Detection of Offensive Language in Social Media Posts.

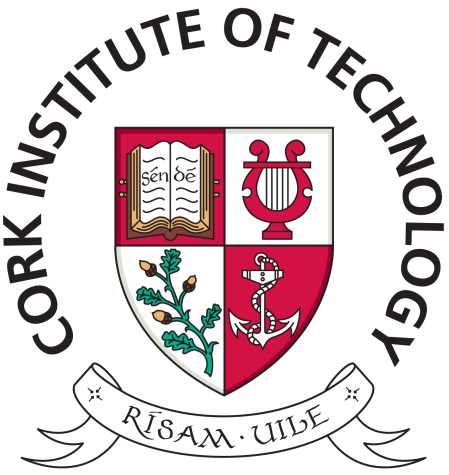
Thesis · May 2020

DOI: 10.13140/RG.2.2.23097.80485

2 authors:

Sidharth Mehra
Cork Institute of Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Mohammed Hasanuzzaman
Munster Technological university
44 PUBLICATIONS **97** CITATIONS
SEE PROFILE



Detection of Offensive Language in Social Media Posts

by

Sidharth Mehra

This thesis has been submitted in partial fulfillment for the degree of Master of Science in MSc in Artificial Intelligence

in the
Faculty of Engineering and Science
Department of Computer Science

May 2020

Declaration of Authorship

I, Sidharth Mehra , declare that this thesis titled, 'Detection of Offensive Language in Social Media Posts' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an masters degree at Cork Institute of Technology.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Cork Institiute of Technology or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
- I understand that my project documentation may be stored in the library at CIT, and may be referenced by others in the future.

Signed: _____

Date: _____

CORK INSTITUTE OF TECHNOLOGY

Abstract

Faculty of Engineering and Science
Department of Computer Science

Master of Science

by Sidharth Mehra

Posting offensive or abusive content on the social media have been a serious concern in the recent years. This has created a lot of problems because of the huge popularity and usage of the social media sites like Facebook and Twitter. The main motivation lies in the fact that our model will automate and accelerate the detection of the posted offensive content so as to facilitate the relevant actions and moderation of these offensive posts. We would be using the publicly available benchmark dataset OLID 2019 (Offensive Language Identification Dataset) [1] for this research project. The scope of our work lies in predicting whether the tweet post is offensive or not. We contributed by making the training dataset balanced using the Random Under-sampling technique. We also performed the thorough comparative analysis of various Feature Extraction Mechanisms and the Model Building Algorithms. The final comparative analysis concluded that, the best model came out to be Bidirectional Encoder Representation from Transformer (BERT). Our results outperforms the previous work achieving the Macro F1 score of 0.82 on this OLID dataset. Finally a real time system could be deployed on various social media platforms to detect and analyze the offensive post content and taking the appropriate action in order to normalize the behaviour on these sites and the society.
...

Acknowledgements

I'm grateful to Dr. Mohammed Hasanuzzaman for guiding me through out this AI research project. His valuable suggestions and encouragement helped me a lot in the development of this project and the thesis. Moreover I felt very fortunate enough to work under his expertise in the field of Natural Language Processing. As a project Supervisor, he played a crucial role in the success of this research project.

I am immensely obliged to all the lecturers, specially the Head of Artificial Intelligence Dr.Ted Scully from the Department of Computer Science at Cork Institute of Technology for teaching us the latest modules in the field of AI along with the best practices in programming.

I would like to pay my sincere regards and thanks to Mr. Nitin Seth- Vice Chairman at G.D. FOODS MFG.(I) PVT. LTD. for financially supporting my Master's here at Ireland.

I would also like to give my greatest thanks to my seniors of Artificial Intelligence Aniruddha Kalburgi and Praveen Joshi for helping me in the best way possible throughout this Master's and this research project.

Lastly, I offer my regards to all of those who supported me in any respect during the completion of my Master's thesis....

Contents

Declaration of Authorship i

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables vii

Abbreviations viii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Executive Summary . . . 2
- 1.3 Contribution . . . 3
- 1.4 Structure of This Document . . . 4

2 Background 5
- 2.1 Thematic Area within Computer Science . . . 5
- 2.2 Project Scope . . . 6
  - 2.2.1 Artificial Intelligence . . . 6
  - 2.2.2 Machine Learning, Deep Learning and Natural Language Processing 6
  - 2.2.3 Moving towards Offensive Language Detection . . . 11
- 2.3 A Review of the Thematic Area . . . 12
  - 2.3.1 Motivation and Challenges . . . 13
  - 2.3.2 Datasets Involved in Previous Work . . . 16
  - 2.3.3 Existing Methodology . . . 18
  - 2.3.4 Results Analysis and Evaluation Metrics . . . 21
- 2.4 Current State of the Art . . . 22

3 Problem - Offensive Language Detection in Social Media 25
- 3.1 Problem Definition . . . 25
- 3.2 Dataset Description . . . 26
- 3.3 Research Objectives . . . 27

4 Implementation Approach **29**
4.1 Experiment Design and Methodology 29
4.1.1 Data Pre-processing 29
4.1.2 Feature Extraction Techniques 31
4.1.2.1 Feature Extraction with Statistical Models 31
4.1.2.2 Feature Extraction with Word Embeddings 33
4.1.3 Handling Imbalance 34
4.1.3.1 Oversampling Techniques 34
4.1.3.2 Undersampling Techniques 35
4.1.4 Algorithmic Modeling 36
4.1.4.1 Machine Learning Based Models 36
4.1.4.2 Deep Learning Based Models 37
4.1.5 Evaluation Metrics 38
4.2 Implementation Details and Results 40
4.2.1 Process Flow and Implementation Details 40
4.2.2 Various Machine Learning Models with Different Feature Extraction Techniques 41
4.2.2.1 Feature Extraction with BOW 41
4.2.2.2 Feature Extraction with Tf-idf 42
4.2.2.3 Feature Extraction with Word2Vec 43
4.2.2.4 Feature Extraction with GloVe 43
4.2.3 Results of Various Deep Learning Models 44
4.3 Evaluation and Analysis 44
4.3.1 Comparative Analysis of Various Feature Extraction techniques along with Machine Learning and Deep Learning Models 44
4.3.2 Best Model for Offensive Language Detection 45

5 Conclusions and Future Work **46**
5.1 Conclusion 46
5.2 Future Work 47

Bibliography **49**

List of Figures

2.1 Different Areas Under Artificial Intelligence 7

2.2 Difference between Traditional Programming and Machine Learning . . . 8

2.3 Process Flow of Machine Learning 8

2.4 Categorization of Machine Learning 9

2.5 Artificial Neural Network 10

2.6 Categorization of Deep Neural Networks 11

2.7 Composition of Natural Language Processing 11

2.8 Relationship between AI, ML, DL and NLP 12

3.1 Distribution of 3 levels of annotation in OLID 27

3.2 WordCloud for Non-Offensive and Offensive Tweets 28

4.1 Feature Extraction through BOW 32

4.2 Training set distribution of class labels 34

4.3 Test set distribution of class labels 38

4.4 Representation of Confusion Matrix 39

List of Tables

4.1 Evaluation Metric for Feature Extraction with BOW (Uni-gram) 41

4.2 Evaluation Metric for Feature Extraction with BOW (Uni-gram + Bi-gram) 41

4.3 Evaluation Metric for Feature Extraction with BOW (Uni-gram + Bi-gram + Tri-gram) 42

4.4 Evaluation Metric for Feature Extraction with Tf-idf (Uni-gram) 42

4.5 Evaluation Metric for Feature Extraction with Tf-idf (Uni-gram + Bi-gram) 42

4.6 Evaluation Metric for Feature Extraction with Tf-idf (Uni-gram + Bi-gram + Tri-gram) 43

4.7 Evaluation Metric for Feature Extraction with Word2Vec 43

4.8 Evaluation Metric for Feature Extraction with GloVe 43

4.9 Evaluation Metric for Deep Learning Models 44

4.10 Comparative Analysis of Various Feature Extraction, Machine Learning and Deep Learning Techniques 44

Abbreviations

AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
DL	**D**eep **L**earning
NLP	**N**atural **L**anguage **P**rocessing
SVM	**S**upport **V**ector **M**achine
CNN	**C**onvolutional **N**eural **N**etwork
BiLSTM	**Bi**directional **L**ong **S**hort **T**erm **M**emory networks
GRU	**G**ated **R**ecurrent **U**nit
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
OLID	**O**ffensive **L**anguage **I**dentification **D**ataset
BOW	**B**ag **O**f **W**ords
SMOTE	**S**ynthetic **M**inority **O**versampling **T**echnique
TF-IDF	**T**erm **F**requency **I**nverse **D**ocument **F**requency

Dedicated to my Mother Mrs.Anita Mehra and Father Mr.Rakesh Kumar Mehra for providing the constant support and encouragement throughout this Master's degree.. . .

Chapter 1

Introduction

This introduction chapter mainly focuses on four sections. Firstly we will discuss about the motivation behind choosing this particular topic for our AI research project. Here we briefly explains the importance and the major societal impact of choosing this topic and making the contribution in the field. Secondly we will give a brief executive summary about our project describing about the main features along with the functional requirements. Next we will highlight our main contributions achieved in this particular problem area by explaining about the research objectives. Lastly, we present the structure of this thesis report representing what content is there in which chapter and corresponding sections.

1.1 Motivation

Increase in the usage of social media sites likes Facebook and Twitter have given the crowd a great platform to express their opinions/feelings for the individual, groups or events happening around them or in society. This digital media has become a great resource to share the information and also gives the full freedom of speech to everyone on the platform.

With the gaining popularity of these platforms; there also comes the negative part along with its benefits. This feature of the social media to express something openly to the world have created the major problems for these online businesses and negatively impacted the well being of the societal decorum. There are increasing cases of the abuse or offense on the social media like Hate speech, Cyber-bullying, Aggression or general Profanity. It is very much important to understand that this behaviour can not only immensely affect the life of an individual or a group but could be suicidal in some cases; adversely hampering the mental health of the victim/s.

This increasing negative situation on the internet has created a huge demand for these social media platforms to undertake the task of detecting the objectionable content and taking the appropriate action which can prevent the situation becoming more worse. This task of detecting the offensive content can be performed my human moderators manually, but it is both practically infeasible as well as time consuming because of the amount of the data generated on these social media platforms, therefore there is a need to fill this gap. Numerous studies in the past as evident by the following chapter 2 tries to tackle this problem and gap by leveraging the technologies like Artificial Intelligence, Machine Learning and Natural Language Processing. They are able to build the systems that can efficiently detect these types of offensive content so that the appropriate action could be taken as quickly as possible.

So in the last, considering the importance and the sensitivity of this particular topic in the today's digital world there is, still a lot of emerging further scope in tackling and improvise on the previous work done in the field with the help of these new age AI and NLP techniques. Therefore we aim to contribute in this particular direction.

1.2 Executive Summary

In this section of executive summary, we will try to summarize the work carried out in all the chapters of our project in some details. We will start off with some Introduction about the project area that will describe the motivation to take this particular topic. Then we will briefly describe what we have done in the Background chapter to discuss about the extensive review and analysis of the previous work related to our topic. Then we focus on the architecture and implementation approach that we proposed in this thesis. Finally we conclude this summary by the contributions achieved in this problem area of Offensive Language Detection.

The task that we have taken for this research project is to detect whether a given tweet is offensive or not with the help of NLP techniques along with ML and DL. The main motivation for choosing this topic is to develop an efficient system to detect this offensive content on the social media platform which can be removed afterwards. This will prevent the unease and negativity that may spread in society, people or any community group's mind.

