ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Information filtering of social media Amharic texts Based on Sentiment Analysis

Hiwot Wonago Kululo

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment For The Degree of Masters of Science in Software Engineering.

Addis Ababa, Ethiopia

17/7/ 2020

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Information filtering of social media Amharic texts Based on Sentiment Analysis

Hiwot Wonago

Advisor: Ayalew Belay (Ph.D.)

Signed by the Examining Committee:

Name	**Signature**	**Date**
1. Ayalew Belay (PhD),	Advisor _____	
2. ---------- (PhD),	Examiner _____	
3. ---------- (PhD),	Examiner _____	

Abstract

In the last five years, the ever growing usage of social media in Ethiopia has fueled the country's problem against the peaceful coexistence of its people. Illegitimate social media usage has played a significant role in widening the distress between the people. As a result, the government has increasingly relied on the temporary closure of social media sites; nationwide internet shutdowns and filtering websites to suppress polarizing voices and the misuse of social media as the tension among many ethnic groups become more visible. As such, there is a need to develop an intelligent system that automatically detects such inappropriate (offensive) contents by classifying them into socially-offensive, religiously-offensive, politically-offensive and non-offensive categories and filter Toxic online contents. We explain the challenges of the Amharic text that is available on the internet and the role of sentiment analysis in mining Amharic dataset on social media. Using different supervised machine learning techniques, this study analyzed performance variations of the algorithms on Amharic texts. The objective of this paper is to apply the concept of sentiment analysis on Amharic text on social media and presents a comparative study on machine learning algorithms. The created social media content filtering system has been tested on Facebook posts of each class, and it has been observed that SVM with word2vec has performed best in comparison to other classifiers, achieving average precision of (72%), but did worse on recall(63.4%). The experimental evaluation shows how the proposed approach is effective and the results are quite satisfactory.

Keywords – Information filtering, Natural Language Processing, Sentiment classification, word2vec.

Acknowledgment

First of all, I would like to thank God and St. Mary, for having me guided at every stage of my life and for giving me the strength, support, and knowledge in exploring things.

This paper and the research behind it would not have been possible without the exceptional support of my advisor Dr. Ayalew Belay, His enthusiasm; knowledge and exacting attention to detail have been an inspiration and kept my work on track from my first encounter with the research idea to the final submission of this paper.

I am grateful to all of those with whom I have had the pleasure to work during this research. Each member of my class have provided me extensive personal and professional comments and taught me a great deal about both scientific research and life in general. I would especially like to thank Tewodros Abebe(Ph.D.), as a friend and mentor, he has taught me more than I could ever give him credit for here. He has shown me, by his example, what a good scientist (and person) should be. I would like to thank my elder brothers, Wondmagegn Wonago and Dr.Lijalem Abera, whose love and guidance are with me in whatever I pursue. They are the ultimate role model. Most importantly, I want to thank my husband Mesfin Kebede and my friends Amanuel Negash, Bezawit and Engdawerk and Addis Ababa University Computer science department.

Table of Contents

List of Tables

List of Figures

List of Algorithms

ABBREVIATIONS

CBOW	CONTINUOUS BAG OF WORDS
DT	DECISION TREE
FN	FALSE NEGATIVE
FP	FALSE POSITIVE
LG	LOGISTIC REGRESSION
MA	MORPHOLOGICAL ANALYZER
ML	MACHINE LEARNING
NB	NAÏVE BAYES
NLP	NATURAL LANGUAGE PROCESSING
SA	SENTIMENT ANALYSIS
SVM	SUPPORT VECTOR MACHINE
TF	TERM FREQUENCY
TF*IDF	TERM FREQUENCY BY INVERSE DOCUMENT FREQUENCY

Chapter 1 : Introduction

1.1 Background

Social media is computer-based technology that facilitates the sharing of ideas, thoughts, and information through the building of virtual networks and communities. By design, social media is internet-based and gives users quick electronic communication of content. Content includes personal information, documents, videos, and photos. Users engage with social media via computer, tablet or smartphone via web-based software or web application, often utilizing it for messaging. Social media typically features user-generated content and personalized profiles. Content can be information, entertainment or nothing specific at all. Content comes in various forms: video, audio, text or image. It is designed to transfer a certain feeling, information and data to somebody. Furthermore, the overuse of social media is a global problem impacting all generations, and research has shown that substantial Internet usage can have a highly negative impact on our mental and emotional health. Li[1]Symptoms of anxiety, depression and obsessive-compulsive disorder can be triggered by the overuse of social media, as individuals are constantly concerned about their posts and communicating with others. When it comes to the experience of the society as a whole, social media has enormous negative impacts. Some of the disadvantages to society as indicated in the stat-counter report are Cyberbullying, Hacking, Addiction, Fraud, and Scams, Security Issues, Reputation, and others.

The content on social media that is composed, transmitted, accessed, or received may contain contents that could be considered discriminatory, offensive, obscene, threatening, harassing, intimidating, or disruptive to any person. Examples of unacceptable content may include, but are not limited to, sexual comments or images, racial slurs, gender-specific comments, or any other comments or images that could reasonably offend someone on the basis of race, age, gender, religious or political beliefs, national origin, disability, sexual orientation, or any other characteristic protected by law. Because of this and many other problems, social media content has to be filtered.

Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such

as products, services, organizations, individuals, issues, events, topics, and their attributes[2].

On social media, the sentiment of a post can be seen in the tone or emotion conveyed in a text. We can understand exactly how people feel about political, social, religious situations in Ethiopia, by effortlessly analyzing the sentiment of each and every post and monitoring how the community is responding to contents. Social media sentiment analysis involves applying natural language processing (NLP) to social mentions from various sources and determining whether the user is talking about political situation, religious organization or some products in a positive, negative or neutral way. A proper social media sentiment analysis could categorize social media mentions into the right category. Social media sentiment analysis uses some revolutionary machine learning and deep learning algorithms and analyses the text posted online. Sentiment analysis is a challenging task especially for languages having low resources and complex linguistic structures like Amharic. A small amount of resources like training data for Sentiment analysis highly impacts the accuracy of the system. The complex structure of languages also needs the design of important features and the best combination from these features. Amharic as one of low resourced and morphologically rich languages shares the above challenges. The problem is information filtering and sentiment analysis problem, and our goal is to investigate which supervised machine learning methods are best suited to solve it.

1.2 Motivation

The inspiration to explore this topic came from both the increase in the instances of online offensive contents and level of advancement of Amharic language resource. Amharic is under-resourced language (few or no language processing tools available). It is still language for which very few computational linguistic resources have been developed and very little has been done.

Offensive/inappropriate messages are prevalent and challenging in Ethiopian digital society as individuals spread hate messages hiding behind their screens; multiple hate speeches, videos and images are continuously published and circulating among Ethiopians on social

media. This possibly leads to acts of violence or hates crimes and destroys the lives of individuals, families, communities, and the country at large.

