
- Then the further period has mostly neutral and positive sentiments. After May, corona cases started to subside a bit and possible corona vaccine progress has triggered neutral and positive sentiments.

Retweet count plot Figure 12 illustrates the information about the popularity of a kind of tweet. Fig above Retweet vs Polarity shows that maximum number of retweets are accounted for Neutral and Positive sentiments.

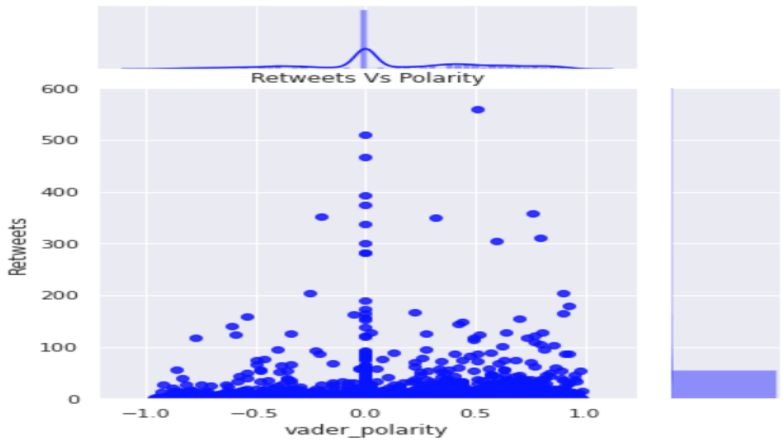

Figure 12 - 3.6 Retweet count against Vader polarity scores

Figure 13 - 3.6 Wordcloud representation of All tweets and Positive Tweets(Left to right)

Figure 14 - 3.6 Wordcloud representation of Negative and Neutral tweets (left to right)

Word cloud representation will provide the information of most frequent words used in the text. The plots Figure 13 & 14 depict the most frequent words for All, Positive, Negative and Neutral categories. All means all the tweets are taken.

Most used words across 3 sentiment categories are below:

Note: Only top100 most frequent words are taken

Positive: readiness, practice, earnings, information, wake, outbreak

Negative: covid, magazine, new, speed, flights, quite, Friday

Neutral: case, coronavirus, bleak, barrel, passed, kits, away, test, detected

For sentiment classification task, it is important to understand the target class distribution of the dataset. In this experiment, gathered data has neutral and positive tweets with a smaller number of negative tweets. Funnel chart is drawn Figure 15 below to illustrate the same information. Interestingly, Negative sentiments are less than 20% of overall tweets related to industries even though a deadly outbreak was going on for such a long period. Neutral sentiments have higher number when compared to the other 2 categories.

Neutral- 8428, Positive – 7579, Negative – 3787

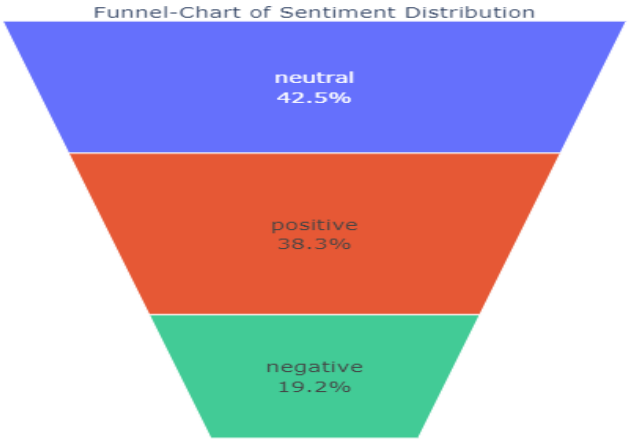

Figure 15 - 3.7 Percentage representation of each tweet category

3.7 Data Preparation

As the original tweet text contains all sorts of symbols, slang words, twitter handles, hashtags, URL's, improper grammar etc. owing to limited sentence length, it gets difficult to process the tweets and train them in a classifier model to perform the tweet classification based on the tweet text. As the current project intends to classify tweets using several machine learning algorithms into one of the many humanitarians categories and compare them in terms of precision, recall and F-scores, while also trying to use tweet sentiments as one of the features to improve the classification accuracy of the models, it is important to clean the tweets before feeding them into the classifier models as well as before performing sentiment analyses on them.

Twitter data preparation in this case includes the following tasks.

First, filter the dataset with industry popular hashtags mentioned in Section 3.4. This is because our objective is to do sentiment analysis on industry specific COVID tweets. Through this process only 19794 tweets left out of 8 lac tweets. Then Expanding contractions such as “ain’t” to “is not”. A list of contractions are taken to perform this task. Contractions reduce the performance of the model. It is always suggested to expand contractions for better accuracy results. Also need to strip spaces in the beginning and ending. URLs have unnecessary characters and don’t contribute for the classification purpose. Hence removed urls. Then removed account handles starting with @ and hashtags starting with # from the text field as there are multiple hashtags and account handles present which makes a confusing sentence in this case. Then removed duplicate entries and utf8 characters.

The cleaned up dataset is then utilized to perform sentiment analyses, named-entity extraction, contextual categorization as well as tweet text classi_cation using several state-of-the-art machine learning classifiers.

3.8 Modelling

The research aim is to implement sentiment classification by finetuning transformer-based models Bert and XLNet. There are 2 stages in this. First, Bert and XLNet base models will be finetuned by using complete data with Dropout and Mixout regularizations applied. In the second stage, finetuning will be performed after under sampling the dataset. The data split used for the implementation of all the models is 60:20:20 as train, test and validation.

3.8.1 Finetuning Bert

This part explains the finetuning BERT base uncased(uncased- trained on lower-case English text) model with 2 regularization techniques which are Dropout and Mixout. Once the cleaned and labelled dataset is imported, the target labels should be encoded for multiclass classification. Encoded target labels are Neutral- 0, Positive -1, Negative -2.

In the finetuning process, Pre-trained BERT base uncased model is taken from Hugging face transformers[16] and applied a classifier layer with dropout regularization. This base model has 12 Encoders with 12 bidirectional self-attention heads with total 110M parameters.

To feed our text to BERT, it must be split into tokens, and then these tokens must be mapped to their index in the tokenizer vocabulary. The tokenization must be performed by the tokenizer included with BERT. These tokenizers are to separate sentences from each other. Encoding also pads sentences to maximum length specified. In this case, max length calculated is 87. So, this maximum length is used to pad sequences. This makes all the sequences of constant length. It also appends attention masks which are typically array of 0s (pad token) and 1s (real token). These are to differentiate real tokens from padding tokens with the "attention mask". Then the features input_ids, attention_masks and labels are converted into torch tensors. A torch. Tensor is a multi-dimensional matrix containing elements of a single data type. After this, to process the data in batch mode, dataloaders must be created for train, validation, and test sets. This avoids loading all the data into memory at once.

After initialising the model, a classifier: a sequential layer is added as given below. This classifier layer consists of dropout layer with 0.5 value for the first model. Then Pass input_ids and attention masks created. By extracting last hidden state of the '[CLS]' token and passing it to classifier layer, outputs are computed.
Values for the classifier layer are below.

```
D_in, H, D_out = 768, 50, 3
```

Sequential(Linear layer – Input(768), output(50)
Relu - Activation function, Dropout – Regularization(0.5)
Linear layer – Input(50), output(3))

Model has been compiled with AdamW optimizer as suggested in (Devlin et al., 2019) and CrossEntropyLoss function. Loss function measure the performance of the classification model by producing a probability value between 0 and 1. After training

[16] https://huggingface.co/

and validating, test dataloader is created to predict the model on test data. Computed probabilities using softmax function. Further to check the model performance, Accuracy, Classification report and Confusion matrix are taken which can provide information about Precision, Recall, f1-score and predictions. In some cases, learing graphs have been used to understand the model fitting.

In the second case, Bert is finetuned with mixout regularization instead of dropout. To do this, the classifier layer is added which has only a linear layer with 768 input features and 3 outfeatures. Further, using mixout code, this layer is converted into Mixlinear by adding mixout value of 0.5. Figure 16 below. The procedure followed for the rest of the process is similary except this change. Model is compiled with the optimizer and loss function specified above and predicted the model on test data after training. All the performance metric reports mentioned above are considered similar to Bert with dropout case.

```
self.bert=BertModel.from_pretrained("bert-base-uncased")
# Instantiate an one-layer feed-forward classifier
self.classifier = nn.Linear(768,3)
```

```
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (classifier): MixLinear(mixout=0.5, in_features=768, out_features=3, bias=True)
)
```

Figure 16 - 3.8.1 BERT before and after applying Mixout

The hyperparameter values used are same for both the implementations.

Batch_size = 32 #Recommend by the authors

Learning rate= 2e-5

Epsilon value= 1e-8 #default

Num of epochs= 2 #Recommended 2 to 4

Batch size is a hyperparameter that controls the number of training samples to work through before updating the internal parameters of the model.