The background chapter contains all the technical concepts related to AI, ML, DL and NLP which we have used for this project. We have included these details so that it becomes relatively easy for a person who has limited knowledge about the subject to progress towards the later chapters and sections of this document. In this chapter we

critically analysed about 15 academic research papers that revolve around our project topic. This study helped us to form an appropriate research direction and identify the important challenges in the field. This chapter also helped us to come up with the relevant dataset for the problem that is OLID 2019 dataset.

Further, we focussed more on the problem description and also mentioned the important features and the distribution of our OLID dataset. As this dataset is the recent one and not much work has been done on it, we laid down some objectives that we can achieve in order to make the significant contribution after the baseline results on this dataset. We divided our proposed solution methodology into mainly five steps. These were Text Pre-processing, Feature Extraction, Handling Imbalance, Model Building and Model Evaluation. In this chapter, we presented our results for the top performing techniques in each of the steps of this NLP pipeline.

As our main contribution to this field, we contributed by making the training dataset balanced using the Random Under-sampling technique that worked best for our problem. Moreover from our comparative analysis of various Machine learning and Deep learning algorithms, BERT performed the best and achieved the Macro F1 score of 0.82 on this OLID dataset. Therefore we made the small improvement in the field when compared with the previous work on this dataset and opened up a new pace to work more on this novel dataset.

1.3 Contribution

1. For this research project we researched on the recent OLID 2019 dataset which considered the problem as the whole by taking all the offense types into the account. As the dataset is new and not much work has been done on this, therefore to carry out our research to its full extent and explore new insights; we performed a thorough comparative analysis of various Feature Extraction Mechanisms and Model Building Algorithms.

2. We also contributed towards mitigating the imbalance in the training twitter data with the help of Random Under-sampling Technique.

3. Finally we developed an efficient system based on state of the art deep learning algorithm BERT to detect whether a given tweet is offensive or not. We achieved the Macro F1 score of 0.82 which clearly outperformed the performance of previous work [1] on the dataset.

1.4 Structure of This Document

Whole of this thesis document is mainly divided into 5 chapters.

Chapter 1- Introduction

This chapter comprises of four sections. First section talks about the motivation for this particular topic. Second section gives a brief executive summary about the project chapters. Third section focus on the main contributions achieved in this problem area. Finally we close our chapter with the last section which present the structure of this thesis document.

Chapter 2- Background

This chapter comprises of four sections. We start of with the first section of the overview of the Thematic area within Computer Science. Second section lays focus on the Project Scope. Third section is all about the Review of the Thematic Area. Lastly, we close this chapter by the Current State of the Art section.

Chapter 3- Problem (Offensive Language Detection in Social Media)

There are a total of three sections in this chapter. First section provides us the description of the problem topic. Second section gives the insights about the dataset we are going to use for this thesis work.Third and the last section focus on enumerating all the Research Objectives that we need to achieve for the meaningful contribution.

Chapter 4- Implementation Approach

This chapter is broadly divided into three sections. First section is the experiment design and methodology further dividing into the detailed subsections of the phases of our proposed approach. The second section lays focus on the implementation details and the results achieved. Finally we close this chapter with the evaluation and analysis of the results achieved in the previous section.

Chapter 5- Conclusions and Future Work

This chapter has only two sections. First section discusses about the Conclusion reached for this research project. Second section is about the possible Future Work in the problem domain.

Chapter 2

Background

This chapter starts with the thematic area within Computer Science which explains the concrete topic of the project along with the concrete areas and the main area of Computer Science under which our project falls. Next section deals with the project scope which explains all the basics of the topics and the areas discussed in the previous section. This is done to give a short technical overview of the domain knowledge to a person who knows nothing or little about the field. Further, we move on to the review of the thematic area which explains the previous work done related to our project topic in terms of motivation and challenges, datasets involved in the previous work, existing methodology, results analysis and evaluation metrics. Next section of current state of the art focus on critically analyzing the previous work to find the major similarities and differences between our project and each of these works. This will form a strong basis to contribute to this particular problem area.

2.1 Thematic Area within Computer Science

1. This research project is about the detection of offensive language on the social media platforms like Twitter or Facebook. This problem is very serious and increasing largely on these sites, therefore we aim to provide a solution for normalising these derogatory/abusive posts using algorithmic techniques in Machine Learning. This can effectively minimize the negative impact of these posts on the individual or a group.

2. The area under which our project falls is Natural Language Processing which is the art of understanding (Natural Language Understanding) and generating (Natural Language Generation) the human language such as text or speech by the machines in a similar manner like human beings. Here particularly we focus on

Natural language understanding which gives machines, the ability to understand the human text (which is in our case are offensive posts) and automate the process of removing its presence on the social media without the human intervention.

3. The main area of Computer Science under which our project falls is Artificial Intelligence which aims to inculcate the human-like behaviour into the machines and make them capable to think, act and interact as similar to human beings. Our project combines the techniques from Machine Learning, Deep Learning and Natural Language Processing (all falls under the umbrella of Artificial Intelligence) to accomplish the main objective.

2.2 Project Scope

2.2.1 Artificial Intelligence

"Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and reacts like humans."(Andrew Ng, 2015)

Artificial Intelligence is the branch under Computer Science that deals with the development of the machines that mimic the human behavior and intelligence. The main goal of an AI is to work and perform similar to the cognitive functions of a human brain. These functions are mainly decision making, problem solving and learning through the environment. There is an enormous progression in the generation of the data in the today's digital world. This is all facilitated through the web-based life, sensors, gadgets, online businesses and many more factors giving rise to the huge amount of structured, un-structured and semi-structured data. This explosion of data is giving rise to the new economy called data economy. New oil is the data which is precious but useful only when cleaned and processed. This data is the only driving source for the machines with which they learn and then perform similar to humans and derive the meaningful insights. There are many sub-fields under the hood of Artificial Intelligence which are depicted in the below figure 2.1.

2.2.2 Machine Learning, Deep Learning and Natural Language Processing

Our project combines the three areas of AI which are Machine Learning, Deep Learning and Natural Language Processing. Here we will look at the basics of each of the field to get the better sense of the project topic.

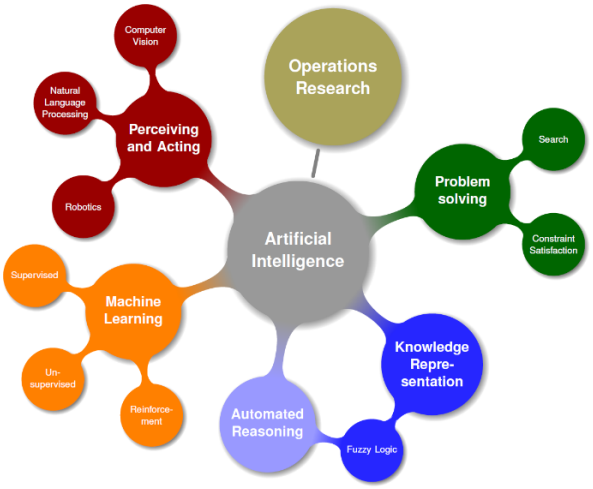


FIGURE 2.1: Different Areas Under Artificial Intelligence [2]

Overview of Machine Learning

"Machine learning is the science of getting computers to learn and act like human do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-word interactions"(Arthur Samuel, 2016).

Machine Learning is one of the most important application areas of Artificial Intelligence. Machines are developed in such a way that they learn from the enormous amount of data and get trained. Once our ML model is trained on this training data, it can make the predictions on the data examples which it has never seen. According to Stanford university definition which is "Machine learning is the science of getting computers to act without getting explicitly programmed".The widely known formal, social and scientific definition of machine learning was given by Mitchell (1997) is "A computer program is said to gain for a fact E regarding some class of errands T and execution measure P, if its execution at assignments in T, as estimated by P, improves with experience E". The task defines the objective of the problem. For example classification, regression, anomaly detection, etc. The sole base for Machine learning is the availability of the data and does not rely on any rule based programming. These algorithms by generalizing from examples can perform important tasks.There is the difference between Machine learning and Traditional programming. In traditional programming we give data and program as inputs to the machine and it will provide the output whereas on the other side if we talk about machine learning, the input to the system is data and output and the output of the machine is the program that have been learnt to make predictions on unknown examples points. Figure 2.2 illustrates the major difference between traditional approach and the approach followed by Machine Learning.

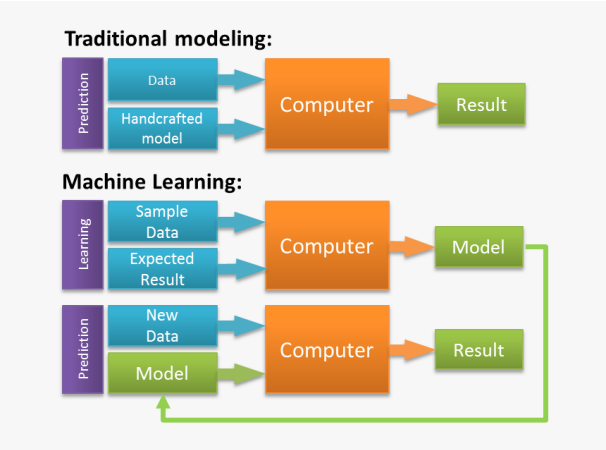


FIGURE 2.2: Difference between Traditional Programming and Machine Learning [3]

Basic Process Flow of Machine Learning

Below Figure 2.3 is the basic process that is carried out in most of the Machine Learning problem statements. We train our Machine learning model on the training data using the various algorithms available. The most common types of tasks that are associated with machine learning are Regression and Classification. Then our trained model makes the prediction on the unseen test examples and finally we evaluate the model performance.

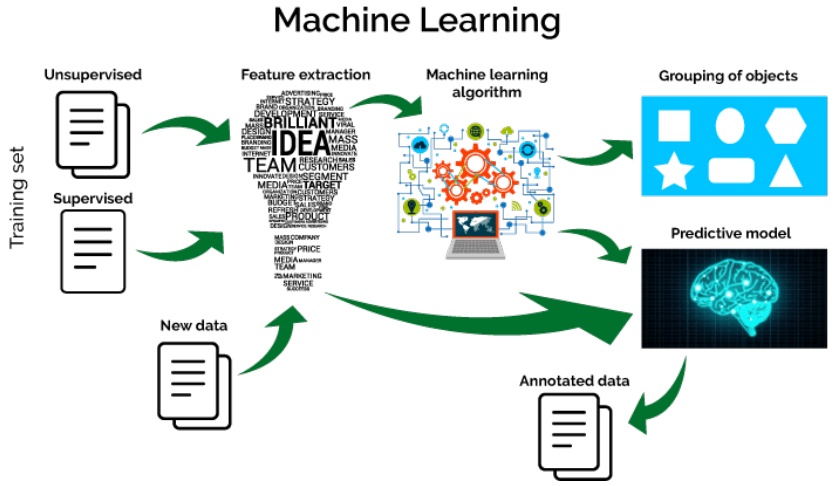


FIGURE 2.3: Process Flow of Machine Learning [4]

Types of Machine Learning Techniques

- **Supervised Learning:** These types of learning algorithms are most common in Machine learning space. In these type of techniques, we are given with the labelled outputs of each of the input training examples so that the model can learn the patterns and relationship between the inputs and those outputs. Some of

the examples of supervised learning algorithms are K-Nearest Neighbours, Linear Regression, Support Vector Machines, Logistic Regression and many more.

- **Unsupervised Learning:** In this type of learning technique, we are not given any input to output mappings in our dataset. The model will analyse the hidden patterns and groupings in the data to find the similarities and differences between each of the data points.
- **Reinforcement Learning:** This type of learning mechanism is based on the agent and environment. The program learns to make a sequence of decisions based on the reward and penalty model and try to maximize the reward.