The Ethiopian government increasingly relies on temporary closure of social media sites and nationwide Internet shutdowns, blocking, and filtering of certain websites to quell dissent channeled via the web.

It is, therefore, of critical importance to monitor and identify instances of biased, racist and misleading information on the internet which have the potential to destruct tolerance, trust, and coexistence values of the societies in our country. As soon as possible their spread has to be prevented on social media.

The reasons mentioned above greatly motivated the work presented in this paper.

1.3 Statement of the problem

Based on sentiment analysis of texts on social media, information filtering solution can protect against Toxic online content or can be used to tackle politically and socially sensitive contents, and prevent illegal or unsuitable social media content from being accessed. Failure to filter social media proved costly in many ways. Social media reflecting our dynamic world, society being dependent on day to day live, relevance of the information is a major concern. Information overload has to be reduced, redundant and unwanted information has to be removed from a stream of information. There is lack of researches on information filtering of Amharic texts on social media.

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. It is useful for quickly gaining insights using large volumes of text data. Only few researches are done on Amharic text sentiment analysis. Furthermore, researches done on Sentiment analysis (SA) up to now are specific to high resourced languages like English[3][4], French[5] and other European languages[6][7]. Despite its large number of speakers, only a few research has been done on Amharic language. Natural language processing uses machine learning and data mining to provide a more complete picture, but the inherent complexity of Amharic language makes it difficult to ensure SA accurately analyze tone and context. Amharic SA has many

challenges due to structure of the language. Factors that limit sentiment analysis of Amharic texts on social media includes

- problem-related to correct interpretation of context in which certain words are used, As affective phenomena, emotions, feelings, sentiments, opinions, are difficult to identify and label, even for humans. The computational task of automatically performing this task is thus faced with many challenges such as grammatical nuances, implied meaning from facial expressions and body language, misspellings, ambiguity, and regional or cultural variations in language.
- Finally, dealing with human language and affect is not only difficult because of the inherent ambiguity of words, but also because of the use of words in figurative senses and the use of irony or sarcasm, i.e. conveying a negative evaluation through the use of a mostly positive choice of words.

The performance of Sentiment Analysis systems is highly influenced by this phenomenon, thus irony detection has become an important challenge related to the field.

Researches in Amharic SA[8][9][10] tried to deal with these challenges in different ways. Their approaches can be summarized as SA using rule-based, Machine learning-based, and hybrid of the two. These researches, however, have limitations. The limitations are on the number of datasets[9][8], very small amount of data for training and testing, supervised machine learning models are being successfully used to respond to a whole range of challenges. However, these models are data-hungry, and their performance relies heavily on the size of training data available. In many cases, it is difficult to create training datasets that are large enough and the goal of a machine learning model is to generalize patterns in training data so that we can correctly predict new data that has never been presented to the model, not applying morphological analyzer to get root forms, not applying state of the art numerical representation for text vectorization and unavailability of a mechanism for filtering of social media content based on the result of sentiment classifier.

Social media filtering is now less of an option for digital society and more of a requirement.

Complex nature of Amharic language and texts on social media platforms has introduced challenges in designing information filtering system that is suitable for Amharic text.

Sentiment analysis and Machine learning techniques can be leveraged to address these challenges and build novel methodologies to address the challenges to build filtering system for social media Amharic content.

A new filtering model based on established techniques in Natural language processing, machine learning and artificial intelligence is proposed.

Finally, previous SA approaches are based on the output of other NLP tasks like tokenizer and morphological analyzer. Being dependent on other NLP tasks, SA creates a performance bottleneck in case of low performing morphological analyzer.

Works on sentiment analysis and social media filtering on different languages exist but as Amharic is a low-resource language, in regards to digitization, there is little attempt on this topic. Therefore, in this work the performance of word2vec and supervised machine learning algorithms on Amharic language is analyzed and explored. Qualitative analysis using machine learning algorithms, determining Word analogies using word2vec and filtering social media contents based on prediction of the algorithms, investigating the impact of automatically generated word vector features in Amharic SA task and comparing different machine learning algorithms for Amharic sentiment analysis are focus of this paper.

1.4 Objectives

General

The main goal of this research is to filter offensive contents from social media based on sentiment analysis of Amharic texts on social media.

Specific Objectives

In order to achieve the general objective, the following specific objectives are identified:

- Conduct a literature review
- Collect offensive words online
- Adopt a machine learning algorithm that is appropriate to the morphological property of Amharic words.
- Testing supervised machine learning classifiers with automatically generated features.
- Develop a prototype of the model for evaluation.

- Evaluate the filtering model

1.5 Methods

In order to conduct this research work, the methodologies mentioned below will be used to select and implement appropriate methods and techniques. We will apply Design science research because our research constitutes categories of artifacts including algorithms, human/computer interfaces, design methodologies (including process models) and languages.

Literature Review

Social media and sentiment classification literature like reports, books, journals, proceedings, research papers, etc. are reviewed in detail to get a better understanding of the area and to have detail knowledge on the various techniques of Amharic text sentiment classification. Since this research work is mainly concerned with social media sentiment classification of Amharic texts, it was compulsory to analyze the nature of Amharic texts that contain sentiments.

Data collection

All of the data (comments/posts) used for conducting the experiment are manually collected from Facebook and labeled to their corresponding class by experts from linguists and Law.

Testing and Evaluation

In our experiments, evaluation metrics for information filtering and sentiment analysis (i.e.,precision, recall, and f-score) are used to evaluate the performance

1.6 Scope and limitation

In social media, people generally use unstructured or semi-structured language for communication. In everyday life conversation, people do not care about the spellings and accurate grammatical construction of a sentence which makes sentiment classification a complex task. Because of this sentiment classification requires effective analysis and processing of documents. Since there are no publicly available Natural Language Processing

(NLP) tools and other resources for social media Amharic sentiment classification, which can be integrated with our model, the scope of our research work is:

- Limited to sentiment classification (only Politically-offensive, Non-offensive, Socially-offensive, Religiously-offensive). I.e. it doesn't cover subjective or objective classification. Classifying a sentence as subjective or objective, known as subjectivity classification is not covered
- We use domain-specific posts/comments that are not grammatically checked and organized.
- Limited to purely Amharic texts on social media.
- Offensive content publishers' identification and reasons for offensive and non-offensive classifications are not covered in this research work.
- Attention is given to most common Amharic offensive sentences used to express sentiments "**ጥሩ፣ ደስ ይላል፣ በጣም ጥሩ፣ መልካም** *ወዘተ*" for Non-offensive sentiments and "***መጥፎ ፣ መንጋ፣ ምናምንቴ ወዘ ተ***" for offensive sentiments. Because of their complicated nature, Amharic expressions such as "ቅ ኔ ያ ዊ አ ነ ጋ ገ ር" are out of the scope of this research work.
- The process of detecting fake news is not covered in this research work as it is a very complicated problem.
- Sentiment analysis can be applied at different levels of scope: Document-level, sentence, and aspect (feature) level. Our scope is **Document-level** sentiment analysis to obtain the sentiment of a complete document or paragraph.