Learning rate is the amount of the weights that will be updated during training

Number of epoch are full training cycle of the model. Number 2 means, model will complete two cycles for the training dataset.

3.8.2 Finetuning XLNet

This part explains the finetuning XLNet base cased model with 2 regularization techniques which are Dropout and Mixout. XLNet also has 12-layer, 768-hidden, 12-heads with 110M parameters. The pre-trained model is taken from Hugging face transformers and a single classifier layer is added during finetuning. Target labels are one hot encoded in this case.

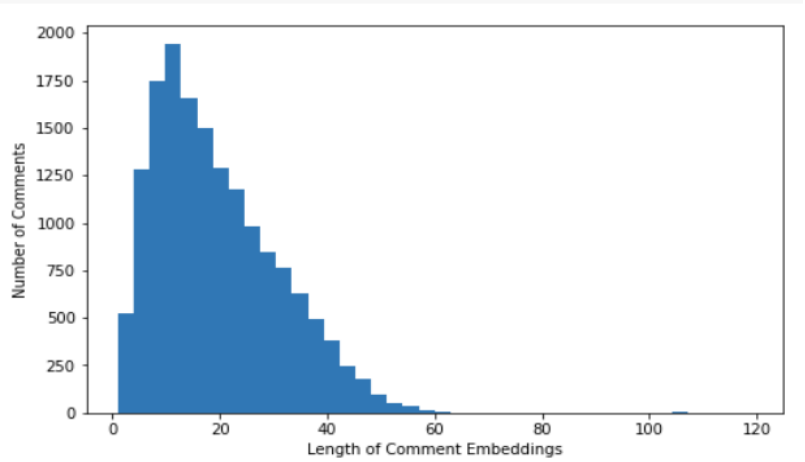

Figure 17 - 3.8.2 Train data embeddings length

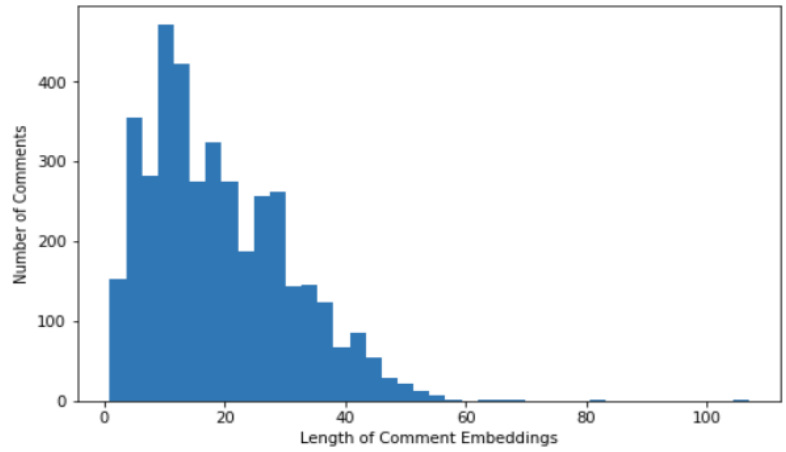

Figure 18 - 3.8.2 Test data embeddings length

Finetuning performed for this is like the one performed in section 3.8.1 as the models work on similar mechanism. After the initial steps, XLNet tokenizer is used to tokenize

the data and add input_ids and attention_masks. Figures 17 & 18 above shows the length of embeddings for both train and test data. As most of the embeddings have less than 60 length, max length is of 87 like section 3.8.1.

Once the input_ids and attention_masks are created, converted them to torch tensors to pass as inputs to the model. After that loaded the XLNet model to add a classifier layer which is sequential with Dropout and ReLU activation function as given below. Similar to the section 3.8.1, model and classifier layer are linked with input values and last hidden state of the '[CLS]' token is taken to compute logits. Logits are the probabilities of the computing function.

Sequential(Linear layer – Input(768), output(50);
Relu - Activation function; Dropout – Regularization;
Linear layer – Input(50), output(3))

Model has been compiled with AdamW optimizer and Binary Cross entropy loss function. Binary function is used as the labels are one hot encoded. After training and validation process, model is used to predict the test data. Softmax is used to get the probabilities of the predictions. Then similar to the section 3.8.1, values for the perforamance metrics considered are taken to evaluate the model performance.

Similar to the second model in section 3.8.1, mixout code is applied in the classifier to convert the linear layer to Mixlinear with mixout percentage of 0.5. Mixout value is fixed after trying with multiple values. Optimizer and Loss function are same as above experiment. Model is trained for 2 epochs, then test dataloader is created to predict the model on test data. Performance metrics used are same. Figure 19 gives the idea of Mixlinear layer.

```
self.xlnet = XLNetModel.from_pretrained('xlnet-base-cased')
# Instantiate an one-layer feed-forward classifier
self.classifier= nn.Linear(768, num_labels)
```

```
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (classifier): MixLinear(mixout=0.5, in_features=768, out_features=3, bias=True)
)
```

Figure 19 - 3.8.2 XLNet model before and after applying mixout

Hyper parameter values are similar to the first 2 experiments in section 3.8.1

Batch_size = 32 #recommended

Learning rat = 2e-5

weight_decay =0.01 #default

Num of epochs = 2 #Recommended 2 to 4

3.8.3 BERT and XLNet finetuning with under sampled data

In the next stage of the experiment, dataset is under sampled to reduce the number of training examples. From each category, 3000 instances are selected to balance target classes which will make balanced dataset and also reduce the number of training instances for the task needed. To check the objective as mentioned in Chapter 1 Section 1.3, mixout and dropout regularizations should be applied on less number of training instances to verify the performance difference of mixout regularization with dropout for classifying tweets. Once the data is under sampled, finetuning of models is done similar to the experiments described in sections 3.8.1, 3.8.2. All the hyperparameter values are kept same.

3.9 Evaluation

Performance prediction can be done by considering different measures. Totally depending on one factor is not the correct way for understanding how better a model is performing. For example, a model can get more than 95% accuracy when the data is not balanced by predicting majority class correctly. So it is better to make sure that the model is able to recognize both positives, negatives and neutral instances correctly as much as possible. Experiments conducted as part of this thesis have both types of datasets. This tells that for this research **Precision**, **Recall, f1-score and Accuracy are** considered as the main performance evaluation metrics. Because Precision summarizes the fraction of examples assigned the class that belong to the same class and recall refers to the percentage

of total relevant results correctly classified by the algorithms. F1-score combines both precision and recall. **Accuracy** is defined as the ratio of correctly predicted examples by the total predicted class. Falsely predicted does make a difference in this case as we are predicting each class. Hence, it is important for the model to have good precision and recall. Accuracy is a good performance metric where the experiments conducted were with under sampled data as the data is balanced across each class.

To understand the result more and make sure models are not giving biased results a confusion matrix is evaluated along with the classification report which will tell the precision, recall, F1-score and other factors for both the target values and each of them is giving results correctly or not. Evaluation of the models is done by comparing the model performance with dropout and mixout implementation for both the models. Also, the comparison includes the performance variation between Bert and XLNet with same regularization technique which used same data.

4. RESULTS, EVALUATION AND DISCUSSION

This chapter mainly covers the final results achieved by different experiments and the description of the performance metrics shown by the classification report. Classification report will provide the prediction information for each class. This gives the deeper intuition of the classifier behaviour over accuracy which can mask the functional weakness of some classes in a multiclass problem. The metrics are defined on the basis True predicted and false predicted for each class. True prediction is when the actual class is the actual class and predicted class matches. If it doesn't match, it is false prediction.

In this research, 0- Neutral, 1- Positive and 2- Negative and confusion matrix will be 3*3 matrix. Classification report also includes macro average (averaging the un-weighted mean per label) and weighted average (averaging the support-weighted mean per label). A confusion matrix is a matrix which shows the performance of a classification model on test data for true values as shown below Table 1.

Size	Positive	Negative	Neutral
Positive	100	2	3
Negative	4	120	5
Neutral	2	3	110

Table 1 – 4 Example of confusion matrix for multiclass classification

4.1 Model Results and Evaluation

This section covers the results obtained by finetuning the pretrained language models on COVID-19 tweets with Dropout and Mixout techniques for sentiment classification.

To clearly understand different models developed, segregated models developed with original dataset and models developed with under sampled data as the main focus is on reduced instances.

Table 2 - 4.1 below has the results of classification report and confusion matrix. It is clearly evident from the results that XLNet with Mixout has performed better than the

rest of the models in terms of accuracy, precision, recall and f1 score. Even from the confusion matrix results, it has high number of true predictions for all the classes when compared to other models. XLNet with dropout has performed less in terms of all the metrics considered for this dataset on COVID19 tweets. The f1-score for negative class and recall for neutral class have registered low values which has impacted in predicting the test results. 515 of 1715 total records have been falsely predicted by the model which is almost 30% of the total instances for that class. When looked at BERT results, the negative class predictions have got low prediction rates which resulted in more false predictions for that class.