Below figure 2.4 shows the three different types of Machine learning techniques that are widely used.

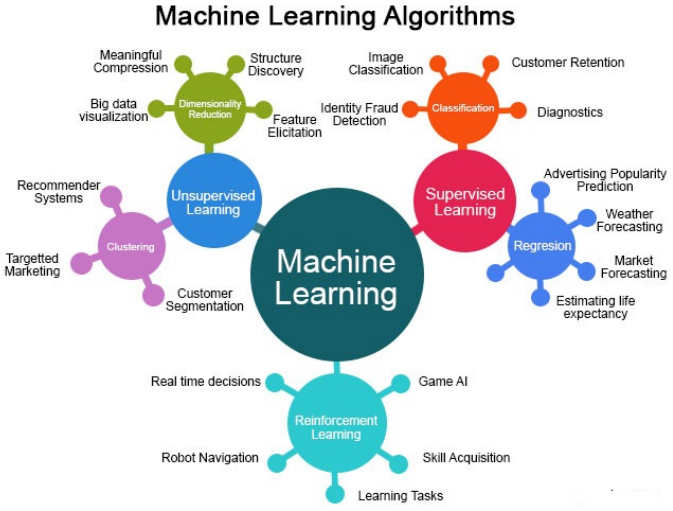


FIGURE 2.4: Categorization of Machine Learning [5]

Overview of Deep Learning

Deep Learning is the field of Artificial Intelligence that comes under the domain of Machine Learning which we have seen above in good details. Deep Learning can be defined as the sub-set of Machine learning that is inspired by the biological structure and function of human brain containing over billions of neurons. These are also known as Deep Neural Networks as they contain many layers of neurons which are capable to recognize patterns from the raw input data and perform the decision making in the same way as the human mind performs. Every Deep Neural Network learns the patterns from the data through the technique called Back-propagation. We have mainly three types of layers in an Artificial Neural Network which are Input Layer, Hidden Layers and Output Layer, each of them are briefly introduced below.

- **Input Layer:** All the neurons in the input layer correspond to each of the feature in the dataset. The values of each instance are the inputs to this layer.
- **Hidden Layers:** The values from the input layer multiplied by the weights are the input to the hidden layer neurons to get the pre-activation outputs. Then we pass these pre-activation outputs to an activation function to get the final outputs from the hidden layer.
- **Output Layer:** The activated outputs from the hidden layers acts as the input to the neurons in the output layer which are multiplied by the weights to the get the pre-activation outputs. Then we pass these pre-activation outputs to the activation function to get the final outputs from the output layer.

Same as Machine learning, Deep learning can be also classified into three categories Supervised learning, Unsupervised learning and Reinforcement learning. Below figure 2.5 shows the structure of an Artificial Neural Network with one input layer, two hidden layers and one output layer.

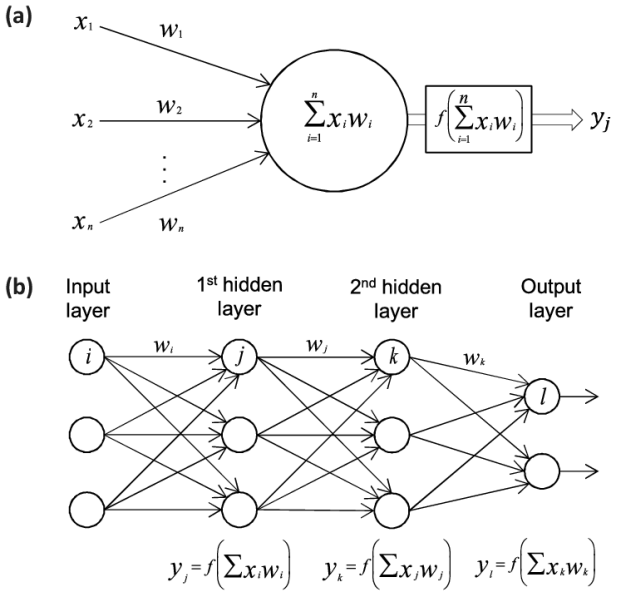


FIGURE 2.5: Artificial Neural Network

Different types of Deep Neural Networks

Depending on the use-case of the problem that we intend to solve with Deep Learning and the type of available data, there are various kinds of Deep Neural Networks that are meant to tackle that particular use-case. For example an Artificial Neural Network may work well on the numerical kind of data but on the other hand a Convolutional Neural Network may perform better on the images. Below figure 2.6 shows a whole host of Neural Networks that exists for solving a particular type of problem.

FIGURE 2.6: Categorization of Deep Neural Networks

Overview of Natural Language Processing

Natural Language Processing is the sub-field that takes the inspiration from the areas of Artificial Intelligence and Linguistics. It enables the computers/machines to process and analyze the large amount of human language data such as speech or text. We can think NLP as the computer programs that can understand and generate the human understandable language to derive meaningful insights. NLP can be further classified into Natural Language Understanding and Natural Language Generation which can be shown by the below figure 2.7

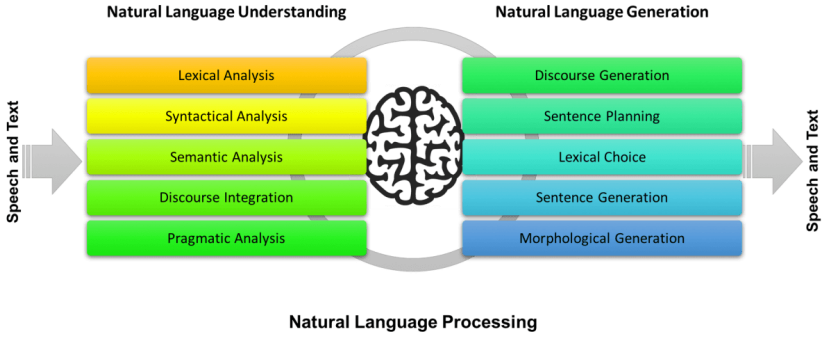


FIGURE 2.7: Composition of Natural Language Processing [6]

Below figure 2.8 is the relationship between AI, ML, DL and NLP. It is quite evident from the Venn Diagram that all ML, DL and NLP comes under the hood of AI and also DL is the sub-set of ML.

2.2.3 Moving towards Offensive Language Detection

Now we have seen the basic concepts and ideas revolving Artificial Intelligence, Machine Learning, Deep Learning and Natural Language Processing; we will now focus on understanding the specifics of our project. The topic of our project is the Detection of Offensive Language in Social Media which encompass these fields AI, ML, DL and NLP.

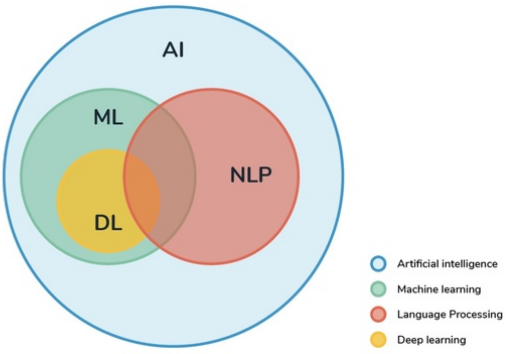


FIGURE 2.8: Relationship between AI, ML, DL and NLP [7]

The problem can be simplified as to detect whether the posts on the twitter are offensive or not and if they are then they can be removed from the platform to minimize the social instability. We as humans are able to detect whether the post is offensive or not and here we are fusing this capability into machines so it relates well with AI. As it deals with understanding of the human written text by machines then this makes it a NLP problem too. Lastly we need to understand how the ML and DL applies to our problem statement. A system or a computer program will be trained on the various offensive and non-offensive tweets so that it recognizes the patterns from them to distinguish between the two. Now in future our Machine Learning or Deep Learning model can accurately classify the unseen tweet as offensive/non-offensive and further action could be automated.

2.3 A Review of the Thematic Area

In this section we would be critically analysing the few academic papers that revolve around the detection of offensive language in social media using machine learning. We would be extracting the summaries of these papers along with their main characteristics which are defined with the proper structure described in the later part of this chapter. We will also determine the common and different points between these papers which will help us to extract the structure and the context of the problem solution. After analysing these papers that are formulated according to the below structure, we will develop a similar structure to illustrate the context of our problem solution. We will also compare the existing methods and try to identify that is there any clear best solution or are there any trade-offs. The main characteristics of these papers are structured as the individual sub-sections of this section following a chronological order.

2.3.1 Motivation and Challenges

The motivation, challenges and the direction of mitigation are mentioned below for each of the related works

Paper [8] was one of the first significant works that was related with detecting cyber-bullying on the social media with the help of NLP. Earlier this problem was studied mostly by the psychiatrists and social scientists. Authors believed that anonymity and lack of meaningful supervision has put out this topic into more attention to computational linguists. They moreover believed that the appropriate action on this content through NLP could prevent the tragic outcomes of bullying on social media platforms. Authors modeled this problem into 2 parts. First one is to determine whether the given comment is sensitive or not making it a binary classification task. And if the comment is sensitive then classify it as the comment revolving around one of the areas in sexuality, race culture or intelligence which made this as multi-class classification subtask.

The very strong baseline paper [9] aimed to tackle the serious issue of cyber-bullying on social media platforms using NLP techniques so that the further investigation on the same domain could be explored in future by the NLP community with the help of their baseline results. They moreover formulated the bullying on the social platforms into 4 major NLP tasks namely text classification, role labeling, sentiment analysis and topic modeling. They defined the bullying traces as the posts by the individuals who have been a victim of bullying. Objective of subtask A was to distinguish the bullying episodes from the non-bullying traces in the dataset examples. The subtask B dealt with the role labelling of the tweets into Author's role and Person mention role which can be thought of a binary text classification task. The other thing in this subtask is to further sequentially tag these roles into one of the 5 categories which are Accuser (A), bully (B), reporter (R), victim (V) and other (0). Subtask C was related with classifying the sentiment of the bullying episode to understand their motivation. This task was a binary task of classifying the tweet as teasing or not. Subtask D was related with latent content modelling which extracts the main topics to better understand the bullying traces.

The authors in [10] realized the need of automating and improving the task of detecting the bullying content on the social media. Authors clearly outlined that the previous work has only investigated the bullying detection only from individual comments. They strongly felt that taking user characteristics and profile information could definitely yield in better model for characterizing the sensitive content.

Authors in [11] believed that they were among the first one to investigate the topic of hate speech detection on social media particularly when it comes to racist comments

which were proliferating hugely on these online platforms because of freedom of speech and the features of re-tweeting. They analysed the racism against the black as they constituted 25% of the whole twitter population at that time and they were the highest one to face the racism on the platform. They formulated the problem as the binary classification task to determine whether the tweet is racist or non-racist.

The motivation of authors in the work [12] was triggered by the murder of Drummer Lee Rigby in Woolwich, London, UK. They observed and believed that immediately after such events there is a potential opportunity of spreading the hate speech on online platforms. They formulated this problem as the binary classification task of predicting whether the tweet is hateful/ antagonistic or not with a focus on race, ethnicity or religion. They derived their features from the context of each tweet and experimented with various probabilistic, rule-based, and spatial-based classifiers with a voted ensemble meta-classifier. They believe that their contribution is closely related to the effective decision and policy making.

In [13], the authors took the problem of detecting hate speech on social media using user comments. The motivation of their work was developed because of increasing cases of hate on the online platforms causing the decline in the online business and the user experience. They aim to develop a low dimensional representation of comments using neural language models which they would be sending to classification algorithms. They addressed the problem of high dimensionality and sparsity due to previous work of BOW model and aimed to achieve accurate predictive models for the domain.