1.7 Application of result

In the current socio-cultural and political situations, controlling social media posts by removing illegal contents, as Culture Minister Margot James said for BBC news "prioritize the protection of users, especially children, young people, and vulnerable adults", is beneficial for the society as a whole. Our government has proposed measures to regulate social media publishers over harmful content, including "substantial" fines and the ability to block services that do not stick to the rules.

Hence, the Amharic Sentiment classification model can be used for different purposes. Some of them are:

- Government organizations can use the system to reduce political, social and religious turbulences around office
- The system can be used to classify Amharic texts as offensive or non-offensive.
- The system can be used to answer sentiment questions. For instance, what is the social media users' reaction to the speech by the prime minister?
- Companies can Use sentiment analytics to gain deep insight into what's happening across their customer support.

Personally, we quite like this task because offensive words, trolling and social media bullying have become serious issues these days and a system that is able to detect such texts would surely be of great use in making the internet and social media a better and bully-free place.

1.8 Thesis organization

The remainder of this thesis report is organized as follows. Chapter Two introduces an overview of sentiment classification (sentiment analysis) and the different techniques used in sentiment analysis researches. Moreover, the general steps in sentiment analysis are also discussed in this chapter. Chapter Three presents' reviews of related researches conducted on social media filtering, sentiment classification. In this chapter, an overview and definition of offensive/inappropriate words, in-depth reviews of researches done on social media information filtering using different techniques for different languages is presented. Chapter Four describes the general architecture of the proposed model for the Amharic sentiment classification model. In addition, implementation-related issues such as pre-processing, stop word removal and classification are also explained in the same chapter. Chapter Five presents the experimental results of the proposed model in general and the different algorithms in particular. Finally, future works, recommendations, and conclusions are given in the last chapter.

Chapter 2 : Literature Review

2.1 Introduction

This chapter deals with the works carried out by different researchers on Sentiment classification and information filtering in general, non-English posts on social media in particular in relation to, Identifying offensive contents, information filtering types, the ways of acquiring social media data, supervised machine learning algorithms and the tools and techniques implemented in each phases of filtering system, the evaluation of filtrating systems, development of filtrating applications and the research findings by various researchers on Sentiment analysis and social media content filtering is presented. Sentiment analysis, machine learning algorithms, social media information filtering are the focus areas of our research.

The goal of this literature review is to investigate existing information/content filtering approaches that has been studied up to now, the appearance and occurrences of social media contents, different methods that are used to filter information available online on social networks and reviewing techniques for filtering offensive contents from users wall with the help of sentiment analysis by using different supervised machine learning algorithms.

2.2 Sentiment Analysis

Sentiment Analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [11]. The analyzer tries to read between lines, identifying idiomatic or colloquial expressions, giving interpretation to negations, modifying polarity of words basing on the related adverbs, adjectives, conjunctions or verbs, taking in account-specific functional-logic complements. Sentiment analysis is also a series of methods, techniques, and tools about detecting and extracting subjective information, such as opinions and attitudes, from language. By using sentiment analysis, we are able to distinguish between offensive and non-offensive posts, comments and status updates. This term, sentiment analysis basically aims to classify the given text into the positive, negative and neutral category. Three basic approaches are available in the literature today for sentiment analysis: Lexicon driven, Machine learning-based, and Hybrid (integration of lexicon and machine learning)[2].

Research on sentiment analysis has been investigated from different perspectives. Methods based on ML are much more effective in terms of the accuracy of classification[12]. With ML algorithms it is more difficult to show improvements by incorporating valence shifters, because they are already included, to some degree, in the basic classifier. Even when the classifier uses only unigrams as features, combinations of features detected by the ML algorithm can capture some aspects of the valence shifters. This happens when combinations of terms, including valence shifters, appear regularly in one class of documents (although not necessarily adjacent to each other). Because all terms in a document will affect its classification, valence shifters might have already been considered in classification. Perhaps the most popular perspective is to categorize these studies into three levels, document level, sentence level, and entity and aspect level described as follows[13]:

- Document-level: The aim here is to determine the overall sentiment of an entire document. For example, given a product review, the task is to determine whether it expresses positive or negative opinions about the product. This level looks at the document as a single entity, thus it is not extensible to multiple documents.
- Sentence level: This level of analysis is very close to subjectivity classification and the task at this level is limited to the sentences and their expressed opinions. Specifically, this level determines whether each sentence expresses a positive, negative or neutral opinion.
- Entity and aspect level: Instead of solely analyzing language constructs (e.g. documents, paragraphs, sentences), this level (a.k.a feature level) provides finer-grained analysis for each aspect (or feature) i.e., it directly looks at the opinions for different aspects itself. The aspect-level is more challenging than both document and sentence levels and consists of several sub-problems[13].

In short, sentiment analysis is the automated process that allows machines to identify and extract opinions within text, such as tweets, emails, support tickets, product reviews, survey responses, etc. besides identifying the opinion, sentiment analysis systems extract attributes of the expression e.g.:

- *Polarity*: if the speaker expresses a *positive* or *negative* opinion,
- *Subject*: the thing that is being talked about,

- *Opinion holder*: the person, or entity that expresses the opinion.

Fine-grained Sentiment Analysis

Sometimes people may be also interested in being more precise about the level of the polarity of the opinion, so instead of just talking about positive, neutral, or negative opinions, someone could consider the following categories: Very positive, Positive, Neutral, Negative, and Very negative.

This is usually referred to as fine-grained sentiment analysis. This could be, for example, mapped onto a 5-star rating in a review, e.g.: Very Positive = 5 stars and Very Negative = 1 star.

Multilingual sentiment analysis

Multilingual sentiment analysis can be a difficult task. Usually, a lot of preprocessing is needed and that preprocessing makes use of a number of resources. Since Amharic is under-resourced language, there are no available resources online (e.g. Amharic text preprocessor, list of Amharic Stop words), and many others have to be created (e.g. translated corpora or noise detection algorithms). The creating resource requires a lot of coding experience and can take longer to implement.

Sentiment Analysis Algorithms

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as:

Rule-based systems that perform sentiment analysis based on a set of manually crafted rules.

Automatic systems that rely on machine learning techniques to learn from data.

Hybrid systems that combine both rule-based and automatic approaches.

Sentiment Analysis Metrics and Evaluation

There are many ways in which we can obtain performance metrics for evaluating a classifier and to understand how accurate a sentiment analysis model is. One of the most frequently used is known as **cross-validation**.

What cross-validation does is splitting the training data into a certain number of training folds (with 75% of the training data) and the same number of testing folds (with 25% of the training data), use the training folds to train the classifier, and test it against the testing folds to obtain performance metrics. The process is repeated multiple times and an average for each of the metrics is calculated.