Model	Target class	Precision	Recall	F1-score	Positive	Negative	Neutral	Accuracy
BERT with Dropout	Positive	0.89	0.85	0.87	1338	29	158	84.17%
	Negative	0.79	0.84	0.81	24	605	92	
	Neutral	0.89	0.83	0.86	124	103	1488	
BERT with Mixout	Positive	0.89	0.86	0.88	1319	30	176	83.92%
	Negative	0.79	0.83	0.81	21	600	100	
	Neutral	0.87	0.82	0.85	149	125	1441	
XLNet with Dropout	Positive	0.86	0.86	0.86	1315	133	77	79.04%
	Negative	0.65	0.85	0.74	44	662	15	
	Neutral	0.93	0.7	0.8	175	340	1200	
XLNet with Mixout	Positive	0.9	0.89	0.9	1361	93	71	84.90%
	Negative	0.77	0.86	0.81	26	665	30	
	Neutral	0.93	0.81	0.87	124	205	1386	

Table 2 - 4.1 BERT and XLNet results without sampling

Table 3 – 4.1 below showing the results of BERT and XLNet finetuning with Dropout and Mixout regularization techniques with under sampled data.

The performance of finetuning 2 pre-trained language models with less training instances is less than the models developed in the first part. In both the cases, models finetuned with Mixout regularization have produced better performance results. BERT with dropout has produced 76.78% accuracy but with mixout, the model was able to achieve 78.78% accuracy. Similarly, XLNet with dropout has got 72.94% and the same model finetuned with mixout has got 81.61% which is almost 9% increment than the base model. As the dataset is balanced in this case, the main performance metric is

accuracy. However, to understand the predictive capability of each model, classification report and confusion matrix are taken.

Results with low recall and precision are marked with yellow in the table. Dropout model for BERT has 66% of recall for neutral class and the same is reflected in predicting the target with more false predictions. Similar results have been observed for Recall with BERT with mixout and XLNet with dropout for neutral class. XLNet with dropout has more number of false predictions when compare to the rest of the models in this scenario. Though the recall(68%) and false predictions(25%) are a bit more for XLNet with mixout model, the accuracy achieved and true prediction percentage is very high compared to the other 3 models.

Model with under sampling	Target class	Precision	Recall	F1-score	Positive	Negative	Neutral	Accuracy
BERT with Dropout	Positive	0.82	0.74	0.78	446	63	92	76.78%
	Negative	0.77	0.89	0.83	24	548	40	
	Neutral	0.8	0.66	0.73	70	102	413	
BERT with Mixout	Positive	0.82	0.81	0.82	487	39	75	78.78%
	Negative	0.82	0.86	0.84	20	526	68	
	Neutral	0.79	0.69	0.74	86	75	424	
XLNet with Dropout	Positive	0.74	0.89	0.82	546	47	21	72.94%
	Negative	0.71	0.72	0.71	106	458	21	
	Neutral	0.88	0.6	0.71	64	185	352	
XLNet with Mixout	Positive	0.83	0.91	0.87	557	37	20	81.61%
	Negative	0.88	0.68	0.76	88	422	75	
	Neutral	0.85	0.88	0.86	19	56	526	

Table 3 - 4.1 BERT and XLNet with under sampled data

4.1.1 BERT finetuning

This section explains the results for BERT model with Dropout and Mixout.

Table 4 – 4.1.1 displays the training loss, validation loss and validation accuracy for 2 epochs. Figure 7 is the classification report drawn after predicting the model on test data. Confusion matrix developed is converted into table for understanding purpose. Finetuning BERT base model with dropout and mixout for just 2 epochs has given almost similar accuracy results on validation data (Table 4). The accuracy achieved on test data prediction is 84.17% and 83.92% with dropout and mixout as mentioned in Table 2 – 4.1.

	Validation Loss	Validation Accuracy
Dropout	0.39	86%
Mixout	0.39	85.47%

Table 4 – 4.1.1 Loss & validation accuracy of BERT after 2 epochs

From the classification report results added in Table 2 – 4.1, model is able to achieve 89% precision for both neutral and positive class and 79% for Negative class in case of dropout but for mixout the scores are slightly less for neutral class. Recall and f1 scores are good in both cases. This suggests that the finetuned model performed well with imbalanced data. But the number of instances in the negative class are very less compared to the other two and the precision achieved is also less compared to the other classes in the report.

From the confusion matrix values in Table 2 – 4.1, true and false predictions have similar results in both cases except for neutral class. Mixout model has more number of wrong predictions.

To conclude, mixout regularization didn't impact the model performance. All the performance metrics have similar results compared to the experiment 1.

4.1.2 XLNet finetuning

This section covers the results achieved by XLNet model with dropout and mixout.
In this, XLNet model is finetuned with dropout. The model is trained for 2 epochs with dropout and mixout techniques. Figures 20 – 4.1.2, 21 – 4.1.2 shows the learning curves for training and validation data for both models. The training and validation loss graphs are decreasing as the number of epochs increases.

After applying the model on test data, 79.04% accuracy has been achieved by dropout model whereas mixout model has got 84.90%. In addition to that XLNet model with dropout achieved less accuracy than BERT model with dropout section 4.1.1. From the classification report in Table 2 – 4.1, it is clear that the model was able to predict Neutral

and positive classes with good precision but for negative class, the precision is just 65%. Recall and f1-scores are still good. The model is biased to majority class. Similarly, with mixout, negative class has less precision when compared to the other two classes.

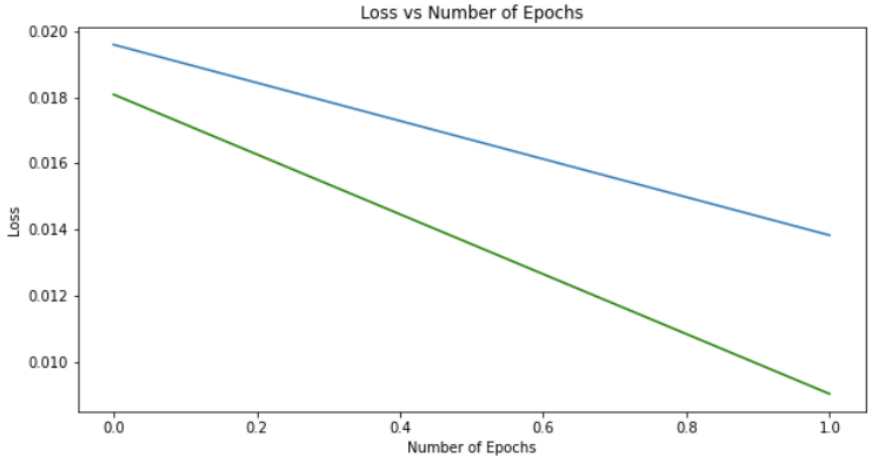

Figure 20 – 4.1.2 Train and validation loss of XLNet dropout model

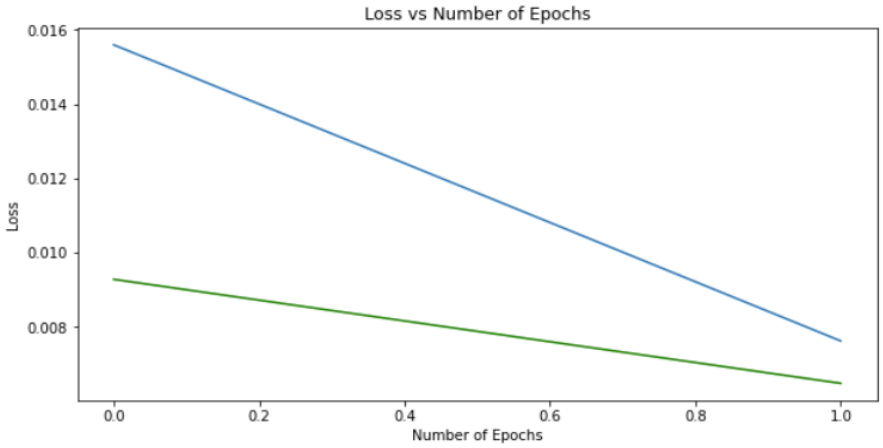

Figure 21 - 4.1.3 Train & validation loss of XLNet mixout model

In terms of confusion matrix given values in Table 2 – 4.1, the correct predictions are good for positive and negative class. For neutral class, 515 wrong predictions are there for a total 1715 instances. This is almost 25% of the data for that class. Similarly, predictions for neutral class are 329 out of 1715 which is significantly high in XLNet with mixout model but less than XLNet with dropout model. Overall, the model didn't perform well when compared to BERT model results mentioned in 4.1.1. But with

mixout, XLNet was able to outperform BERT model with dropout and mixout, XLNet with dropout.