Authors in [14] were the first one up till 2016 to focus on the problem of detecting the offensive language on online platforms rather than just focussing on the specific type of abuse such as cyber-bullying or hate-speech as evident in the past years. They brought out the attention to the drawbacks of regular expression and black-lists in the extreme cases of hate speech which are more subtle and less obvious than regular ones. They tend to outperform the state of the art deep learning approach to build a predictive model and also created a new corpus of user comments as a unique one of its kind.

Authors in [15] raised the challenge of separating the hate speech with other types of offensive content as the previously used lexical methods failed to distinguish between the different offense types. It is important to accurately identify the hate speech particularly from the other offensive content as both tend to have the different implications on the society and the individuals. They worked with twitter data that was labelled into 3 categories namely hate speech, offensive language and neither of these two for which they trained multi-class classifier.

In [16] the authors extended the main challenge from [15] to distinguish the hate speech from general profanity or offense to come up with a lexical baseline for the same.

The work presented in [17] aimed to identify and empirically analyse the relationship; particularly similarities and differences between the different subtasks that come under the offensive language detection. They also proposed a topology that could help the researchers for data annotation and feature construction.

Authors in [18] believed that the offensive content is increasing at an unprecedented rate on the online platforms which have potential to hamper the mental lives of individual and groups leading to a negative effective in the society. They believed that the manual moderation is practically impossible with this rapid generation of the data. Therefore they introduced this problem at TRAC workshop in 2018 and modelled the problem as the multi-class classification task of differentiating the posts and the comments into Overtly Aggressive, Covertly Aggressive, and Non-aggressive texts. They also identified the important challenges in the shared task mainly revolving around the annotation and the language issues.

The related work presented in [19] also aims to distinguish hate speech from the regular type of offensive content / profanity which has not been studied much in the previous work. They used the n-grams, skip-grams and clustering based techniques for the feature representations. They analysed that the previous work formulated the task of hate speech detection as the binary classification problem in which the systems are likely to misclassify the instances as non-offensive that did not contain any objectionable words but are actually offensive in a semantic and deeper sense. They also focussed on ensemble classifiers rather than the single classifiers which were mainly used in the past studies. The work considered that the opinion variation of the annotators is also an issue for this problem which needs to be resolved.

The authors in [20] dealt with the topic of offensive language on the German tweets which was announced as the shared task in GermEval 2018 competition. The problem was formulated into 2 subtasks. First one is the binary classification task to differentiate between the offensive and non-offensive German tweets. The second one is the further classification of offensive tweets into profanity, insult, abuse and other.

Authors in [21] believed that the text with high toxicity on the internet can cause personal attacks, bullying behaviours, threatening and harassment. They utilized the Wikipedia dataset released under Kaggle competition of toxic comment classification which was modelled as the 6-class classification problem. They employed CNN for the learning the structure of words in the document due to the advances in hardware and cloud computing.

The work in [1] considered the problem of the offensive language detection on the social media as the whole rather than focussing on the specific type of abusive content on the web as evident by the previous studies. They came up with the new dataset with three level of annotation scheme providing an opportunity for the researchers to delve deeper into the topic. Subtask A is to determine whether the post is offensive or not. Subtask B further identifies the type of the offense content whether it is targeted or untargeted. Subtask C then categorizes the target of the offensive post into either one of 3 classes namely individual, group and other.

2.3.2 Datasets Involved in Previous Work

Related work [8] used the labeled corpus of 4500 YouTube video comments that were scrapped from web. The sensitive examples from the dataset were labeled into one of the 3 categories namely sexuality, race culture and intelligence.

In paper [9], the authors used the sampled version containing labeled 1762 tweets from the TREC corpus developed in 2011.

At that time as no dataset for the bullying detection was publically available so authors in [10] scrapped the comments from the top 3 videos of different categories found in YouTube movies. The final dataset contained 4626 user comments consisting of 3856 distinct users and were manually labeled as bullying and non-bullying. Comment history of each user was also recorded and on an average the dataset contained 54 comments per user profile.

Authors in [11] prepared a balanced dataset from the twitter accounts that contained 24582 carefully labeled tweets by annotators as racist or non-racist.

The authors in [12] collected the data from twitter API within the 2 week window after the murder of the drummer as they believed that majority of the hate comments bound to happen in the 2 weeks' time and gradually decline. Therefore they wanted to capture this immediate reaction. A total of 450,000 tweets were captured during the study window and labeled as 'Yes' or 'No' meaning that the comment is hateful or not.

In [13], the authors prepared the largest dataset available till 2015 for hate speech detection for over 6 months and named as WWW-2015 dataset. They collected the comments from the Yahoo Finance website containing a total of 951,736 user comments out of which only 5% were hate speech comments and rest 95% of the comments were clean without the use of any offensive language. This was therefore modelled as the binary classification task as well.

The dataset in [14] was the extended version of the one used in [6]. It contained the user comments from the Yahoo finance as well as the Yahoo news adding the diversity to it because of presence of all types of abusive language like hate speech, profanity or derogatory language. They marked each user comment as the "abusive" or "clean" making it a binary classification problem. In the news dataset only 16% of the data was abusive and in the finance dataset only 7% of the comments were abusive.

Authors in [15] prepared the dataset from twitter API that contained 24,802 tweets each of them labeled as hate speech (5%), offensive (76%) or neither of these two (19%).

The authors in [16] used the open source dataset containing 14,509 English tweets which are labeled as HATE (16%), OFFENSIVE (33%), or OK (51%) reflecting the absence of any offensive content.

The dataset in [18] contained roughly 12,000 training comments from Facebook. For the development they provided the participants with 3000 comments in English and Hindi language respectively. Each comment was annotated with a label indicating the levels of aggression namely Overtly Aggressive (OAG), Covertly Aggressive (CAG) and Non-Aggressive (NAG) making it a multi-class classification task. The test set contained 916 and 970 comments in English and Hindi respectively.

The authors in [19] used the same dataset as in [16] making it a 3-class classification problem of distinguishing between hate speech and general profanity.

The dataset in [20] contained over 8500 annotated tweets which were labeled as OFFENSE (33.7%) and OTHER (66.3%). Each offensive tweet was further annotated as ABUSE (20.4%), INSULT (11.9%), PROFANITY (1.4%) and OTHER (66.3%).

Authors in [21] used a dataset having Wikipedia comments such that each comment is annotated into either of 6 classes namely toxic, severe toxic, obscene, threat, insult and identity hate.

In the work [1], the authors came up with a new dataset OLID for the offensive language detection in social media. At level A, each tweet was assigned a label as Not Offensive (NOT) or Offensive (OFF). At level B, the offensive tweets were annotated by either of 2 labels namely Targeted Insult (TIN) and Untargeted (UNT). At level C, the targeted offensive tweets were given an either of 3 labels namely Individual (IND), Group (GRP), Other (OTH).

2.3.3 Existing Methodology

Authors in [8] applied the standard pre-processing techniques like stemming, and removal of stop-words and unimportant characters from the textual data. They experimented with three classifiers Naïve Bayes, SVM and Decision Tree with 10-fold cross validation. They divided their feature space into general features and label specific features. For the general features they used TF-IDF and for the specific features they used unigrams and bigrams.

In [9] the authors trained the model to detect and analyze the cyber-bullying on the social media platforms. They divided their work into 4 subtasks. For the subtask A, after the regular pre-processing techniques they applied three techniques for the featurization namely unigram, unigram+bigram and POS colored unigram+bigram. For the classification task they experimented with Naive Bayes, SVM with linear kernel, SVM with RBF kernel and Logistic Regression with 5 fold cross validation using the WEKA implementation. In subtask B, for the author's role they used the same classifiers as the subtask A with 10-fold cross validation and also tuned the best model jointly with 5-fold cross validation with grid search CV. For categorizing the person mention role into respective categories they used the named entity recognition and trained the linear CRF and SVM respectively with 10-fold cross validation. In subtask C, they used the same feature representation, classifiers and parameter tuning as for the previous 2 subtasks with 10-fold cross validation. They used LDA as well as its variational inference implementation as their exploratory tool to discover the relevant topics from the bullying trace in the subtask D.

Authors in [10] used three feature sets to train the cyber-bullying classifier which were content-based, cyber-bullying based and user-based features. For the pre-processing they removed all the stop words and applied stemming to their dataset. They trained a Support Vector Machine to classify the bullying comments and non-bullying comments with 10-fold cross validation.

To deal with the problem of hate against black community on twitter the authors in [11] trained a Naïve Bayes classifier to able to classify the new tweet as racist or nonracist. They pre-processed the dataset by eliminating the URL's, mentions, stop words and punctuation along with lowercasing and replacing the wrong spellings with the correct ones. Authors found that 86% of the tweets that were racist only because they contained the offensive words so they preferred unigram model to featurize the training data.

Work in [12] was a binary classification task of predicting whether the comment is hateful or not. They followed a pipeline starting from Data collection and annotation, Feature selection, Data pre-processing, Feature preparation and finally Model selection. They

realized that the offensive words from the tweet could be the important features so they utilized the frequency of occurring of unigram and bigram. As the offensive tweet contain the certain instances following a particular pattern therefore for the extraction of typed dependencies within the tweet text they employed a Stanford lexical parser along with a context free lexical parsing model which represented the syntactic grammatical relationship in a sentence that are used a important features for the classifier. They came out with more common sense type of reasoning approach for this feature extraction phase. For the pre-processing phase they followed a generalized pipeline of tokenization, lowercase conversion, removal of stop words and alphanumeric characters, stemming. To preserve the context of words and the surrounding they employed unigrams to trigrams. They experimented with the 2 approaches of n-grams and collection of derogatory or hateful terms to check the contribution of other terms in determining the strong predictors. They ran a Bayesian Logistic Regression using all the typed dependencies features and came up with the vector representation of the tweet containing list of ngrams that included words, typed dependencies or combination of both. They used the three classifiers Bayesian Logistic Regression, Random Forest Decision Tree and Support Vector Machine for this binary classification task. They also employed the meta voting ensemble classifier made from these classifiers.

After applying the standard pre-processing techniques, they [13] divided their work into two parts for the detection of hate speech from the user comments. First they employed paragraph2vec to learn the distributed representation of comments and words using the neural language model of the continuous BOW (CBOW). This produced a low dimensional embeddings where the semantically similar comments resided in the same part of the space. Secondly a logistic regression classifier was trained on these embeddings to classify the type of user comment as hateful or clean.

Authors in [14] used the Vowpal Wabbit's regression model to measure the different aspect of the user comments using NLP features. They divided their features into 4 categories which were N-grams, Linguistics, Syntactic and Distributional Semantics. Due to noise found in the data they performed some mild-preprocessing for the first three features but did not performed any normalization for the fourth feature.

In [15], the pre-processing part was undertaken as to convert the tweet into lowercase and performed stemming through the porter stemmer. After that they featurized the tweets as weighted TF-IDF unigrams, bigrams and trigrams followed by the construction of the POS tagging using NLTK. They used Flesch-Kincaid Grade Level and Flesch Reading Ease scores to capture the quality of each tweet and also assigned the sentiment scores to each of the tweet. For the hashtags, mentions, retweets and URL's, they included binary and count indicators and for the number of characters, words and syllables, they

included features. They tried various models in Scikit-learn like Logistic regression, Naïve Bayes, Random Forest, Decision Tree and Linear SVM to train the model using 5-fold cross validation along with L1 regularization to reduce the dimensionality of the text data. They also performed the grid search parameter tuning to find the optimal parameters.