If the testing set is always the same, it might be overfitting to that testing set, which means it might be adjusting the analysis to a given set of data so much that it might fail to analyze a different set. Cross-validation helps prevent that. The more data we have, the more folds we will be able to use.

Precision, Recall, and Accuracy

Precision, recall, and accuracy are standard metrics used to evaluate the performance of a classifier.

- Precision measures how many texts were predicted correctly as belonging to a given category out of all of the texts that were predicted (correctly and incorrectly) as belonging to the category.
- Recall measures how many texts were predicted correctly as belonging to a given category out of all the texts that should have been predicted as belonging to the category. We also know that the more data we feed our classifiers with, the better the recall will be.
- Accuracy measures how many texts were predicted correctly (both as belonging to a category and not belonging to the category) out of all of the texts in the corpus.

Most frequently, precision and recall are used to measure performance since accuracy alone does not say much about how good or bad a classifier is.

For a difficult task like analyzing sentiment, precision and recall levels are likely to be low at first. As we feed the classifier with more data, performance is improved.

Sentiment analysis of Amharic text is a complex task. Amharic language is from the Semitics branch which is the official working language of Ethiopia and some of the federal states and serves as the major lingua franca in Ethiopia, with 14.8 million people speaking it

as a mother tongue and 4 million as a second language (2014 census). However, it is one of the least researched and under-resourced languages with relatively few computational linguistic works. Users have been using the Latin alphabet to transliterate Amharic texts and express their views on social media. Recently, the introduction of Amharic Unicode font and its integration in different Technologies has paved a way for many online publishers and users to interact using the native Ethiopic script. This trend of Amharic being used as a medium of online communication among speakers is contributing to the enrichment of the web with Amharic contents thereby opening further study opportunities for the language.

2.3 Amharic Language

Amharic is Ethiopian national language which has about thirty-two alphabets. More than ninety million people speak this language as native and second language. However, it is one of the least researched and under-resourced languages with relatively few computational linguistic works. The data sources for this research are social Media that contain Amharic scripts which are considered as they are written using incorrect language structure. Social media users have been using the Latin characters to transliterate Amharic texts and express their views on social media. Recently, the introduction of Amharic Unicode font and its integration in different technologies has paved a way for many online publishers and users to interact using the native Ethiopic script. This trend of Amharic being used as a medium of online communication among speakers is contributing to the enrichment of the web with Amharic contents thereby opening further study opportunities for the language. This is because it lacks basic tools and resources for carrying out natural language processing research and applications [2] [3] [4]. Certain Amharic words are labeled in the society as vulgar, offensive, or disparaging. Words in these categories, which include those referring to sexual or excretory functions and racial, ethnic, religious or social groups, are usually inappropriate and should be treated with caution. There are many ways where their use can be hurtful and upsetting.

Most Social media posts use commissive speech acts like threats and they can be categorized as inappropriate content which is considered as an act of offensiveness in our research because they contain insults to the addressees based on their race, religion, and sexual

orientation. This article will make focus on the analysis from the language side and Ethiopian constitution (FDRE constitution) dominantly.

By looking at the substances of offensive contents in the Amharic language, this article will focus on the units of language such as word, phrase, clause, and sentence. The units of languages were observed from the main data which had been taken from Facebook pages.

The FDRE Constitution has provided the 2005 Criminal Code of Ethiopia, and Information and Network Security Agency Re-establishment Proclamation No.808/2013, Computer crime proclamation 958/2016, Draft proclamation to prevent the dissemination of hate speech and false information. The proclamation categorize the computer crimes in to three those are, First, committed against computer, computer system, computer data or computer network. Second, conventional crime committed by means of a computer. Third, illegal computer content data disseminated through a computer, computer system, or computer network. The online hate speech is mainly covered under the category of illegal contented data, section three of the proclamation more specifically Article 13(1) of the Proclamation criminalizes posting any material online that might be consider as "intimidating" is subject to criminal liability. Similarly, article 14 criminalizes the publication of any content that incites chaos, fear, violence or conflict, would result in the potential imprisonment.

Furthermore, the proclamation comes up with the concept of crimes against liberty and reputation of persons. As per article 13 and 14 of the computer crime proclamation, Whosoever intentionally intimidates or threatens another person or his families with serious danger or injury by disseminating any writing, video, audio or any other image through a computer systems shall be punishable, with simple imprisonment not exceeding three years or in a serious cases with rigorous imprisonment not exceeding five years.

Second, Whosoever causes fear, threat or psychological strain on another person by sending or by repeatedly transmitting information about the victim or his families through computer system or by keeping the victim's computer communication under surveillance shall be punishable with simple imprisonment not exceeding five years or in serious case with rigorous imprisonment not exceeding ten years. Thirdly, disseminates any writing, video, audio or any other image through a computer system that is defamatory to the honor or reputation of another person shall be punishable, upon complaint, with simple imprisonment not exceeding three years or fine not exceeding Birr 30,000or both. Defamation and insult

are also crime as per art. 613 and the following of the criminal code and even entail civil liability [or payment of compensation for the victim] according to art 2044-2049 and 2109 of the civil code.

Therefore, Articles 13 and 14 would discourage whistleblowers, who may have a chance of causing chaos, from forwarding their mess; and it would also discourage writers from publishing any such bedlam in social media.

Characterizing and defining /inappropriate online contents

In recent years, social media conducts have gone beyond the control of the government that it triggers offline chaos. Therefore, there has to be a mechanism to control online activities[14]. The Ethiopian criminal justice system has its own ways and extent of regulation of the conduct on social media to curb the negative effects that may otherwise arise as a result of the unduly exercise of the right and the international human right system helps the criminal justice system through setting uniform standards work on regulating the limit on the right which consider helping the substantive convergence that exist between the countries of the world that in one or other ways makes the prosecution of the criminal conduct on the social media easier.

The case of offensive and violent communication conducted over the internet can be referred to as cyber-hate. It is a narrow and specific form of cyber-bullying and it can be defined as "any use of electronic communications technology to spread racist, religious, extremist or terrorist messages" it is different from cyber-bullying in that offensive post can target not only individuals but it also has implications on whole communities[15]. Brown [15] has defined hate speech as any textual or verbal practice that implicates issues of discrimination or violence against people in regard to their race, ethnicity, nationality, religion, sexual orientation and gender identity.

According to Anis[16], hate speech can occur in different linguistic styles and several acts like insulting, provocation, abusing and aggression. To overcome this problem social media content filtering has been proposed by many researchers.

What are offensive words? According to Amir H. Razavi [16], offensive phrases are words that could mock or insult somebody or a group of people (attacks such as aggression against some culture, a subgroup of the society, race or ideology in a tirade. The authors have listed

several types of offensive language in this category: Taunts, References to handicaps, squalid language, Slurs, Racism, Extremism, Crude language, Provocative language and Taboos (expressions which are forbidden in a certain society/community), Unrefined language: some expressions that lack polite manners and the speaker is harsh and rude

In our research context, we define any textual message or conversation offensive if it is one of the following categories.