4.1.3 BERT finetuning – Under sampled data

In this part, BERT base model is finetuned with Dropout and Mixout regularization techniques. The original dataset is under sampled by selecting 3000 instances for each class. The dataset is balanced here with reduced number of total instances. Training, validation and test splits are 5400,1800,1800. Maximum validation accuracy achieved is approximately same in both cases.

When the models were tested on test data, dropout model got 76.78% accuracy and mixout model got 78.78% which is more than the BERT dropout model with under sampled data.

	Validation Loss	Validation Accuracy
Dropout	0.55	79%
Mixout	0.54	78.40%

Table 5 - 4.1.3 Validation loss & accuracy for BERT with under sampled data

From the classification report values given in Table 3 – 4.1, we can see that the precision is around 80% for all 3 categories which suggests that the model did a good job here. However, the recall percentage for neutral class is bit low compared to other classes. Similar results are seen for Mixout model as well with under sampled data as mentioned in Table 3 – 4.1. That is why the false predictions in neutral class are high in both cases in this experiment section.

4.1.4 XLNet finetuning – Under sampled data

This part explains the results of finetuning XLNet with dropout and mixout by using reduced data. The original dataset is under sampled and split in the ratio of 60:20:20 similar to the section 4.1.3. The model is trained for 2 epochs in each case. From the learning graphs, we can observe the decrease in training and validation loss for Dropout and Mixout models.

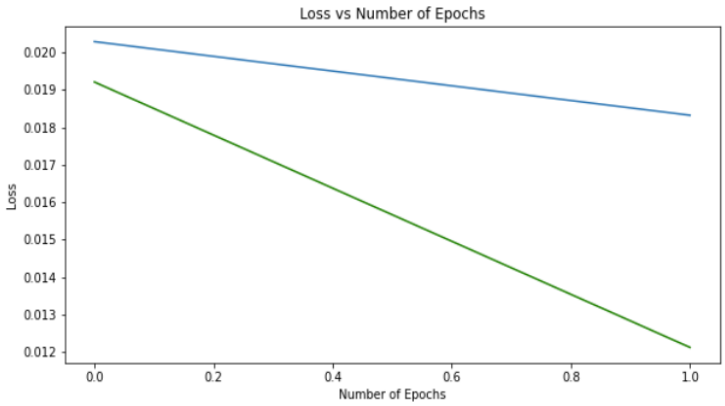

Figure 22 - 4.1.4 Validation loss vs epochs for XLNet with under sampled data

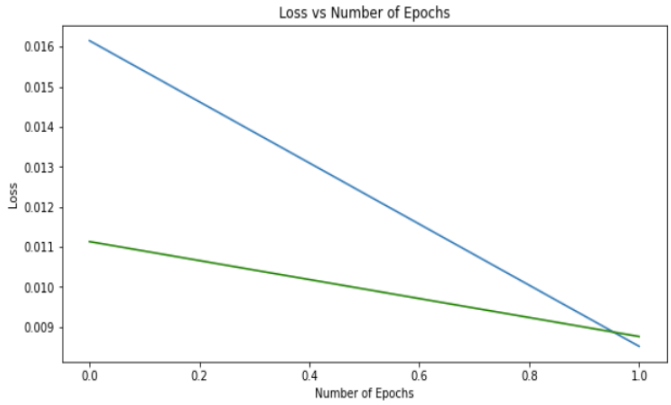

Figure 23 - 4.1.4 Validation loss vs epochs for XLNet with Mixout -under sampled data

In terms of accuracy comparison, XLNet with dropout in this case has achieved 72.94% which is even less than the Bert model developed without sampling. XLNet model with mixout was able to produce 81.61% accuracy which is higher than all the three models finetuned with sampled data.

The classification report values are given in Table 3 – 4.1 which suggests good precision rate for neutral class but recall is not that great for the model with dropout regularization. True and false predictions are better than the rest for Positive class. Same things can be observed from the confusion matrix table. Neutral class has highest number of false predictions. Overall, the model performance is lower than the rest of the models developed with under sampled data.

Precision, recall and f1-scores are good for positive and neutral classes in case of Mixout implementation. From the confusion matrix also, more number of true predictions for positive and neutral class. For negative class, false predictions are significantly high compared to the other 2 classes.

From all the observations, this model has performed better than the BERT models with dropout & Mixout, XLNet model with dropout illustrated in Section 4.1.3.

4.2 Discussion

Data is extracted from twitter and pre-processed using Natural Language Processing. Polarity assignment is done using Vader Analyzer. Feature extraction is done after doing tokenization with model tokenizers in both models BERT and XLNet which is already explained in the Design and Methodology. Then finetuned XLNet and BERT base models with dropout and mixout regularization techniques as explained in the experimentation part for each. After that, under sampled the data to reduce training instances and finetuned same models with both dropout and mixout regularization. Results comparison is done in the previous section with the performance metrics considered. This part has the brief discussion of the results and evaluation.

4.2.1 BERT and XLNet comparison

Table above has the results of classification report and confusion matrix. From the results achieved in Table 2 – 4.1, XLNet with Mixout has performed better than the rest of the models in terms of accuracy, precision, recall and f1 score. Even from the confusion matrix results, it has high number of true predictions for all the classes when compared to other models. XLNet with dropout has performed less in terms of all the metrics considered for this dataset on COVID19 tweets. The f1-score for negative class and recall for neutral class have registered low values which has impacted in predicting the test results. 515 of 1715 total records have been falsely predicted by the model which is almost 30% of the total instances for that class. When looked at BERT results, the negative class predictions have got low prediction rates which resulted in more false predictions for that class.

From the results obtained, it can be concluded that adding mixout doesn't degrade the model performance when enough training instances are present. This is true in both cases as it didn't register any decrement in the model performance results in either case except a small margin by BERT model. But it is not significant enough to say that the model performance is not good when compared to dropout model.

It has been mentioned in the research paper of XLNet that the model has beat BERT in 20 different tasks such as; question answering, natural language inference, sentiment analysis and document ranking (Yang et al., 2019). However, we didn't achieve better results for XLNet than BERT when dropout used as regularization technique with the data gathered. There might be influencing factors as the data taken is extracted manually and labelled with NLP lexicon libraries in python. Or more number of epochs and hyper parameter tuning might give better results than BERT with similar regularization strategy. However, with mixout the results are better than BERT model.

4.2.2 BERT and XLNet with under sampled data comparison

Table 3 – 4.1 mentioned in Section 4.1 showing the results of BERT and XLNet finetuning with Dropout and Mixout regularization techniques with under sampling. Combined classification and confusion matrix results are given the in table for all the models.

The performance of finetuning the two pre-trained language models with less training instances is less than the models developed with full data. In both the cases, models finetuned with Mixout regularization have produced better performance results. XLNet model with Mixout has produced higher accuracy results than the other three models. As the dataset is balanced in this case, Accuracy can be considered as the main performance metric. However, to understand the predictive capability of each model, classification report and confusion matrix are taken. Details in Table 3 – 4.1.

XLNet with dropout has more false predictions when compared to the rest of the models in this scenario. But, XLNet model with mixout, the accuracy achieved and true prediction percentage is very high compared to the other 3 models.

To conclude, the objective is proved in both the cases; finetuning BERT, XLNet with Mixout has produced better results than finetuning with Dropout regularization.

5. CONCLUSION

This chapter provides conclusions for the work done in all the chapters above. It briefly explains on the research overview given, problem definition, experiment design, results and evaluation as discussed in the previous chapters. Towards the end, it discusses the contributions and impact of the experiment conducted in this work also explains the future work and recommendations for further studies in this domain.

5.1 Research Overview

The research in this thesis was conducted in four parts – Extracting data from twitter using popular hashtags for COVID-19, label the dataset using Vader analyser polarity scores, Analysing the tweets extracted to understand sentiment variation for the entire period and performing text classification on those tweets by finetuning pretrained language models with two different regularization techniques. Two stages of modelling were there. Performing text classification by finetuning BERT and XLNet for the entire data (around 19000 tweets) with dropout, mixout and finetuning the same models with reduced data (9000 total) after under sampling. The performance of the tweet text classification models was evaluated for each model with regularization techniques and change in the sample size. The classification performance of each model was compared in terms of precision, recall, f1 score and accuracy. This comparison has given a clear view to either accept or reject the formulated hypothesis of the research.

5.2 Problem Definition

The research problem was defined by the question: *"To what extent finetuning Transformer based deep learning models like XLNet and BERT with Mixout can provide better accuracy results when compared to finetuning with Dropout when there are less training instances in a Multiclass sentiment classification using Twitter tweets on COVID-19?"* and four sub-questions:

Is there any difference in classification performance of covid19 related tweets when finetuned with BERT and XLNet with dropout in a multiclass problem?