Authors in [16] used a LIBLINEAR SVM implementation for this multi-class classification task which has proven to be very effective on Native language identification and temporal text identification. For the features they used character n-grams, word n-grams and word skip-grams.

In [17], the authors defined their topology based on the prior work in the field of detection of different types of abusive language. They considered a 2 fold approach where the first aspect is to analyse the target of the abuse and another aspect is to analyse the degree to which the abuse is explicit. They also laid the implications of this topology on the annotation and the modeling of this problem. They suggested that the data annotation strategies should be dependent on the type of the abuse that is intended to be identified. On the other hand to select the most relevant features for the modeling, it is important to identify whether the abuse is directed, generalized, explicit or implicit.

In TRAC workshop proceedings [18], there were a total of 30 teams who submitted their systems for English and Hindi Language. Participants applied various techniques like LSTM, CNN, SVM, BiLSTM, Logistic Regression, Random Forest and many more to classify the English and Hindi Facebook comments.

The participants of the shared task of GermEval [20] used tokenization, POS-tagging, lemmatization and stemming and parsing as the methods for tweets pre-processing. They used SVM, Logistic Regression, Naïve Bayes, CNN, LSTM, GRU and the combination for the classification of the tweets.

The authors in [21] compared word embeddings and CNN against the BOW approach with the classifiers such as SVM, Naïve Bayes, k-NN and LDA that were applied on the Document Term Matrix.

Related work as evident in [1] applied various machine learning and deep learning techniques for each of the subtask within the problem domain. They first applied linear SVM trained on word unigrams followed by BiLSTM with the softmax activation function in the final layer with FastText embeddings. Finally they also experimented with CNN on this dataset.

2.3.4 Results Analysis and Evaluation Metrics

In [8] the authors achieved the best accuracy of 80.2% with the help of rule based JRIP whereas kappa measure was highest for SVM with the average value of 0.75.

In paper [9], for the subtask A the best model came out to be Linear SVM with a combination of unigrams and bigrams achieving an F-measure of 0.77. For the subtask B they achieved the cross validation accuracy of 61% with SVM linear along with the combination of unigram and bigram for the author's role. For the person mention role linear CRF outperforms the SVM achieving the accuracy of 87% and F-measure of 0.47. For the subtask C the best validation accuracy of 89% is achieved by linear SVM. For the subtask D, LDA discovered 5 topics from the bullying post namely feeling, suicide, family, school, verbal bullying and physical bullying.

The results from the work in [10]clearly indicated that the detection accuracy was boosted upon adding the bullying specific features and the user context. The model achieved the F-measure of 0.64 when they used all the three feature space to train the model.

Authors in [11] achieved a 10-fold cross validation accuracy of 76% using the Naïve Bayes classifier and an error rate of 24%.

Results as evident from [12] indicated that the most efficient features proved to be a combination of ngram typed dependencies and hated terms. All the classifiers BLR, SVM, RFDT and Ensemble performed equally well on the test set with an F-measure of 0.77

In [13] the authors achieved the Area Under Curve (AUC) value of 0.80 with the paragraph2vec representation trained on a logistic regression classifier.

In [14] the authors achieved the best F-measure of 0.79 on the finance data and 0.81 on the news data when trained considering all of the features rather than the selective features.

Results from the work in [15] indicated that Logistic regression with L2 regularization performed significantly better than the other models with the F1 measure of 0.90.

In [16] , the authors achieved a good accuracy of 78% with the character 4-gram model with Linear SVM evaluated using the stratified 10-fold cross validation.

In the shared task of TRAC [18], the best ranked team used LSTM with the data augmentation strategy. They used a combination of CNN and RNN on the surprize twitter

dataset for the feature representation. The team performed the spelling correction, emojis conversion and sentiment score computation as the part of text pre-processing and achieved F-score of 0.64 for both the Hindi and English on the Facebook test set. They also managed to come up as the team that significantly performed better on the twitter dataset despite being trained on the Facebook dataset achieving the F-measure of 0.60 and 0.50 for English and Hindi respectively.

The results from GermEval [20] displayed that the top performing systems came out to be CNN and LSTM for the both the shared tasks. They achieved an average macro F1 score of 76.77% and 52.71 % for the binary-class coarse-grained subtask A and multi-class fine-grained subtask B respectively.

Authors in [21], for the 6-class classification problem of toxic comments into respective categories achieved a descent accuracy of 91.2% with CNN.

In [1], as the dataset is fairly imbalanced for each of the levels so the authors used per class precision, recall and F1-score with the weighted average to compute the performance of each of the models. For the detection of the offensive language, CNN came out to be the best model with macro average F1 score of 0.80. For the categorization of the offensive language, CNN was the best model with the average macro F1 score of 0.69. In the identification of the target of the offensive posts, both CNN and BiLSTM performed equally well achieving an F1-macro of 0.47.

2.4 Current State of the Art

In this section we would be critically analyzing the above discussed related work on the detection of abusive/ offensive language on the various online platforms. This will eventually lead and form a strong basis to our contribution in the problem area through this research project.

The work presented in [10], [8], [9] focussed on the detection of **Cyber-Bullying** on the social media platform like Youtube and Twitter. Some of them modeled this as the binary classification problem of a content being sensitive or not while one work took the user characteristics and profile information into the account. One work also formulated the problem as sentiment analysis and topic modeling.

Related work [21] focussed mainly on the detection and classification of **Toxic Comments** into 6 respective categories namely toxic, severe toxic, obscene, threat, insult and identity hate.

Majority of the related work exists in the detection of the **Hate Speech** because of its large presence on the social media when compared to the other types of the offensive content on these platforms. The work presented in [12], [13], [11] are carried out with the same objective. Some formulated the problem as the binary classification of distinguishing racist content from the non-racist one, while some of them posed it as differentiating the hateful comments from non-hateful/clean ones based on race, ethnicity and religion. Work presented in [15], [16], [19] primarily contributed in distinguishing the hate speech from general profanity and modeled the problem as 3-class classification task with the annotation of the instances as Hate, Offensive or Clean.

The work in [18] was the identification of the **Aggression** on the social media particularly the English and Hindi language comments of Facebook and Twitter. The contribution formulated the idea as the 3-class classification task of classifying the posts into 3 categories namely non-aggressive, covertly aggressive and overtly aggressive.

Related work presented in [14], [20], [1] focussed on the detection of **Offensive language** rather than the specific type of offence or the abuse on these platforms. Some modeled the problem as the binary classification task of categorizing the content as abusive or clean while other focussed on the further multi class classification of the detected offensive German tweets as the profane, insult or abuse. The most recent work came out with the very suitable dataset for the detection of offensive language with the 3 levels of annotation scheme helpful in identifying the type and the target of the offensive posts for an in-depth analysis and moderation.

Upon analyzing each of the above work we can reach to the conclusion that the previous studies revolving around the detection of abusive/offensive content on the social media platforms can be classified into 5 main categories. These were mainly Cyber-bullying detection, Hate speech detection, Aggression identification, Toxic comment classification and the last but not least is the detection of overall Offensive content that includes all these previous types.

We strongly believe that **OLID dataset** [1] aims to capture the differences and similarities between the pre-existing datasets revolving around the above mentioned 5 tasks of offensive language detection. This dataset also treats the problem as the whole because it covers all these 5 aspects which was not evident in the previous studies. It also at the same time enable the identification of the type and the target of the offensive tweets for the further analysis.

As the OLID dataset was recently released in 2019, therefore it opens up a new opportunity for the researchers to explore this novel dataset for the further improvement in

the field. We also aim to contribute and come with the different techniques in order to make significant improvements after the work carried in [1]

Chapter 3

Problem - Offensive Language Detection in Social Media

This chapter introduces us with the problem definition and description. This section particularly give a brief about the motivation for choosing the topic of offensive language detection on social media and the reasoning behind choosing the OLID dataset for our problem followed by choice of sub-task withing the dataset that we are opting for our research project. Next section will give a detailed description about the OLID dataset giving us the insights about the features and the target class distribution. Finally we end this chapter by the research objectives that we need to achieve to successfully contribute in this problem area.

3.1 Problem Definition

Now a days with the increasing usage of social media platforms like Facebook and Twitter, we often see people to misuse this freedom of speech. Some of them try to use this platform to post the offensive or abusive things about a person or a group. This in turn can negatively impact the mental health of a community/ group or an individual. Considering the sensitivity of the topic, we aim to tackle this problem of detecting the offensive language in social media through cutting edge techniques in Machine Learning, Deep Learning and Natural Language Processing.

To proceed further with our topic we choose the Offensive Language Identification Dataset (OLID) which was released in 2019. But the question is that, why we choose this particular dataset when there exists various datasets in the problem domain. The previous datasets only aims to capture a particular type of offense such as hate speech

or cyber bullying, but OLID is a well diverse dataset which covers all the offense types. Therefore we choose this recent novel dataset which opened up various new research opportunities to contribute in.

There are three sub-tasks for this new dataset which involved the detection, predicting the type and the target of the offensive tweet respectively. We particularly focus on predicting whether the post is offensive or not which is the first and the most crucial sub-task for this dataset making it a binary classification problem.

3.2 Dataset Description

To carry out the research project we would be using the publicly available benchmark dataset [1] which is named as OLID (Offensive Language Identification Dataset). This dataset generalizes the task of detecting the all types of offensive content that was described in the background chapter using its three level annotation scheme. This types of dataset is very first of its kind and opens up the new opportunity to explore it even further which we also aim to do in our research project.

This OLID dataset annotate each tweet instance/post with the 3 level annotation scheme meaning that each instance is labeled with 3 corresponding type of labels. The first level denotes whether the tweet is offensive (OFF) or not (NOT). The second level identifies the type of the offensive tweet which could be targeted insult (TIN) or un-targeted (UNT). The third level identifies the target of the offensive post categorized mainly into individual (IND), group (GRP) or other (OTH).

OLID is a collection of 14,100 annotated tweets obtained using the twitter API. The training dataset consists of 13240 annotated tweets while the test partition contained about 860 tweets. The whole problem can be formulated as the 3 sub-tasks of detecting the offensive content, identifying the type of the offensive content and lastly to categorize the target of the targeted offensive content.

The distribution of OLID dataset is reflected with the figure 3.1 and the distribution for the training set is depicted below in the hierarchical order of annotation.

1. **For sub-task A** – The distribution of the each of the labels in the training set for this subtask A is below
    - OFF – 4480 tweets (33.23%)
    - NOT- 8840 tweets (66.76%)

A	B	C	Train	Test	Total
OFF	TIN	IND	2,407	100	2,507
OFF	TIN	OTH	395	35	430
OFF	TIN	GRP	1,074	78	1,152
OFF	UNT	—	524	27	551
NOT	—	—	8,840	620	9,460
All			13,240	860	14,100

FIGURE 3.1: Distribution of 3 levels of annotation in OLID

2. **For sub-task B** – The distribution of the each of the labels in the training set for this subtask B is below

   - TIN- 3876 (88.09%)
   - UNT- 524 (11.90%)

3. **For sub-task C** – The distribution of the each of the labels in the training set for this subtask C is below

   - IND – 2407 (62.10%)
   - GRP – 1074 (27.70%)
   - OTH – 395 (10.19%)

3.3 Research Objectives

1. As our main task, we need to determine whether a tweet is offensive or not. Since the dataset is new and not much exploration is done on that, we aim to explore numerous techniques in various phases of the project pipeline and evaluate the impact of each of them on the performance. Finally we, aim to develop and come up with an efficient system for this task and try to improve the previous results on the dataset.

2. In data exploration phase, we found out that this dataset is very messed up with full noisy data to be fed into a Machine/Deep learning model. Therefore we will perform a bunch of pre-processing steps to clean this dataset. Much details would be provided in the next chapter.