Given textual information on social media is defined as offensive if its intent is any of the following

(a) if a post is rude or discourteous (impolite) or exhibiting lack of respect toward certain individuals or group of individuals (ethnic groups):

- **Slurs**: phrases that try to attack a culture or ethnicity in some way,
- **Racism** phrases that intimidate race or ethnicity of individuals,
- **Crude language** expressions that embarrass people, mostly because it refers to sexual matters or excrement,
- **Taboo** expressions which are forbidden in a certain society/community. There are lots of expressions that are forbidden because of what they refer to, not necessarily there are some particular taboo words used in the expression.
- **Unrefined languages** some expressions that lack polite manners and the speaker/writer is harsh and rude.

(b) To cause or capable of causing harm (to oneself or others): harms including physical, psychological, etc.

- **References to handicaps***:* These phrases attack the reader using his\her shortcomings (i.e., "IQ challenged").

(c) related to an activity which is illegal as per the laws of the country:

The rise of irresponsible social media activism and fake news in recent times is being blamed as the catalyst especially for ethnic related violence in various parts of the country[17][18].

(d) has cause of extreme violence:

- *Extremism:* These phrases target some religion or ideologies, Provocative language (expressions that may cause anger or violence).

- Social media posts that make society to get angry with one another.

Based on the above definitions, when we say offensive language detection, implicitly we are talking about every context that falls into one or more of the defined cases. Characterizing, certain expression as 'offensive' have an important role in advancing the values of dignity and equality which underpin **offensive** international human rights law. The term is highly emotive and susceptible. Online offensive contents can be disseminated by individuals and groups and the experience reveals that the most common groups against whom such speech is directed are disabled as well as religious, ethnic minorities. And also there are difficulties in defining the term being problematic to decide whether speech falls under the purview of freedom of expression or under the limitations as while there is no commonly accepted definition to the offensive content on many international, though regional and national laws tried to define the concept and there exists a significant difference in the definitions given to the concept. In our research, we consider hate speech as a type of offensive content. Below we will see definitions for offensiveness and hate speech. Brown[15] had pointed that there are ten clusters of regulations that constrain uses of hate speech, such as (1) group defamation, (2) negative stereotyping or stigmatization, (3) the expression of hatred, (4) the incitement to hatred, (5) threats to public order, (6) acts of mass cruelty, violence, or genocide, (7) dignitary crimes or torts, (8) violations of civil or human right, (9) expression oriented hate crimes, and (10) time, place, and manner restrictions. While Feinberg[19] comes with offense principle which states "offense" in the strict sense of ordinary language specifies a subjective condition the offending act must be taken by the offended person to wrong him whether, in fact, it does or not. As a result, the specific speech may be limited without considering the existence of harm results therein**.** And the author states that there are factors that need to be taken into account to limit the speech based on the offense principle. These factors, inter alia, includes extent, duration and social value of the speech, the ease with which the effect of the speech can be avoided, the motives of the speaker, the number of people offended, the intensity of the offense, and the general interest of the community at large[19].

2.4 Types of content filtering

On researches done up to now, content filtering and the products that offer this service can be divided into Web filtering, the screening of Web sites or pages, and e-mail filters, the screening of e-mail for spam or other objectionable content.

In general, six types of filtering are present they are Browser based filtering, E-mail filtering, Client-side filtering, Content limited ISP's, Network-based filtering, Search engine filtering[18]. Each of them has its own advantage and disadvantages over the other. Filters can be implemented in many different ways: by a software program on a personal computer, via network infrastructure such as proxy servers that provide Internet access.

- *Browser-based filters*

Browser-based content filtering solution is the most lightweight solution to do the content filtering and is implemented via a third-party browser extension.

- *E-mail filters*

E-mail filters act on information contained in the mail body, in the mail headers such as sender and subject, and e-mail attachments to classify, accept, or reject messages. Bayesian filters, a type of statistical filter, are commonly used. Both client and server-based filters are available.

- *Client-side filters*

This type of filter is installed as software on each computer where filtering is required. This filter can typically be managed, disabled or uninstalled by anyone who has administrator-level privileges on the system. These filters are installed directly on the users' computer. It is password protected only those with the password can change or edit filter settings. Others have to just work with implemented filters. They do not have any right to change settings or privilege of any other kind. In this case, a user (with the password) can customize implemented filters to meet certain or specific needs or requirements. This type of filter is best for homes or businesses that need to filter only certain computers or sets of computers. But in case large network or organization this filter doesn't fit in the requirement.

- *Content-limited (or filtered) ISPs*

Content-limited (or filtered) ISPs are Internet service providers that offer access to only a set portion of Internet content on an opt-in or a mandatory basis. Anyone who subscribes to this type of service is subject to restrictions. The type of filters can be used to implement government, regulatory or parental control over subscribers.

- *Network-based filtering*

This type of filter is implemented at the transport layer as a transparent proxy, or at the application layer as a web proxy. Filtering software may include data loss prevention functionality to filter outbound as well as inbound information. All users are subject to the access policy defined by the institution. The filtering can be customized, so a school district's high school library can have a different filtering profile than the district's junior high school library.

- *Search-engine filters*

Search engines like Google, Bing, Yahoo! etc. provide additional security filters to all the users. When a user turns these filters on, the search engine results content that is safe for browsing. All the inappropriate content and text is blocked automatically. But someone who already knows the URL of a site featuring sexually explicit content could still access it without using a search engine. Some search engines like Yahoo! & Lycos offer kids-oriented filters as well. When these filters are turned on, the results that appear are minutely checked and scrutinized if they are appropriate for kids. This type of filter can be beneficial when trying to avoid sites known to contain viruses or pornography that may use misleading descriptions to entice the user to visit. Apart from the types of filtering, web content filtering has eight types of tasks including MIME structure, Text encoding, Images, HTML structure, Phishy URLs, Bad phrasing or Appearance, URL reputations, Fiddy trivia. Event filtering and profile selection are the two types of processes that involve filtering. There are five tools available in filtering, Dansguardian, K9, Open DNS, Squid guard / Squid, Host file[20].

Information filtering techniques

The traditional web page blocking systems goes by the Boolean methodology of either displaying the full page or blocking it completely. With the increased dynamism in the web

pages, it has become a common phenomenon that different portions of the web page hold different types of content at different time instances. instead of completely blocking the page it would be efficient to block only those segments which hold the contents to be blocked[20]. Our research idea is related to blocking/filtering contents that are inappropriate to view. In different areas of research Information filtering technique has been applied, as a kind of techniques for solving information overload problems, has been applied in many application fields, such as web information search/retrieval, medical information classification, dynamic data rectification in industrial process, Websites design, recommender systems, and emergency management. Generally speaking, information filtering can be treated as information classification, in which information is divided into different classes, and therefore machine learning methods, such as evolutionary computation, artificial neural networks, and probabilistic learning, are widely used in information filtering.