Does using mixout regularization technique to finetune BERT and XLNet improves classification performance when compared to Dropout regularization with enough training instances?

Does using mixout strategy instead of dropout regularization improves performance of multiclass classification when there are less training instances?

Which classifier performs best in terms of accuracy, precision, recall and f1-score for classifying covid19 tweets in both cases of training instances mentioned above?

The main purpose of the research was to establish the validity of the following hypothesis:

Null Hypothesis: If Mixout regularization is used when there are less training instances to finetune pre-trained language models such as BERT and XLNet base models to address sentiment classification problem of twitter tweets on COVID19, they cannot statistically outperform finetuning the same models with Dropout regularization on classification accuracy.

Alternate Hypothesis: If Mixout regularization is used when there are less training instances to finetune pre-trained language models such as BERT and XLNet base models to address sentiment classification problem of twitter tweets on COVID19, they can statistically outperform finetuning the same models with Dropout regularization on classification accuracy.

The research was mainly focussed on analysing the application of Mixout regularization strategy to finetune pretrained language models BERT and XLNet with less training data. And to check the impact when finetuned with enough training data(>10K) as mentioned in (Lee, 2020).

5.3 Experiment, Evaluation & Results

The design of the experiment was clearly mentioned with fine-grained details about how the language models have been used by finetuning on the data gathered for multiclass text classification. The dataset was good in size (around 19000) tweets for COVID-19 outbreak analysis. Industry popular hashtags are used to filter the data gathered by using

COVID-19 hashtags. These tweets were labelled with sentiment score from Vader Sentiment Analyzer as Positive, Negative and Neutral. The dataset was not balanced in the first case. Performance metrics were chosen accordingly. In the second case, under sampling was used to balance and reduce the data.

The approach to perform the tweet text classification was well chosen after thorough research. The models chosen are leading language models at this time which are known to provide best results for related tasks. Experiments carried out were finetuning language. In addition, finetuning of the models BERT and XLNet has been performed on COVID-19 tweets with and without under sampling to verify the impact of mixout when there are enough training examples(>10K). The whole process has given a clear picture of mixout regularization in two cases.

From the results obtained, it was concluded that BERT has produced better results than XLNet in case of dropout regularization in terms of Precision, Recall, f1 score and Accuracy with enough training instances. In case of Mixout, XLNet beat BERT with a small margin. But overall, the conclusion is that mixout didn't produce any detrimental impact on the performance with more data. In case of under sampled data also, BERT beat XLNet when dropout was used. But, XLNet has given better results than BERT with dropout and mixout models, XLNet with dropout model. Also, BERT with mixout performed better than BERT with dropout model. Hence, null hypothesis can be rejected for this work as Mixout models performed better than Dropout models with less training examples.

5.4 Contributions and Impact

In the current work, a thorough analysis was done to extract and process the data from Twitter. The richness of useful information obtained from twitter regarding COVID-19 was demonstrated in this work. Although the focus of the current work was limited to textual data obtained from Twitter, it has the capacity to be supplemented with additional information such as images, multimedia content etc. Also, mixout technique can be applied to other pretrained language models and deep learning models to check the effectiveness of the regularization technique or this could be a starting point for other methods to come.

The innovation of this work is that the data taken is completely new which has covered a period of six months for different impacted industries despite of the limitations with Twitter end. Though the concept is based on the existing literature, mixout was not applied to XLNet model and BERT base models. This work has the potential to pave a way for the researchers who wants to explore regularization methods for finetuning pretrained language models.

5.5 Future Work & Recommendations

Applying mixout regularization technique to different pretrained language models can be implemented by adding additional features in aspect-based sentiment analysis. Also, this work can be expanded to check the performance of various pretrained models for cross domain adaptability. Future work could also look into combining industry stocks performance with the sentiments on twitter for sentiment classification to understand the correlation between social media sentiments and stock performance. Another area of exploration can also involve gathering more data for each day on COVID-19 cases to understand the sentiment variation during this recovery period. It is also advisable to look for or prepare a COVID-19 dataset with verified labels to improve the classification performance.

One other of future work can use hyperparameter tuning for some parameters during finetuning for the specific task. Most importantly, future work can also focus on applying this regularization strategy to the entire model instead of classifier layer by keeping weights intact.

6. BIBLIOGRAPHY

A., V., & Sonawane, S. S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, *139*(11), 5–15. https://doi.org/10.5120/ijca2016908625

A 31. (n.d.-a). https://doi.org/10.1038/nature07634

A Gentle Introduction to Transfer Learning for Deep Learning. (n.d.). Retrieved August 31, 2020, from https://machinelearningmastery.com/transfer-learning-for-deep-learning/

Abd-Alrazaq, A., Alhuwail, D., Househ, M., Hamdi, M., & Shah, Z. (2020). Top Concerns of Tweeters During the COVID-19 Pandemic: Infoveillance Study. *Journal of Medical Internet Research*, *22*(4), e19016. https://doi.org/10.2196/19016

Abd El-Jawad, M. H., Hodhod, R., & Omar, Y. M. K. (2019). Sentiment analysis of social media networks using machine learning. *ICENCO 2018 - 14th International Computer Engineering Conference: Secure Smart Societies*, 174–176. https://doi.org/10.1109/ICENCO.2018.8636124

Abid, F., Alam, M., Yasir, M., & Li, C. (2019). Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter. *Future Generation Computer Systems*, *95*, 292–308. https://doi.org/10.1016/j.future.2018.12.018

Alharbi, A. S. M., & de Doncker, E. (2019). Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. *Cognitive Systems Research*, *54*, 50–61. https://doi.org/10.1016/j.cogsys.2018.10.001

Alsaeedi, A., & Khan, M. Z. (2019). A study on sentiment analysis techniques of Twitter data. *International Journal of Advanced Computer Science and Applications*, *10*(2), 361–374. https://doi.org/10.14569/ijacsa.2019.0100248

Beigi, G., Hu, X., Maciejewski, R., & Liu, H. (n.d.). *An Overview of Sentiment Analysis in Social Media and its Applications in Disaster Relief*.

BERT Explained: State of the art language model for NLP | by Rani Horev | Towards Data Science. (n.d.). Retrieved August 31, 2020, from https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Binti Hamzah, F. A., Lau, C. H., Nazri, H., Ligot, D. C., Lee, G., Tan, C. L., & et al.

(2020). CoronaTracker: World-wide Covid-19 outbreak data analysis and prediction. *Bulletin of the World Health Organization*, *March*, Submitted.

Bouazizi, M., & Ohtsuki, T. (2018). Multi-Class Sentiment Analysis in Twitter: What if Classification is Not the Answer. *IEEE Access*, *6*, 64486–64502. https://doi.org/10.1109/ACCESS.2018.2876674

Cai, M. (2013). *Sentiment analysis of tweets using Neural Networks*. *Nips*.

Caramanis, C., & Barber, K. S. (2017). *Comparison of Algorithms for Twitter Sentiment Analysis APPROVED BY SUPERVISING COMMITTEE*.

Chen, E., Lerman, K., & Ferrara, E. (2020). *COVID-19: The First Public Coronavirus Twitter Dataset*. 4–5. http://arxiv.org/abs/2003.07372

Chew, C., & Eysenbach, G. (n.d.). *Pandemics in the Age of Twitter: Content Analysis of Tweets during the 2009 H1N1 Outbreak*. https://doi.org/10.1371/journal.pone.0014118

Citation, R. (2019). *Detection of Offensive YouTube Comments , a Performance Comparison of Deep Learning Approaches Detection of offensive YouTube comments , a performance comparison of Deep Learning approaches*.

Çoban, Ö., Özyer, B., & Özyer, G. T. (2015). A comparison of similarity metrics for sentiment analysis on Turkish twitter feeds. *Proceedings - 2015 IEEE International Conference on Smart City, SmartCity 2015, Held Jointly with 8th IEEE International Conference on Social Computing and Networking, SocialCom 2015, 5th IEEE International Conference on Sustainable Computing and Communic*, 333–338. https://doi.org/10.1109/SmartCity.2015.93

Darwich, M., Mohd Noah, S. A., Omar, N., & Osman, N. A. (2019). Corpus-Based Techniques for Sentiment Lexicon Generation: A Review. *Journal of Digital Information Management*, *17*(5), 296. https://doi.org/10.6025/jdim/2019/17/5/296-305

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186.