3. The another important objective of the project is to balance the training set of our OLID data by experimenting with various Undersampling and Oversampling techniques for re-balancing it.

(a) WordCloud- Non-Offensive Tweets (b) WordCloud- Offensive Tweets day

FIGURE 3.2: WordCloud for Non-Offensive and Offensive Tweets

4. For feature extraction, we will experiment with various available techniques like TF-IDF, BOW, Uni-gram, Bi-gram, Tri-gram and their combinations. We will also compare the performance with pre-trained embeddings like Word2Vec and GloVe.

5. For the purpose of model building, we will apply a whole range of Classical Machine learning models like SVM, Naive Bayes, Logistic Regression, Decision Tree along with the ensembles like Random Forest, Gradient Boosting and Ada Boost to classify our tweets.

6. Further to investigate in much detail, we will build the Deep learning models like CNN, GRU, LSTM along with state of the art techniques such as BERT by Google, fine-tuned on our dataset.

7. To evaluate the performance of our Machine learning and Deep Learning models, we will be using per class Precision, Recall and F1-score. The overall performance of the model is evaluated using the Macro F1 score as the test set is imbalanced as evident in the exploration phase.

8. Finally, we will present the top Feature extraction mechanisms along with the Machine/Deep learning models that worked best for our problem of detecting the offensive posts. In the next chapter we would be reporting the results of our experiments in much details through comparative analysis.

Chapter 4

Implementation Approach

In this chapter, we will start of with the experiment design and methodology where we will discuss the techniques employed in data pre-processing, various feature extraction mechanisms, techniques for handling the imbalance and various modeling techniques for both Machine learning and Deep learning. Further we will proceed with the implementation details and the process flow of our research project in terms of softwares, platforms and programming language used to experiment with the techniques in each phase of the pipeline of our process flow. Then we present the results of various feature extraction techniques with variety of machine learning models followed by the results of deep learning models on our dataset. This chapter ends with the evaluation and comparative analysis of these machine learning and deep learning models to come up with the best model for our problem.

4.1 Experiment Design and Methodology

4.1.1 Data Pre-processing

Text Pre-processing is the technique of cleaning the text data so as to discard the irrelevant information from it which makes it more meaningful and reduces the size of our dataset. This can reduce the training time of our model as well.

Our dataset OLID has two features, first one is tweet and the second one is the corresponding label Offensive(OFF) or Not-Offensive(NOT). Therefore there is a need to pre-process the tweet column for both our training and the test set. If the cleaned text is fed into our algorithms, then it will produce more reliable results when compared with the uncleaned or garbage data where the model will not be able to interpret much information which is semantically or syntactically insignificant.

Below are the sequence of steps in text pre-processing that can be employed with respect to our dataset to get a better version of the twitter text.

- **Tokenization:** Tokenization is the process of splitting of sentences or paragraph (sequence of strings) into respective tokens which are individual words. This is important as it gives us the lot of insights about the meaning of the text through the extraction of words present in it. It can give us the total word count or individual word frequency.
- **Stemming:** Inflected forms of a word is formed by adding the affixes to the root/base form of the word. Stemming is the process of converting or reducing the inflected or derivationally related forms of a word to its base form by removing the suffix from a word. It is not necessary that the root word produced has some dictionary meaning. For example *beautiful* and *beautifully* will be both stemmed to *beauti* which has no dictionary meaning in english.
- **Lemmatisation:** Lemmatisation also aims to convert the inflected form of a word to its base form by removing the inflectional endings. But this root form generated is the dictionary form of a word called Lemma. It takes into account the meaning of word in a sentence to convert the word into its root form. The process takes the support of a vocabulary and performs the morphological analysis of words. For example, *beautiful* and *beautifully* are lemmatised to *beautiful* and *beautifully* without changing the meaning of words. But *good*, *better* and *best* are lemmatized to *good* since all the words have same meaning
- **Lower Case Conversion:** This is mainly converting all the words in our text to lowercase form. This is done to normalize our textual data.
- **Stop Words Removal:** When we have to create the meaningful features from the text, then these stopwords have no significance. These are generally most frequently occurring words in a text which can be articles such as *a, an, the*, conjunctions such as *for, yet, but, so* or prepositions such as *in, towards, before.* This is done to limit the computation time by reducing the size of our vocabulary.
- **Punctuation Removal:** Punctuations are the symbols which are used in writing to clarify the meaning of the text by separating the sentences and their elements. These are of no use to convey the meaning of text and only used for good readability so we will remove these from our dataset.
- **User and URL Mentions Removal:** We have our text data full of USER and URL mentions which doesn't convey any meaning to our text, so we have removed their presence from our training and the test data.

- **Special Characters Removal:** Special characters are often the non-alphanumeric characters which does not make any sense for our task, so we choose to remove them.
- **Numbers Handling:** There are two ways to handle the numbers in the text data. We can either remove them or convert them to their textual form. In our case, as numbers don't make any contribution in predicting whether the tweet is offensive or not, we choose to remove them.
- **Emojis Handling:** Emojis are visual representations that convey an idea or an emotion while including them in a text. There are two ways to handle them. First is to substitute an emoji with a textual word and second one is to simply remove them. For our task we initially proceeded with removing them as they don't contribute much in our text data.
- **Contraction Expansion:** Contractions are the short form of words which are created using removing one of the vowels from them. For example contraction of *do not* is *don't*. We need to convert these contractions to the expansion forms which helps in text standardization.
- **White Space Removal:** After we perform all the pre-processing steps there is the generation of the white spaces that are of no use as will take unnecessary of memory space so better to remove them.

4.1.2 Feature Extraction Techniques

Feature Extraction with respect to NLP tasks is to extract the features and produce the numerical representations of the text data so that it can be supported by the Machine Learning and Deep Learning algorithms for predictive calculations.

4.1.2.1 Feature Extraction with Statistical Models

1. **Bag Of Words:** Raw text data has the variable length, but most of the machine learning algorithms expect the numerical feature vectors as the input. Vectorization is the process of converting the text documents into the numerical feature vectors. *Bag of Words* [22] does this with the help of Tokenisation, Counting and Normalization.

   This model takes in the whole corpus of text documents and produces a sparse matrix where each row in the matrix represents the word frequency of one document and each column or feature is the corresponding token or unique word in the

whole corpus. This model ignore the relative position information of the words in the document and based only on the word occurrences in each document. The basic idea is shown by the below figure 4.1

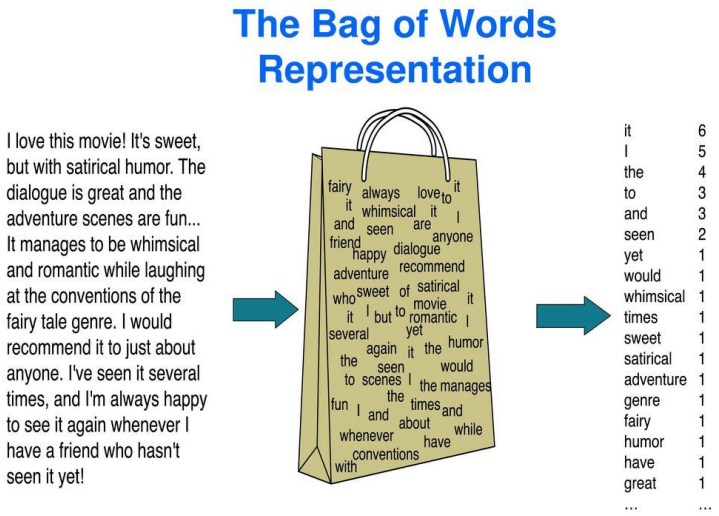


FIGURE 4.1: Feature Extraction through BOW

2. **Bag of Words Unigram, Bigram, Trigram and their Combinations:** A n-gram is the sequence of n words which aims to capture more meaning from the text document by preserving the sequence information (order in which the word appears in the document) when compared to the simple unigram BOW. This representation changes both the scope and the vocabulary of our model. This way, the each feature of our obtained matrix through n-grams model is the combination of n consecutive words that occur in the whole corpus.

   Each row represents the combination frequency of each feature of that particular document in the text corpus. *Bi-Gram* model forms the features as all the possible combinations of two consecutive words occurring in the corpus text whereas the *Tri-Gram* captures the possible combinations of three consecutive words occurring in the corpus for feature extraction. Therefore, we can either individually consider the Unigram, Bigram, Trigram or Ngram in BOW model or can make different combinations of them for extracting the important features from the text.

3. **Tf–Idf term weighting:** When we create the vector from a document based on word frequency, then the most frequent words/terms which carry very little information about the content, starts to dominate over the rarer words which are more significant and interesting. So instead of scoring the terms in the documents by the simple frequency in one document, we rescale it to the frequency of the words appearing in all the documents. This will incur the penalty for the words that are frequent in a document but also frequent across all the documents.

Tf in Tf-Idf denotes the score based on frequency of the word in the current document and Idf gives the score based on how rare that word is across all the documents. The multiplication of both these scores gives us the final numerical weight of the word/term when converting a document to a vector representation. The basic idea of *Tf-Idf* is that, the most frequent words are less contextually important than the domain specific or rare words which are more significant in the corpus. Therefore this approach of feature extraction tends to associate each word of the documents present in the corpus with a less weight if it is frequent across all the documents and a high weight if it is rare across all the documents.

4. **Tf–idf term weighting Unigram, Bigram, Trigram and their Combination:** We have seen the concepts of Bi-grams, Tri-grams and N-grams in the above paragraphs. There we saw that, this idea can be integrated with BOW method for extracting the useful features. Here also we can use these combinations with Tf-idf weighting scheme for feature extraction.

4.1.2.2 Feature Extraction with Word Embeddings

1. **Word2Vec**: This technique was introduced by the researchers of Google in 2013. The paper [23] aims to address the shortcomings of the previous approaches for vectorizing the text documents like BOW or Tf-Idf. These traditional approaches results in the large sparse matrices giving only the information about the documents but not display the meaning of the words. This Word2Vec technique takes the semantic meaning into consideration. This word embedding technique receives the input as the large corpus of text and produces the vector space field of several hundreds of dimensions with the help of shallow neural network; where each unique word in the corpus is assigned the vector in that space. In short Word2Vec is a methodology to convert the words to their numerical vector form preserving the meaning and the context of the words in the vector space. Closer words in the vector space signify that they share the same context and are similar.

2. **GloVe** : The abbreviation GloVe denotes the Global Vectors was first introduced in the work [24] at Stanford University. This technique also aims to find the d-dimensional vector representations for each word in vector space. The distance between these words is dependent on the semantic similarity. GloVe is based on matrix factorization techniques on the word-context matrix and is one of the most popular embeddings used in NLP.

3. **BERT**: BERT stands for Bidirectional Encoder Representations from Transformers [25] is the significant **state of the art** technique in Natural Language Understanding given by the researchers of Google AI Language. The main aim of

this BERT model is to produce the pre-trained vector representations of the words and can be used for range of NLP tasks by just little fine tuning and addition of just one output layer. Bi-directional in the model signifies that it has learnt the representation from both left and right context for all the processing layers which makes it even more efficient at understanding the context of words in a sentence. Encoder representation means that it refers to the algorithm which has trained this model to represent the words from huge corpus of text which is from Wikipedia and Google's book corpus estimating around 3300 million words. Architecture of BERT is based on Transformer layers containing multiple attention heads.

4.1.3 Handling Imbalance

Upon the Exploratory data analysis, we found that our training set was imbalanced as the Offensive tweets were 33% and Non-offensive ones were 66%. We have experimented with various oversampling and undersampling techniques for handling this issue. It is very essential to handle this large imbalance because otherwise the results would be biased for the majority class only. Below figure 4.2 shows the class imbalance in the training set of OLID. The main **goal** of these resampling techniques is to balance the distribution of majority and minority classes in the dataset.