As has been discussed in the previous section Information filtering is concerned with the problem of selecting relevant information to the needs of the individuals. Users of a filtering system specify their needs in a profile reflecting their long term wants, i.e. information needs, interests and preferences, relevant to their work, use these profiles to automatically match them with the incoming information. Filter profiles could be constructed to reflect the needs of a group of individuals to cover their common fields of interest.

Even though the filtering system can be of different types, the filtering process can take place in 3 possible locations[21]:

- At the information source

  In this approach, a user posts his/her profile to an information provider. In turn, the user is supplied with information that matches the profile. This type of filtering is called a 'clipping service'. This type of filtering is expensive since information providers usually charge fees per number of search terms, retrieved documents, and connect time (while in the other locations of operation the service is usually free of charge except for the cost of accessing the Internet).
- At a filtering server

Some filtering systems are implemented at special intermediate servers. On the other hand, users post their profiles to the servers, and on the other hand, information providers send data items to these servers, which filter and distribute relevant items to respective users. A server can serve different geographic locations, or specialize in a certain subject of interest. User-agent programs, which are constructed from user models, examine the data items as they arrive and decide for which users each data item is appropriate. The developer suggests this system as a tool for distributing web pages.

- At the user site

  This is the most popular location of filtering operations. Each incoming stream of data items is evaluated by a local filtering system, which removes the irrelevant items or rank-orders them by their relevance. Filtering at the user site implements passive filtering, as the data items flow in automatically, and only then are they evaluated.

Information filtering approaches

As indicated by[21], Information filtering deals with the delivery of information that the user is likely to find interesting or useful. An information filtering system assists users by filtering the data source and deliver relevant information to the users. When the delivered information comes in the form of suggestions an information filtering system is called a recommender system. Because users have different interests the information filtering system must be personalized to accommodate the individual user's interests. This requires the gathering of feedback from the user in order to make a user profile of his preferences. Content-based filtering and collaborative filtering are the two major approaches that exist for information filtering.

Although information filtering is often divided into content-based and collaborative filtering the two approaches can also be used together. Hybrid systems that follow this approach are based on the idea that incorporating both content and social information could lead to a better filtering technique.

Content-based Filtering: also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is

represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items that have been seen by the user. Several issues have to be considered when implementing a content-based filtering system. First, terms can either be assigned automatically or manually. When terms are assigned automatically a method has to be chosen that can extract these terms from items. Second, the terms have to be represented such that both the user profile and the items can be compared in a meaningful way. Third, a learning algorithm has to be chosen that is able to learn the user profile based on seen items and can make recommendations based on this user profile [21]

Collaborative filtering also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie, for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see[22].

A hybrid approach, a system that combines content-based filtering and collaborative filtering could take advantage of both the representation of the content as well as the similarities among users. Although there are several ways in which to combine the two techniques a distinction can be made between two basic approaches. A hybrid approach combines the two types of information while it is also possible to use the recommendations of the two filtering techniques independently.

Social media and it's content

Social media are internet-based applications designed to facilitate social interaction and for using, developing and diffusing information through society. Social media is built on many of the same concepts and technologies as Web 2.0, most basically, the creation and exchange of user-generated content[23].

To extract data from social media, the so-called User Generated Content (UGC) in text form is a valuable resource that can be exploited for many purposes, such as cross-lingual

information retrieval, opinion mining, enhanced web search, social science analysis, intelligent advertising, and so on.

In order to mine the data from Web 2.0, we first need to understand its contents. Analysis of UG content is challenging because of its casual language, with plenty of abbreviations, slang, domain-specific terms, compared to published edited text, with a higher rate of spelling and grammar errors. Standard NLP techniques, which are used to analyze text and provide formal representations of surface data, have been typically developed to deal with standard language and may not yield the expected results on UGC. For example, shortened or misspelled words, which are very frequent in the Web 2.0 informal style, increase the variability in the forms for expressing a single concept[24].

In general, three kinds of data are available for harvest on social media platforms. The first type is content data. It could be user comments on web forums, user profiles on social networking sites (SNS), news articles on news-sharing websites, photos on photo-sharing websites, videos on video-sharing websites, and so forth. The second type is behavior data. Users read online news; comment on posts, blogs, and videos; write reviews for products; listen to music; and watch videos, among many other daily behaviors that are recorded on social media with precise timestamps. The third type is network structure data, that is, visual or hidden hyperlinks among users and/or content. The format of content data could be structured and unstructured[24].

There are three ways to obtain large-scale data on social media platforms. First, the most direct way is to download the data from databases on the web servers. This kind of data is known as “log-file” data. Log-file data is unique for getting behavioral data, such as log-in information, browsing history, which are not visible on web pages. However, it is nearly impossible to get this kind of data unless researchers have close collaborations with the social media companies.

The second method is collecting data through application programming interfaces (APIs)[25][26]. An API is basically an interface of a computer program that allows the software to interact with other software. It is a small script file written by users, following the rules specified by the web owner, to download data from its database other than web pages. To put it simply and informally, APIs are special URLs that web owners intentionally provided for developers to download data from their databases. As part of their business

model, social media companies often make their APIs available to third parties. The primary purpose of providing APIs is to enable the development and enhancement of social media services. However, the API is also an interface for researchers to collect data from social media platforms. Through small software scripts, researchers can access the API to retrieve, store, and manipulate digital traces left by the users of a service for further empirical analysis. The third method is web scraping. This method is particularly useful for those websites that do not provide APIs. Where data available on the website are not available through the API, an alternative method is to crawl the social media website with an automated script that explores the website and collects data using HTTP requests and responses. Web scraping is the process of taking unstructured information from web pages and turning it into structured information that can be used in a subsequent stage of analysis[24]. It is possible to collect data from APIs about location, demographic information, newsfeed, uploaded material, the social graph, and so on. Evidently, the users who are most likely to generate most of these data in the systems are hardly representative of the entire population of users of social media[27].

The structure of social media data is unorganized and is displayed in different forms such as text, voice, images, and videos.

Data mining techniques are capable of handling the three dominant disputes with social media (SM) data which are **size, noise, and dynamism**[28]. *SM* data sets are very voluminous and require automated information processing for analyzing it within a reasonable time. As data mining also requires huge data sets to mine remarkable patterns from data, *SM* sites appear to be perfect sites to work on especially where opinion/sentiment expression is involved [22]. *SM* data sets are also characterized by noisy data such as spam blogs and irrelevant tweets in the case of twitters. The dynamism in *SM* data sets causes it to evolve rapidly over time and data mining techniques are versatile in handling such dynamic data. The use of data mining techniques on *SM* data is an enabling factor for advanced search results in search engines and also helps in a better understanding of data for research and organizational functions[28]. For the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral we can apply Sentiment Analysis.

Sentiment analysis uses data mining processes and techniques to extract and capture data for analysis in order to discern the subjective opinion of a document or collection of documents, like blog posts, reviews, news articles, and social media feeds like tweets and status updates.