Dholpuria, T., Rana, Y. K., & Agrawal, C. (2018). A sentiment analysis approach through deep learning for a movie review. *Proceedings - 2018 8th International*

Conference on Communication Systems and Network Technologies, CSNT 2018, 173–181. https://doi.org/10.1109/CSNT.2018.8820260

Dimitrios Gunopulos, R. (2002). Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection. *In Proceedings of Workshop on Data Cleaning and Preprocessing (DCAP 2002), at IEEE International Conference on Data Mining (ICDM 2002).*, 613–623. http://alumni.cs.ucr.edu/~ratana/DCAP02.pdf

Documentation Home | Docs | Twitter Developer. (n.d.). Retrieved August 29, 2020, from https://developer.twitter.com/en/docs

Dubey, A. D. (2020). Twitter Sentiment Analysis during COVID19 Outbreak. *SSRN Electronic Journal*, *March*, 1–9. https://doi.org/10.2139/ssrn.3572023

El Zowalaty, M. E., & Järhult, J. D. (2020). From SARS to COVID-19: A previously unknown SARS- related coronavirus (SARS-CoV-2) of pandemic potential infecting humans – Call for a One Health approach. *One Health*, *9*. https://doi.org/10.1016/j.onehlt.2020.100124

Elbagir, S., & Yang, J. (2018a). Sentiment analysis of twitter data using machine learning techniques and scikit-learn. *ACM International Conference Proceeding Series*, *June*. https://doi.org/10.1145/3302425.3302492

Farra, N., Challita, E., Assi, R. A., & Hajj, H. (2010). Sentence-level and document-level sentiment mining for arabic texts. *Proceedings - IEEE International Conference on Data Mining, ICDM*, *October 2014*, 1114–1119. https://doi.org/10.1109/ICDMW.2010.95

Gao, Z., Feng, A., Song, X., & Wu, X. (2019). Target-dependent sentiment classification with BERT. *IEEE Access*, *7*, 154290–154299. https://doi.org/10.1109/ACCESS.2019.2946594

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, *457*(7232), 1012–1014. https://doi.org/10.1038/nature07634

Gong, X. R., Jin, J. X., & Zhang, T. (2019). Sentiment Analysis Using Autoregressive Language Modeling and Broad Learning System. *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*, 1130–1134. https://doi.org/10.1109/BIBM47256.2019.8983025

H. Manguri, K., N. Ramadhan, R., & R. Mohammed Amin, P. (2020). Twitter Sentiment

Analysis on Worldwide COVID-19 Outbreaks. *Kurdistan Journal of Applied Research*, *May*, 54–65. https://doi.org/10.24017/covid.8

Hallac, I. R., Ay, B., & Aydin, G. (2019). Experiments on Fine Tuning Deep Learning Models With News Data For Tweet Classification. *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*, 1–5. https://doi.org/10.1109/IDAP.2018.8620869

Hallsmar, F., & Palm, J. (2016). *Multi-class Sentiment Classification on Twitter using an Emoji Training Heuristic*. 1–27. https://kth.diva-portal.org/smash/get/diva2:927073/FULLTEXT01.pdf

Hao, M., Rohrdantz, C., Janetzko, H., Dayal, U., Keim, D. A., Haug, L. E., & Hsu, M. C. (2011). Visual sentiment analysis on twitter data streams. *VAST 2011 - IEEE Conference on Visual Analytics Science and Technology 2011, Proceedings*, 277–278. https://doi.org/10.1109/VAST.2011.6102472

Jahanbin, K., & Rahmanian, V. (2020). Using twitter and web news mining to predict COVID-19 outbreak. *Asian Pacific Journal of Tropical Medicine*, *March*, 26–28. https://doi.org/10.4103/1995-7645.279651

Ji, X., Chun, S. A., & Geller, J. (2013). Monitoring public health concerns using twitter sentiment classifications. *Proceedings - 2013 IEEE International Conference on Healthcare Informatics, ICHI 2013*, *September*, 335–344. https://doi.org/10.1109/ICHI.2013.47

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep Convolution Neural Networks for Twitter Sentiment Analysis. *IEEE Access*, *6*, 23253–23260. https://doi.org/10.1109/ACCESS.2017.2776930

Jordan, S. E., Hovet, S. E., Fung, I. C. H., Liang, H., Fu, K. W., & Tse, Z. T. H. (2019). Using twitter for public health surveillance from monitoring and prediction to public response. *Data*, *4*(1), 1–20. https://doi.org/10.3390/data4010006

Kamiş, S., & Goularas, D. (2019). Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data. *Proceedings - 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications, Deep-ML 2019*, 12–17. https://doi.org/10.1109/Deep-ML.2019.00011

Kaur, A. (2019a). *Analyzing Twitter Feeds to Facilitate Crises Informatics and Analyzing Twitter Feeds to Facilitate Crises Informatics and Disaster Response During Mass Emergencies Disaster Response During Mass Emergencies*.

https://arrow.tudublin.ie/scschcomdis

Kaur, A. (2019b). *Analyzing Twitter Feeds to Facilitate Crises Informatics and Disaster Response During Mass Emergencies*. https://arrow.dit.ie/scschcomdis

Kaur, C., & Sharma, A. (2020). Twitter sentiment analysis on Coronavirus using Textblob. *EasyChair Preprint*, *2974*, 1–10.

Ke, P., Ji, H., Liu, S., Zhu, X., & Huang, M. (2019). *SentiLR: Linguistic Knowledge Enhanced Language Representation for Sentiment Analysis*. http://arxiv.org/abs/1911.02493

Kim, H., Jang, S. M., Kim, S. H., & Wan, A. (2018). Evaluating Sampling Methods for Content Analysis of Twitter Data. *Social Media and Society*, *4*(2). https://doi.org/10.1177/2056305118772836

Lai, A. L., Millet, J. K., Daniel, S., Freed, J. H., & Whittaker, G. R. (2020). *Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- company ' s public news and information website . Elsevier hereby grants permission to make all its COVID-19-r*. *January*, 19–20.

Lee, C. (2020). *M IXOUT : E FFECTIVE R EGULARIZATION TO F INETUNE L ARGE - SCALE P RETRAINED L ANGUAGE M ODELS*. 1–17.

Li, S., Wang, Y., Xue, J., Zhao, N., & Zhu, T. (2020). The impact of covid-19 epidemic declaration on psychological consequences: A study on active weibo users. *International Journal of Environmental Research and Public Health*, *17*(6). https://doi.org/10.3390/ijerph17062032

Medford, R. J., Saleh, S. N., Sumarsono, A., Perl, T. M., & Lehmann, C. U. (2020). An "Infodemic": Leveraging High-Volume Twitter Data to Understand Public Sentiment for the COVID-19 Outbreak. *MedRxiv*, 2020.04.03.20052936. https://doi.org/10.1101/2020.04.03.20052936

Mollema, L., Harmsen, I. A., Broekhuizen, E., Clijnk, R., De Melker, ; Hester, Paulussen, T., Kok, G., Ruiter, R., & Das, ; Enny. (n.d.). *Disease Detection or Public Opinion Reflection? Content Analysis of Tweets, Other Social Media, and Online Newspapers During the Measles Outbreak in the Netherlands in 2013*. https://doi.org/10.2196/jmir.3863

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, *42*(24), 9603–9611.

https://doi.org/10.1016/j.eswa.2015.07.052

Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2017). Sentiment analysis of Twitter data for predicting stock market movements. *International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016 - Proceedings*, 1345–1350. https://doi.org/10.1109/SCOPES.2016.7955659

Pota, M., Esposito, M., Palomino, M. A., & Masala, G. L. (2018). A subword-based deep learning approach for sentiment analysis of political tweets. *Proceedings - 32nd IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2018*, *2018-Janua*, 651–656. https://doi.org/10.1109/WAINA.2018.00162

Ramadhani, A. M., & Goo, H. S. (2017). Twitter sentiment analysis using deep learning methods. *Proceedings - 2017 7th International Annual Engineering Seminar, InAES 2017*, 9–12. https://doi.org/10.1109/INAES.2017.8068556

Rane, A., & Kumar, A. (2018). Sentiment Classification System of Twitter Data for US Airline Service Analysis. *Proceedings - International Computer Software and Applications Conference*, *1*, 769–773. https://doi.org/10.1109/COMPSAC.2018.00114

Rietzler, A., Stabinger, S., Opitz, P., & Engl, S. (2019). *Domain Adaptation through BERT Language Model Finetuning for*.

Ruangkanokmas, P., Achalakul, T., & Akkarajitsakul, K. (2016). Deep Belief Networks with Feature Selection for Sentiment Classification. *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, *0*, 9–14. https://doi.org/10.1109/ISMS.2016.9

Sadia, A., Khan, F., & Bashir, F. (2018). An overview of lexicon-based approach for sentiment analysis. *International Electrical Engineering Conference, IEEC*, 1–6.