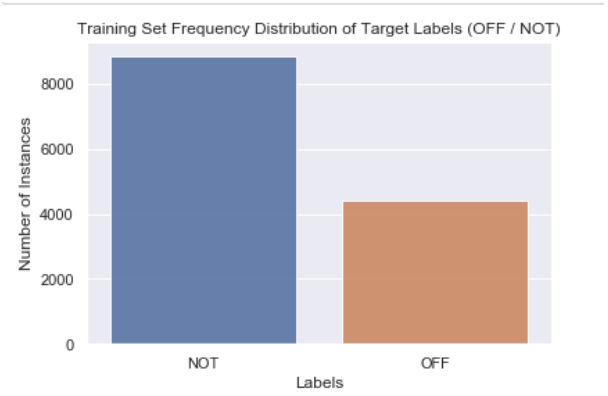


FIGURE 4.2: Training set distribution of class labels

4.1.3.1 Oversampling Techniques

The main goal of Oversampling techniques is to rebalance the dataset by increasing the instances of minority class to match the distribution of majority class in the dataset. This overall increases the size of the dataset.

The main drawbacks of Oversampling is that our model might overfit on the training data.

1. **Random Oversampling:** This technique will randomly increase the size of minority class by randomly duplicating instances from the minority class until the distribution of the b classes are equal and balanced. This is similar to observation weighting.

2. **SMOTE:** This technique [26] allows us to create the new artificial data points in our dataset. Below are the steps that are need to be taken in order to create these artificial points.

    2.a) Select a data point from the minority class.

    2.b) Select the k nearest neighbours of the selected data point in the feature space.

    2.c) Select any one random neighbours from k of them.

    2.d) Calculate the difference vector between the original data point and this randomly selected neighbour.

    2.e) Multiply this vector by the random number between 0 and 1 to get the intermediate random value.

    2.f) This random value obtained is the artificial data point in the feature space that will lie between the original data point and it's neighbour.

4.1.3.2 Undersampling Techniques

The main goal of Undersampling techniques is to rebalance the dataset by decreasing the instances of majority class to match the distribution of minority class in the dataset. This overall decreases the size of the dataset.

The main drawback of Undersampling is that we may loose the important information from our dataset.

1. **Random Undersampling:** This technique will randomly remove data from the majority class until the distribution of the two classes are equal and balanced.

2. **Tomek Links:** Tomek links is the method of under-sampling and removes the instances of majority class in areas of significant overlap between the classes.

    - Let x be an instance of class A and y an instance of class B.
    - Let d(x, y) be the distance between x and y.
    - (x, y) is a Tomek link, if for any instance z, $d(x, y) < d(x, z) < d(y, z)$.

- This means that y is the nearest neighbour of x or that x is the nearest neighbour of y.
- Tomek links can be used to reduce the level of overlap between the majority and minority classes.

4.1.4 Algorithmic Modeling

Algorithmic Modeling is the process of building a range of both Machine learning and Deep Learning models to carry out the predictions on the unseen or the test data.

4.1.4.1 Machine Learning Based Models

Below are the Machine Learning algorithms that we have used to determine whether a given tweet is offensive or not.

1. **Support Vector Machine:** The Support Vector Machine was first introduced in the work [27]. They are the supervised machine learning algorithms that are used for both classification and regression tasks which can be linear or non-linear. The main of this algorithm is to come up with the hyper-plane such that the marginal width is maximum. Marginal width is the distance between the 2 marginal lines where each marginal line is at closest data point of each of the classes present in our dataset. These closest points belonging to each class from the hyper plane are called support vectors [28]. SVM works very well in the cases of high dimensional data and also in the cases where number of dimensions are greater than the number of instances or data points.

2. **Naive Bayes:** This type of classifiers are the class of probabilistic classifiers **??**. The algorithm is based on the mathematical formulation of Bayes Theorem. It is called naive because the assumption of this algorithm is that, the features are independent of other features in the dataset; which is never fully true. The formulation of Naive Bayes Classifiers is depicted below by the equation 4.1 where 'p(x)' is the probability of feature given values for 'y' features.

$$p(x|y) = \prod_{i=1}^{p} p\left(x_i|y\right) \tag{4.1}$$

3. **Logistic Regression:** This supervised machine learning algorithm as presented in the book [29] is generally used for classification problems rather than regression. In linear regression the output can be anything positive or negative values, but

here in logistic regression, we squeeze our output between 0 and 1. The aim here also is to find the decision surface which best separates the 2 classes in our dataset. The data points below this decision line will result in positive values and the ones above the line results in negative values. These values are passed into the sigmoid function such that the positive values are greater than 0.5 and the negative values are less than 0.5. Therefore we can easily classify our 2 classes. The idea can be extended to multi-class classification problems also, but generally works best for binary classification problems.

4. **Decision Tree:** A decision tree [30] is a graphical representation of a decision process. This type of machine learning model takes input as the vector of feature values and produces the prediction class/value. It is a flow like structure where the nodes of the tree represents the features in our dataset and the leaves denotes the predicted class. Nodes are the test on the attribute, each branch is the outcome of the test and leaf or terminal node holds a class label where the topmost node is the root node. These algorithms are simple to interpret and, understand and visualize. These also explicitly performs the feature selection and can handle both numerical and categorical data.

5. **Ensemble Techniques:** Ensemble methods in Machine learning utilize or combine the results of various learning algorithms to obtain a better predictive performance that could be obtained from the same base learner or multiple base learners. All of the ensembles have two things in common. First is that, they construct multiple, diverse predictive models from adapted versions of the training data. Second is that they combine the predictions of these models in some way, often by simple averaging or voting. Ensembles can be broadly classified into two types which are Bagging (Random Forest [31]) and Boosting (Gradient Boosting [32]/ Ada Boosting)

4.1.4.2 Deep Learning Based Models

Some of the Deep Learning Architectures that we have experimented are stated below:-

1. **Artificial Neural Networks**
2. **Convolutional Neural Netoworks**
3. **Long Short Term Memory Networks**
4. **Bidirectional Long Short Term Memory Networks**
5. **Hybrid Architectures of CNN and LSTM**

4.1.5 Evaluation Metrics

Evaluation Metrics are the ways to measure the performance and the quality of our Machine learning or Deep learning models. These metrics should be choosen very carefully depending upon the test dataset we have at our disposal. For example, if we choose accuracy on the imbalanced test set then it might give a good overall accuracy, but actually the majority class accuracy is dominating over minority class accuracy. So here a better metrics could be F1-score or confusion matrix to get the actual insights of the performance of all the classes present in our test dataset.

To evaluate the performance of our Machine learning and Deep Learning models for detecting the offensive language, we will be using per class Precision, Recall and F1-score. The overall performance of the model is evaluated using the Macro F1 score as the test set is imbalanced as evident in the below figure 4.3

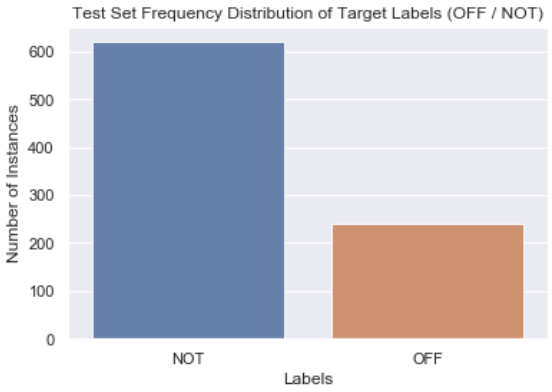


FIGURE 4.3: Test set distribution of class labels

Some concepts that will be important to understand the further concepts are as follows and can be depicted by the figure 4.4

- **True Positives (TP)-**Number of instances of majority class that were correctly classified as majority class.
- **True Negatives (TN)-**Number of instances of minority class that were correctly classified as minority class.
- **False Positives (FP)-**Number of instances of minority class that were incorrectly classified as majority class.
- **False Negatives (FN)-**Number of instances of majority class that were incorrectly classified as minority class.

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

FIGURE 4.4: Representation of Confusion Matrix

Below are some evaluation metrics that we will be using to determine the prediction performance of our models on this OLID test set.

1. **Precision:** It is the measure of the positive predictive value. It tells how confident we can be that any instance predicted as belonging to a certain class actually belongs to that class.The formula for the precision for any class is as follows

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{4.2}$$

2. **Recall:**It is the measure of sensitivity or the true positive rate. It tells how confident we can be that all instances belonging to a specific class have been correctly classified by the model. The formula for the recall for any of class present in the target feature is as

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{4.3}$$

3. **F1-Score:** It is interpreted as the weighted average of precision and recall for a particular class. For F1 score to be high of any class, both the precision and recall values should be high as well. The best value for f1 score is 1 and worst value is 0. The formula for F1-score is

$$F1\ Square = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{4.4}$$

4. **Confusion Matrix:** It is the representation of the information about the actual and predicted classifications by the classification algorithm. It gives the insight into the accuracy of each of the classes that are present in the target feature meaning that we get the number of test instances for each class that are correctly as well incorrectly predicted by our model.

5. **Classification report:** This report provides an insight into the precision, recall and f1 score broken down by the classes present in our dataset. For our problem as the data is imbalanced, so we will not see the normal accuracy, rather our evaluation for this biased classification would be on the basis of confusion matrix (indication of the accuracy of each of the class). Also we would be considering

the f1 score across each of the 2 classes which tell about how good the balance is between the precision and recall values. Moreover this classification report also generates the mean and weighted average of precision, recall and f1 score for each of the individual classes in our target feature. So we would be considering the macro average and weighted average of f1 scores for both the classes as the one of the metric for evaluation.

4.2 Implementation Details and Results

4.2.1 Process Flow and Implementation Details

Our whole process of carrying out this research project is divided into 5 major steps and implemented in Python Programming Language :-

1. **Text Pre-processing**: For carrying out this step of processing our text into some cleaned version, we took the help of Natural Language Toolkit [33] (NLTK). This platform provides the various libraries and built in modules to deal with the textual data.

2. **Feature Extraction**: We experimented with various feature extraction techniques like BOW and TF-IDF present in Scikit-learn API [22]. Further we explored the pre-trained word embeddings like Word2Vec and GloVe with the help of Gensim library [34].

3. **Handling Imbalance**: For handling the major issue of imbalance in our training data, we leveraged the imbalanced-learn API [35] whose packages and modules we have utilized for this project. This toolkit implemented the overall solution of imbalance with four techniques. These were under-sampling, over-sampling, combination of both over sampling and under sampling and ensemble learning methods. We only experimented with over-sampling and under-sampling approaches.

4. **Model Building**: For building and using various machine learning models for detecting the offensive tweets, we used the Scikit-learn API [22]. And for the deep learning algorithms, we used the TensorFlow API [36]

5. **Model Evaluation**: For evaluating our model performance, we came up with various evaluation metrics available in Scikit-learn API [22] such as precision, recall and F1-score. Finally we reported the model performance with confusion matrix insights and macro F1 score.

4.2.2 Various Machine Learning Models with Different Feature Extraction Techniques

We found that Oversampling techniques such as SMOTE and Random Oversampling did not performed well when compared with Random Undersampling. The results of these oversampling methods on the test set were biased towards the majority class that is Non-Offensive tweets. Therefore, we choose Random Undersampling technique to compare various Feature Extraction Mechanisms.

In this sub-section we will present the results of trying the different Feature Extraction Mechanisms like BOW, Tf-idf along with their Bi-gram and Trigram Combinations with Random Undersampling. We will also see the results with pre-trained embeddings like Word2Vec and GloVe.