People make judgments about the world around them when they are living in the society. They make positive and negative attitudes about people, products, places, and events. These types of attitudes can be considered as sentiments. Sentiment analysis is the study of automated techniques for extracting sentiments from written languages. The growth of social media has resulted in an explosion of publicly available, user-generated text on the World Wide Web. These data and information can potentially be utilized to provide real-time insights into the sentiments of people. Most social media data sources are dominated by non-English information in Ethiopia and there are various claims that have been made about the text in social media text being noisy and offensive.

At its most basic, sentiment analysis is a social media analytics tool that involves checking how many negative and positive keywords are present in a chunk of conversation. If there are more positive keywords than negative, it is considered positive content[9]. With the help of sentiment analysis, we can check how negative or positive a comment or post is on social media.

2.5 How social media data are filtered

There are lots of techniques to filter social media data. In most researches, techniques widely used for social media text filtering include machine learning, Rule-based and sentiment analysis. In this section, we will discuss each and every technique one by one.

Social media filtering by using machine learning (ML), ML is a series of algorithms that enable computers to identify patterns in data and classify it in clusters. This is perfectly adapted to unstructured data as social media postings don't follow any rules. It is usually a mix of text, images, sounds, and video. The results of such an analysis can give actionable insights about the selected users. Machine learning with Natural language processing offers valuable clues about the age, gender, location, and preferences of the authors of posts on social media. The data coming from an NLP API can help with customer segmentation based on real-world data instead of statistics or educated guesses. There are several reasons to deploy ML in social media analysis which are dictated by the 3 Vs. of Big Data (volume,

velocity, and variety). Filtering can also be done through a flexible **rule-based system**, that allows users to set the filtering criteria to be applied to their walls, and a Machine Learning technique based soft classifier algorithm automatically labeling messages in support of content-based filtering. To do this, the Black List (BL) mechanism is proposed in many systems, which avoids undesired creator messages. BL is used to determine which user should be inserted in Black List and decide when the retention of the user is finished. Machine Learning Text Categorization (MLTC) is also used to categorize the short text messages.

Many kinds of machine learning algorithms are used to build sentiment classifiers. We will discuss four in-depth in this section: multinomial naive Bayes and multinomial logistic regression, also known as the maximum entropy or MaxEnt classifier. These exemplify two ways of doing classification. Generative classifiers like naive Bayes build a model of each class. Given an observation, they return the class most likely to have generated the observation. Discriminative classifiers like logistic regression instead learn what features from the input are most useful to discriminate between the different possible classes. While discriminative systems are often more accurate and hence more commonly used, generative classifiers still have a role.

Multinomial Naïve Bayes

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. It assumes an underlying probabilistic model. And it allows us to capture uncertainty about the model in a principled way by determining the probabilities of the outcomes.

The Multinomial Naïve Bayes classifier is also based on the Bayes Theorem as referred by[29], it is a popular classifier used for document-level sentiment classification and relatively yields good output and performance. This algorithm can be trivially used and applied for the data stream as it plays straightforwardly for updating the counts that are required to estimate the conditional and algorithmic probabilities.

Advantages

- It is easy and fast to predict the class of the test data set, performs well in multi-class prediction, easily trained even with a small dataset

- When the assumption of independence holds, a Naive Bayes classifier performs better compared to other models with less training data.
- It performs well in the case of categorical input variables compared to a numerical variable(s). For numerical variables, a normal distribution is assumed.

Disadvantages

- If the categorical variable has a category (in a test data set), which was not observed in the training data set, then the model will assign a 0 (zero) probability and will be unable to make a prediction.
- On the other side, naive Bayes is also known as a bad estimator, so the probability outputs are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors that are completely independent.
- It assumes every feature is independent, which isn't always the case

Support Vector Machine (SVM)

Support Vector Machine (SVM) as described by[30], [31] is a supervised learning model with efficient results in traditional text categorization leaving behind Naïve Bayes and MaxEnt. SVM was originally introduced by[32] it basically locates the best possible boundaries to separate between positive and negative training samples and is extensively used because of their exceptional performance over other methods used in most machine learning models as referred by[33] and[29]. There are multiple extensions available for SVM making it more efficient and adaptable to real-world requirements.

Advantages

- SVM Classifiers offer good accuracy and perform faster prediction compared to the Naïve Bayes algorithm. They also use less memory because they use a subset of training points in the decision phase. SVM works well with a clear margin of separation and with high dimensional space.

Disadvantages

- SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes. It works poorly with overlapping classes and is also sensitive to the type of kernel used.

Decision Tree (DT)

A DT is a hierarchical model composed of decision rules that recursively split independent variables into homogeneous zones[34]. The objective of the DT building is to find the set of decision rules that can be used to predict the outcome from a set of input variables. A DT is called a classification or a regression tree if the target variables are discrete or continuous, respectively[34]. The main advantage of DT is that DT models have the capability of modeling complex relationships between variables. They can incorporate both categorical and continuous variables without strict assumptions with respect to the distribution of the data[35]. In addition, DTs are easy to construct and the resulting models can be easily interpreted.

Furthermore, the DT model results provide clear information on the relative importance of Input factors[34]. The main disadvantage of DTs is that they are susceptible to noisy data and that multiple output attributes are not allowed.

Advantages

- Computationally cheap to use, easy for humans to understand results and it can deal with irrelevant features also
- Decision trees are easy to interpret and visualize.
- It requires fewer data preprocessing from the user, for example, there is no need to normalize columns.
- It can be used for feature engineering such as predicting missing values, suitable for variable selection.
- The decision tree has no assumptions about distribution because of the non-parametric nature of the algorithm.

Disadvantages:

- Prone to over-fitting.

Logistic regression

It's a classification algorithm, which is used where the response variable is *categorical*. The idea of Logistic Regression is to find a relationship between features and the probability of a particular outcome. Multinomial Logistic Regression is the regression analysis to conduct when the dependent variable is nominal with more than two levels.

Similar to multiple linear regressions, the multinomial regression is a predictive analysis. Multinomial regression is used to explain the relationship between one nominal dependent variable and one or more independent variables. Logistic regression measures the relationship between the dependent variable (our label, what we want to predict) and the one or more independent variables (or features), by estimating probabilities using it's underlying logistic function. These probabilities must then be transformed into binary values in order to actually make a prediction. This is the task of the logistic function, also called the sigmoid function. The Sigmoid-Function is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits. These values between 0 and 1 will then be transformed into either 0 or 1 using a threshold classifier. We want to maximize the likelihood that a random data point gets classified correctly, which is called Maximum Likelihood Estimation. Maximum Likelihood Estimation is a general approach to estimating parameters in statistical models. We can maximize the likelihood of using different methods like an optimization algorithm. Newton's Method is such an algorithm and can be used to find maximum (or minimum) of many different functions, including the likelihood function. Instead of Newton's Method, we could also use Gradient Descent.