Sailunaz, K., & Alhajj, R. (2019). Emotion and sentiment analysis from Twitter text. *Journal of Computational Science*, *36*, 101003. https://doi.org/10.1016/j.jocs.2019.05.009

Sentiment Analysis using TextBlob | by Parthvi Shah | Towards Data Science. (n.d.). Retrieved August 31, 2020, from https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524

SENTIMENTAL ANALYSIS USING VADER. interpretation and classification of... | by Aditya Beri | Towards Data Science. (n.d.). Retrieved August 31, 2020, from

https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

Shalunts, G., Backfried, G., & Prinz, K. (2014). Sentiment analysis of German social media data for natural disasters. *ISCRAM 2014 Conference Proceedings - 11th International Conference on Information Systems for Crisis Response and Management*, *May*, 752–756.

Sharma, R., Nigam, S., Jain, R., Tech Scholar, M., Vidyapith, B., & Rajasthan, I. (2014). OPINION MINING OF MOVIE REVIEWS AT DOCUMENT LEVEL. *International Journal on Information Theory (IJIT)*, *3*(3). https://doi.org/10.5121/ijit.2014.3302

Shelar, A., & Huang, C. Y. (2018). Sentiment analysis of twitter data. *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, 1301–1302. https://doi.org/10.1109/CSCI46756.2018.00252

Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information . (2020). *January*.

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* (Vol. 15).

Sun, C., Huang, L., & Qiu, X. (2019). Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 380–385.

Talpada, H., Halgamuge, M. N., & Tran Quoc Vinh, N. (2019). An analysis on use of deep learning and lexical-semantic based sentiment analysis method on twitter data to understand the demographic trend of telemedicine. *Proceedings of 2019 11th International Conference on Knowledge and Systems Engineering, KSE 2019*. https://doi.org/10.1109/KSE.2019.8919363

Tang, D., Wei, F., Qin, B., Liu, T., & Zhou, M. (2015). *Coooolll: A Deep Learning System for Twitter Sentiment Classification*. *SemEval*, 208–212. https://doi.org/10.3115/v1/s14-2033

Tang, L., Bie, B., Park, S. E., & Zhi, D. (2018). Social media and outbreaks of emerging

infectious diseases: A systematic review of literature. In *American Journal of Infection Control* (Vol. 46, Issue 9, pp. 962–972). Mosby Inc. https://doi.org/10.1016/j.ajic.2018.02.010

The pandemic of social media panic travels faster than the COVID-19 outbreak. (2020). https://doi.org/10.1093/jtm/taaa031

Turney, P. D. (2002). *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. Retrieved September 1, 2020, from http://www.google.com

Twitter - Wikipedia. (n.d.). Retrieved August 29, 2020, from https://en.wikipedia.org/wiki/Twitter

Understanding Sentiment Analysis in Social Media Monitoring | Unamo Blog. (n.d.). Retrieved August 30, 2020, from https://unamo.com/blog/social/sentiment-analysis-social-media-monitoring

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Worldometer. (2020). Coronavirus Cases. *Worldometer*, 1–22. https://doi.org/10.1101/2020.01.23.20018549V2

Xu, H., Liu, B., Shu, L., & Yu, P. S. (n.d.). *BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis*. Retrieved September 1, 2020, from https://www.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. *NeurIPS*, 1–18. http://arxiv.org/abs/1906.08237

Yin, D., Meng, T., & Chang, K.-W. (2020). *SentiBERT: A Transferable Transformer-Based Architecture for Compositional Sentiment Semantics*. 3695–3706. https://doi.org/10.18653/v1/2020.acl-main.341

APPENDIX A

This section presents code, figures, tables and other work that was conducted as a part of the study but hasn't been included in the chapters of this report.

A.1 Mixout code used to change the linear layer to Mixlinear

```
import torch
from torch.autograd.function import InplaceFunction

class Mixout(InplaceFunction):
    # target: a weight tensor mixes with a input tensor
    # A forward method returns
    # [(1 - Bernoulli(1 - p) mask) * target + (Bernoulli(1 - p) ma
sk) * input - p * target]/(1 - p)
    # where p is a mix probability of mixout.
    # A backward returns the gradient of the forward method.
    # Dropout is equivalent to the case of target=None.
    # I modified the code of dropout in PyTorch.
    @staticmethod
    def _make_noise(input):
        return input.new().resize_as_(input)

    @classmethod
    def forward(cls, ctx, input, target=None, p=0.0, training=Fals
e, inplace=False):
        if p < 0 or p > 1:
            raise ValueError("A mix probability of mixout has to b
e between 0 and 1,"
                             " but got {}".format(p))
        if target is not None and input.size() != target.size():
            raise ValueError("A target tensor size must match with
 a input tensor size {},"
                             " but got {}". format(input.size(), t
arget.size()))
        ctx.p = p
        ctx.training = training

        if target is None:
            target = cls._make_noise(input)
            target.fill_(0)
        target = target.to(input.device)
```

```
        if inplace:
            ctx.mark_dirty(input)
            output = input
        else:
            output = input.clone()

        if ctx.p == 0 or not ctx.training:
            return output

        ctx.noise = cls._make_noise(input)
        if len(ctx.noise.size()) == 1:
            ctx.noise.bernoulli_(1 - ctx.p)
        else:
            ctx.noise[0].bernoulli_(1 - ctx.p)
            ctx.noise = ctx.noise[0].repeat(input.size()[0], *([1]
 * (len(input.size())-1)))
        ctx.noise.expand_as(input)

        if ctx.p == 1:
            output = target.clone()
        else:
            output = ((1 - ctx.noise) * target + ctx.noise * outpu
t - ctx.p * target) / (1 - ctx.p)
        return output

    @staticmethod
    def backward(ctx, grad_output):
        if ctx.p > 0 and ctx.training:
            return grad_output * ctx.noise, None, None, None, None
        else:
            return grad_output, None, None, None, None

def mixout(input, target=None, p=0.0, training=False, inplace=Fals
e):
    return Mixout.apply(input, target, p, training, inplace)
```

```
import math
import torch
import torch.nn as nn
import torch.nn.init as init
import torch.nn.functional as F
```

```
from torch.nn import Parameter

#from mixout import mixout

class MixLinear(torch.nn.Module):
    __constants__ = ['bias', 'in_features', 'out_features']
    # If target is None, nn.Sequential(nn.Linear(m, n), MixLinear(
m', n', p))
    # is equivalent to nn.Sequential(nn.Linear(m, n), nn.Dropout(p
), nn.Linear(m', n')).
    # If you want to change a dropout layer to a mixout layer,
    # you should replace nn.Linear right after nn.Dropout(p) with
Mixout(p)
    def __init__(self, in_features, out_features, bias=True, targe
t=None, p=0.0):
        super(MixLinear, self).__init__()
        self.in_features = in_features
        self.out_features = out_features
        self.weight = Parameter(torch.Tensor(out_features, in_feat
ures))
        if bias:
            self.bias = Parameter(torch.Tensor(out_features))
        else:
            self.register_parameter('bias', None)
        self.reset_parameters()
        self.target = target
        self.p = p

    def reset_parameters(self):
        init.kaiming_uniform_(self.weight, a=math.sqrt(5))
        if self.bias is not None:
            fan_in, _ = init._calculate_fan_in_and_fan_out(self.we
ight)
            bound = 1 / math.sqrt(fan_in)
            init.uniform_(self.bias, -bound, bound)

    def forward(self, input):
        return F.linear(input, mixout(self.weight, self.target,
                                      self.p, self.training), self
.bias)

    def extra_repr(self):
        type = 'drop' if self.target is None else 'mix'
        return '{}={}, in_features={}, out_features={}, bias={}'.f
ormat(type+"out", self.p,
            self.in_features, self.out_features, self.bias is not
None)
```

After defining the model with classifier layer added, convert the layer to mixlinear with mixout percentage.

```
model3 = XLNetForMultiLabelSequenceClassification(num_labels=len(Y
_train[0]))
for name, module in model3.named_modules():
    if name in ['dropout'] and isinstance(module, nn.Dropout):
        setattr(model3, name, nn.Dropout(0))
    if name in ['classifier'] and isinstance(module, nn.Linear):
        target_state_dict = module.state_dict()
        bias = True if module.bias is not None else False
        new_module = MixLinear(module.in_features, module.out_feat
ures,
                               bias, target_state_dict['weight'],
0.5)
        new_module.load_state_dict(target_state_dict)
        setattr(model3, name, new_module)
```

A.2 Training and Validation Batch wise

BERT with dropout and Mixout

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
1	50	1.037929	-	-	25.13
1	100	0.935576	-	-	25.96
1	150	0.811460	-	-	25.16
1	200	0.716476	-	-	24.96
1	250	0.643528	-	-	25.22
1	300	0.592527	-	-	25.20
1	350	0.571612	-	-	25.25
1	371	0.522583	-	-	10.28
1	-	0.745881	0.473372	82.93	209.27