4.2.2.1 Feature Extraction with BOW

Results of BOW (Uni-gram) with Random Undersampling

Model	**Macro F1-Score**	**Majority Class Accuracy**	**Minority Class Accuracy**
SVM	0.70	0.77	0.67
Naive Bayes	0.67	0.68	0.73
Logistic Regression	0.73	0.83	0.65
Decision Tree	0.66	0.72	0.66
Random Forest	0.73	0.81	0.66
Gradient Boosting	0.74	0.94	0.49
Ada Boost	0.73	0.92	0.49

TABLE 4.1: Evaluation Metric for Feature Extraction with BOW (Uni-gram)

Results of BOW (Uni-gram + Bi-gram) with Random Undersampling

Model	**Macro F1-Score**	**Majority Class Accuracy**	**Minority Class Accuracy**
SVM	0.72	0.90	0.50
Naive Bayes	0.64	0.62	0.77
Logistic Regression	0.73	0.84	0.62
Decision Tree	0.66	0.73	0.64
Random Forest	0.71	0.87	0.54
Gradient Boosting	0.73	0.94	0.48
Ada Boost	0.73	0.92	0.49

TABLE 4.2: Evaluation Metric for Feature Extraction with BOW (Uni-gram + Bi-gram)

Results of BOW (Uni-gram + Bi-gram + Tri-gram) with Random Undersampling

Model	Macro F1-Score	Majority Class Accuracy	Minority Class Accuracy
SVM	0.69	0.94	0.40
Naive Bayes	0.64	0.62	0.78
Logistic Regression	0.73	0.84	0.62
Decision Tree	0.68	0.73	0.67
Random Forest	0.73	0.91	0.52
Gradient Boosting	0.73	0.93	0.48
Ada Boost	0.73	0.92	0.49

TABLE 4.3: Evaluation Metric for Feature Extraction with BOW (Uni-gram + Bi-gram + Tri-gram)

4.2.2.2 Feature Extraction with Tf-idf

Results of Tf-idf (Uni-gram) with Random Undersampling

Model	Macro F1-Score	Majority Class Accuracy	Minority Class Accuracy
SVM	0.74	0.85	0.62
Naive Bayes	0.68	0.70	0.72
Logistic Regression	0.72	0.83	0.62
Decision Tree	0.63	0.67	0.66
Random Forest	0.71	0.83	0.60
Gradient Boosting	0.74	0.92	0.51
Ada Boost	0.71	0.93	0.46

TABLE 4.4: Evaluation Metric for Feature Extraction with Tf-idf (Uni-gram)

Results of Tf-idf (Uni-gram + Bi-gram) with Random Undersampling

Model	Macro F1-Score	Majority Class Accuracy	Minority Class Accuracy
SVM	0.71	0.80	0.64
Naive Bayes	0.67	0.67	0.74
Logistic Regression	0.70	0.78	0.65
Decision Tree	0.65	0.69	0.65
Random Forest	0.70	0.85	0.53
Gradient Boosting	0.72	0.93	0.48
Ada Boost	0.71	0.92	0.46

TABLE 4.5: Evaluation Metric for Feature Extraction with Tf-idf (Uni-gram + Bi-gram)

Results of Tf-idf (Uni-gram + Bi-gram + Tri-gram) with Random Undersampling

Model	Macro F1-Score	Majority Class Accuracy	Minority Class Accuracy
SVM	0.71	0.80	0.63
Naive Bayes	0.67	0.69	0.74
Logistic Regression	0.70	0.77	0.65
Decision Tree	0.66	0.70	0.67
Random Forest	0.69	0.84	0.54
Gradient Boosting	0.73	0.93	0.48
Ada Boost	0.71	0.92	0.47

TABLE 4.6: Evaluation Metric for Feature Extraction with Tf-idf (Uni-gram + Bi-gram + Tri-gram)

4.2.2.3 Feature Extraction with Word2Vec

Model	Macro F1-Score	Majority Class Accuracy	Minority Class Accuracy
SVM	0.73	0.78	0.68
Naive Bayes	0.69	0.80	0.68
Logistic Regression	0.74	0.72	0.74
Decision Tree	0.64	0.70	0.72
Random Forest	0.70	0.85	0.70
Gradient Boosting	0.69	0.90	0.55
Ada Boost	0.73	0.91	0.50

TABLE 4.7: Evaluation Metric for Feature Extraction with Word2Vec

4.2.2.4 Feature Extraction with GloVe

Model	Macro F1-Score	Majority Class Accuracy	Minority Class Accuracy
SVM	0.75	0.80	0.71
Naive Bayes	0.70	0.82	0.70
Logistic Regression	0.73	0.73	0.75
Decision Tree	0.67	0.71	0.69
Random Forest	0.72	0.87	0.73
Gradient Boosting	0.71	0.92	0.60
Ada Boost	0.74	0.93	0.55

TABLE 4.8: Evaluation Metric for Feature Extraction with GloVe

4.2.3 Results of Various Deep Learning Models

Model	**Macro F1-Score**	**Majority Class Accuracy**	**Minority Class Accuracy**
Artificial Neural Network (BOW)	0.73	0.94	0.52
CNN+LSTM (BOW)	0.75	0.93	0.60
CNN+BiLSTM (GloVe)	0.61	0.73	0.53
CNN+BiLSTM (BOW)	0.77	0.89	0.64
BERT	0.82	0.91	0.70

TABLE 4.9: Evaluation Metric for Deep Learning Models

4.3 Evaluation and Analysis

In this section we will be doing the comparative analysis of various feature extraction methods to come up with the top Machine learning models for each of these methods. We will be selecting the best models based on the Macro F1 score and the insights from the confusion matrix (majority and minority class accuracy). We will also present our top performing Deep learning models in this section. All this analysis will help us to come up with the best model for predicting the offensive tweets with our dataset OLID.

4.3.1 Comparative Analysis of Various Feature Extraction techniques along with Machine Learning and Deep Learning Models

Feature Extraction/Machine Learning Model	**Macro F1-Score**	**Majority Class Accuracy**	**Minority Class Accuracy**
BOW(Unigram) + SVM	0.70	0.77	0.67
BOW(Unigram+Bigram+Trigram) + Decision Tree	0.68	0.73	0.67
TF-IDF(Unigram+Bigram+Trigram) + Logistic Regression	0.70	0.77	0.65
TF-IDF(Unigram) + Naive Bayes	0.68	0.70	0.72
Word2Vec + Logistic Regression	0.74	0.72	0.74
GloVe + SVM	0.75	0.80	0.71
BOW + CNN-BiLSTM	0.77	0.89	0.64
BERT	0.82	0.91	0.70

TABLE 4.10: Comparative Analysis of Various Feature Extraction, Machine Learning and Deep Learning Techniques

4.3.2 Best Model for Offensive Language Detection

From the above comparative analysis of various Machine Learning and Deep Learning models, we found that the **BERT** achieved the highest F1 score of **0.82** beating the performance of the previous work carried in [1].

Chapter 5

Conclusions and Future Work

This chapter starts with the main conclusions that we have come up with, in terms of background, problem description and the solution approach for this research project. Then we end this chapter with the possible areas of future work that can be undertaken in order to further contribute meaningfully to the topic of problem area.

5.1 Conclusion

The following are the main conclusion points that are being achieved in effort to contribute to the problem of detecting the offensive language on the social media platforms through this project:-

1. The extensive literature review from the previous work in the area of Offensive Language Detection helped us to explore the novel benchmark dataset - OLID in the great depth as it was recently created.

2. We explored the various techniques of handling the imbalance in the training set, but for our dataset; the random under-sampling worked best minimizing the difference between the majority and the minority class accuracy and giving the higher macro F1 scores.

3. We performed an in depth comparative analysis of various feature extraction mechanisms, machine learning and deep learning models to explore the dataset even further and set the significant base point from which the future research could proceed with the dataset and the results achieved with this thesis work.

4. Finally the evaluation and the analysis of our comparisons lead to the fact that the state of the art model by Google - BERT outperformed the previous performance achieving the F1 macro score of 0.82 on this dataset.

5.2 Future Work

Below are some future research directions that we have planned if we would have some more time to contribute on the problem area.

1. We have only focused on the sub-task A that is the detection of offensive language for this research project. But on the same dataset there are two more sub-tasks A and B which focused on determining the type and the target of the offensive post. This can even lead us to further insights and analysis of offensive posts detected on the social media to take the relevant actions. In future, we would like to solve these both sub-tasks using variety of approaches to contribute more in the domain.

2. Human language is very much diverse and certain posts may not look offensive from the surface, but actually they are when analysed by the human annotator. In future we will like to conduct an analysis to capture the syntactic and semantic features along with their combination and other pre-trained features. This have actually the potential to improve our performance than what we have achieved now.

3. We have clearly seen that to tackle the issue of imbalance, the random under-sampling has performed the best with our dataset. But at the same time the big problem with this approach is that we have lost a good amount of significant information which might be useful in the training of our model. This approach have reduced our training data to a great amount, therefore we lost important features from the dataset. In future we may come up with the suitable approach and exploration that may tackle this problem of huge imbalance in the dataset efficiently without the reduction in training data.

4. In future we would also like to combine both machine learning and deep learning models to create a significant ensemble model which can help to improvise upon the current results.

5. In our previous chapter, it is quite visible that deep learning models are not performing that good because of less depth in their architectures. In future we aim to increase the complexity of these models by increasing the layers and the configuration settings given the computational resources and the time.

6. We also aim to perform the hyper-parameter tuning of both machine learning and deep learning models to come up with the optimal values of these hyper-parameters for our dataset. This may also contribute in the boosting of the performance by our models.

Bibliography

[1] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 1415–1420.

[2] K. Keppner, "Artificial intelligence in supply chain planning – why a hybrid ai concept is the better choice," 2018.

[3] S. Kassel, "Predicting building code compliance with machine learning models," 2017.

[4] J. K. Gill, "Automatic log analysis using deep learning and ai," 2018.

[5] J. Chugh, "Types of machine learning and top 10 algorithms everyone should know," 2018.

[6] Shriyanka, "What is natural language processing ?" 2017.

[7] A. LLC, "Ai, machine learning (ml) and natural language processing (nlp)," 2019.

[8] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *fifth international AAAI conference on weblogs and social media*, 2011.

[9] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, "Learning from bullying traces in social media," in *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics, 2012, pp. 656–666.

[10] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in *European Conference on Information Retrieval*. Springer, 2013, pp. 693–696.

[11] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

[12] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy & Internet*, vol. 7, no. 2, pp. 223–242, 2015.

[13] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 29–30.

[14] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 145–153.

[15] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Eleventh international aaai conference on web and social media*, 2017.

[16] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," *arXiv preprint arXiv:1712.06427*, 2017.

[17] Z. Waseem, T. Davidson, D. Warmsley, and I. Weber, "Understanding abuse: A typology of abusive language detection subtasks," in *Proceedings of the First Workshop on Abusive Language Online*, 2017, pp. 78–84.

[18] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 1–11.

[19] S. Malmasi and M. Zampieri, "Challenges in discriminating profanity from hate speech," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 30, no. 2, pp. 187–202, 2018.

[20] M. Wiegand, M. Siegel, and J. Ruppenhofer, "Overview of the germeval 2018 shared task on the identification of offensive language," 2018.

[21] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, 2018, pp. 1–6.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine

learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

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

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[27] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[28] N. Cristianini, J. Shawe-Taylor *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[29] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[30] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[31] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest."

[32] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[33] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 2002, pp. 63–70.

[34] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[35] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/