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
2	50	0.457944	-	-	25.64
2	100	0.474602	-	-	25.14
2	150	0.443110	-	-	25.16
2	200	0.437218	-	-	25.08
2	250	0.455403	-	-	25.12
2	300	0.415825	-	-	25.26
2	350	0.421685	-	-	25.13
2	371	0.463276	-	-	10.23
2	-	0.444828	0.391676	85.99	208.66

Figure 24 - A.2 validtion loss and accuracy of BERT with dropout

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
1	50	0.952661	-	-	23.76
1	100	0.742218	-	-	23.95
1	150	0.702664	-	-	25.06
1	200	0.578745	-	-	25.80
1	250	0.543755	-	-	24.94
1	300	0.506545	-	-	24.91
1	350	0.473995	-	-	25.29
1	371	0.528935	-	-	10.32
1	-	0.637337	0.425634	83.43	206.07

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
2	50	0.359508	-	-	25.51
2	100	0.342826	-	-	25.08
2	150	0.330193	-	-	25.24
2	200	0.352443	-	-	25.19
2	250	0.317294	-	-	25.06
2	300	0.333143	-	-	25.20
2	350	0.301658	-	-	25.23
2	371	0.259943	-	-	10.27
2	-	0.329762	0.390717	85.47	208.78

Figure 25 - A.2 Validation loss and accuracy of BERT with mixout

BERT with dropout and mixout – sampled data

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
1	50	1.063172	-	-	46.95
1	100	0.905156	-	-	45.83
1	150	0.791193	-	-	45.68
1	168	0.776095	-	-	16.27
1	-	0.905378	0.650273	75.33	173.10

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
2	50	0.645211	-	-	46.58
2	100	0.597411	-	-	45.65
2	150	0.590114	-	-	45.62
2	168	0.583768	-	-	16.24
2	-	0.608224	0.556733	78.67	172.42

Figure 26- A.2 Val loss and accuracy of BERT with dropout after sampling

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
1	50	0.991671	-	-	46.90
1	100	0.781173	-	-	45.80
1	150	0.651324	-	-	45.65
1	168	0.676971	-	-	16.26
1	-	0.795181	0.596887	76.43	172.88

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
2	50	0.469435	-	-	46.49
2	100	0.473128	-	-	45.66
2	150	0.454175	-	-	45.71
2	168	0.489429	-	-	16.25
2	-	0.468143	0.544950	78.40	172.41

Figure 27 - A.2 val loss & accuracy of BERT with mixout- after sampling

A.3 Data Exploration

Bar plot for sentiment classes.

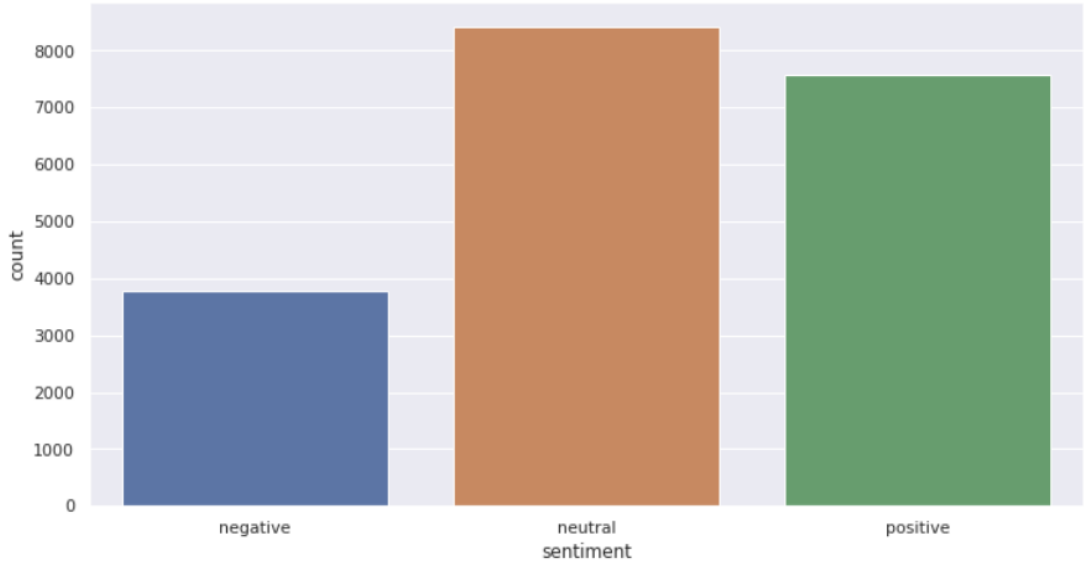

Figure 28 - A.3 Bar plot of sentiment counts

Histogram plot for Vader polarity:

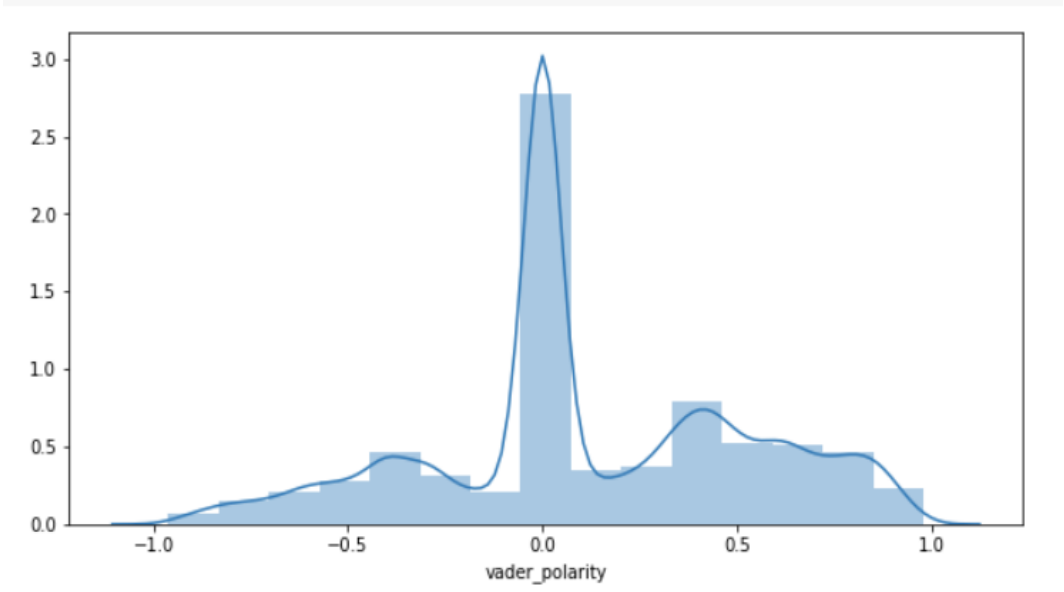

Figure 29 - A.3 Vader sentiment score -histogram plot

Histogram plot for Textblob polarity

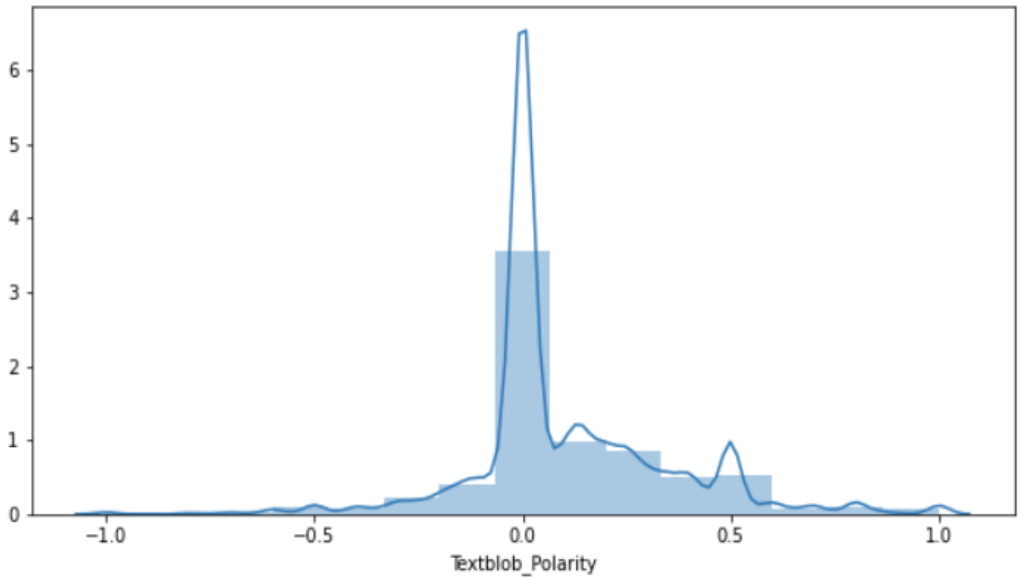

Figure 30 - A.3 Textblob sentiment score -histogram plot