Logistic regression[36] belongs to the family of classifiers known as the exponential or log-linear classifiers. Like naive Bayes, it works by log-linear classifier extracting some set of weighted features from the input, taking logs, and combining them linearly (meaning that each feature is multiplied by a weight and then added up). Technically, logistic regression refers to a classifier that classifies an observation into one of two classes, and multinomial logistic regression is used when classifying into more than two classes.

Logistic regression is much more robust to correlated features; compared to Naïve Bayes classifier; if two features f1 and f2 are perfectly correlated, regression will simply assign half the weight to w1 and half to w2. Thus when there are many correlated features, logistic regression will assign a more accurate probability than naive Bayes. Nonetheless, these less accurate probabilities often result nonetheless in naive Bayes making the correct classification decision.

Advantages / Disadvantages

It is a widely used technique because it is very efficient, does not require too many computational resources, it's highly interpretable, it doesn't require input features to be scaled, it doesn't require any tuning, it's easy to regularize, and it outputs well-calibrated predicted probabilities.

Like linear regression, logistic regression does work better when attributes that are unrelated to the output variable, as well as attributes that are very similar (correlated) to each other, are removed. Therefore Feature Engineering plays an important role in regards to the performance of Logistic and also Linear Regression. Another advantage of Logistic Regression is that it is incredibly easy to implement and very efficient to train. We typically started with a Logistic Regression model as a benchmark and try using more complex algorithms from thereon.

Because of its simplicity and the fact that it can be implemented relatively easy and quick, Logistic Regression is also a good baseline that can be used to measure the performance of other more complex Algorithms.

A disadvantage of it is that we can't solve non-linear problems with logistic regression since its decision surface is linear.

Logistic Regression is also not one of the most powerful algorithms out there and can be easily outperformed by more complex ones. Another disadvantage is its high reliance on a proper presentation of data. This means that logistic regression is not a useful tool unless all the important independent variables are already identified. Since its outcome is discrete, Logistic Regression can only predict a categorical outcome. It is also an Algorithm that is known for its vulnerability to overfitting. There are ways to use Logistic Regression to do multiclass classification with sklearn and why it is a good baseline to compare other Machine Learning algorithms with. One vs one (OvO) and one vs all (OvA) tricks can be used to predict multiple classes. In the next section, we will discuss, feature extraction, word embedding method.

Text feature extraction is the process of taking out a list of words from the text data and then transforming them into a feature set that is usable by a classifier. This work emphasizes on the review of available feature extraction methods. The following techniques can be used for extracting features from text data[37].

Word embedding

Computers are unable to understand the concepts of words. It requires data to be converted into a numeric format to perform any machine learning task. In order to perform such tasks, various word embedding techniques are being used i.e., Bag of Words, TF-IDF, word2vec to encode the text data. In order to process natural language, a mechanism for representing text is required. Word embeddings are commonly used in many Natural Language Processing (NLP) tasks because they are found to be useful representations of words and often lead to better performance in the various tasks performed. Given its widespread use, in this paper, we will try to introduce the concept of word embedding's with respect to Amharic language. Using different word embeddings we can represent the sentence differently in numbers. Here we will discuss TF-IDF and Word2Vec.

So a natural language modeling technique like Word Embedding is used to map words or phrases from a vocabulary to a corresponding vector of real numbers. As well as being amenable to processing by ML algorithms, this vector representation has two important and advantageous properties:

1. Dimensionality Reduction - it is a more efficient representation
2. Contextual Similarity - it is a more expressive representation

- TF-IDF

Using TF-IDF embeddings, word will be represented as a single scalar number based on TF-IDF scores. TF-IDF is the combination of TF (Term Frequency) and IDF (Inverse Document Frequency). TF gives the count of word t in document d. Mathematically we can write tf(t,d). IDF gives information about how the word is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word.

Mathematically, idf(t,D) = log (N/dfi) , where N or |D| = Total Number of Document, and dfi = Number of document where the term t appears.

TF-IDF(t, d, D) = tf(t,d) . idf(t, D), the number of times certain words (w) appear in a document (d). 'D' represents the ***entire text corpus.*** The absolute value sign-on 'D' represents the size of the corpus, how many documents there are in total. In the bottom, 'df(d,w)', represents how many ***documents*** the ***word*** appears in. We then end up with a logarithmically scaled value of the number of documents in the corpus divided by the number of times word w appears throughout the corpus for the idf(w, D) value.

When we see the Bag of Words approach, it is known it often results in huge, very sparse (containing many zeros) vectors, where the dimensionality of the vectors representing each document is equal to the size of the supported vocabulary. On large data sets, this could cause performance issues. Additionally, one-hot encoding does not take into account the semantics of the words. So words like ቆንጆ and አማላይ are considered to be two different features. While we know that they have a very similar meaning. Word embeddings address these two issues. Word Embedding aims to create a vector representation with a much lower-dimensional space.

Word Embedding is used for semantic parsing, to extract meaning from text to enable natural language understanding. For a language model to be able to predict the meaning of text, it needs to be aware of the contextual similarity of words. For instance, that we tend to find offensive words (like መንጋ, ሰፋሪ , ጠባብ or ዘረኛ) in sentences where they're used to insult certain ethnic group, but wouldn't expect to find those same concepts in such close proximity to, say, the word ደስ _የሚል, መልካም, ደግ, ሩህሩህ.

The vectors created by Word Embedding preserve these similarities, so words that regularly occur nearby in the text will also be in close proximity in vector space. In general, word embedding is a means of building a low-dimensional vector representation from the corpus of text, which preserves the contextual similarity of words; the semantic relationships between words are reflected in the distance and direction of the vectors. The standard mechanism for text representation is **word vectors** where words or phrases from a given language vocabulary are mapped to vectors of real numbers. The way word2vec works is - two words are similar if they appear in a similar context[38].

The purpose and usefulness of Word2vec is to group the vectors of similar words together in vector-space. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words. It does so without human intervention.

Given enough data, usage, and contexts, Word2vec can make a highly accurate guess about a word's meaning based on past appearances. Those guesses can be used to establish a word's association with other words (e.g. "ሰውዬው" is to "ልጁ" and "ሴትየዋ" is to "ልጅቷ"), or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis, and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management.

The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words. There are two main training algorithms for word2vec, one is the continuous bag of words (CBOW), and another is called skip-gram. The major difference between these two methods is that CBOW is using context to predict a target word while skip-gram is using a word to predict a target context. Generally, the skip-gram method can have a better performance compared with CBOW method, for it can capture two semantics for a single word[39].

An interesting feature of word vectors is that because they're numerical representations of contextual similarities between words (which might be gender, tense, geography or something else entirely), they can be manipulated arithmetically just like any other vector[40].

The vectors are very good at answering analogy questions of the form *a* is to *b* as *c* is to? For example, ወንድ is to ሴት as ኦንት is to? (ኦክስት) using a simple vector offset method based on cosine distance.
For example, here are vector offsets for six-word pairs illustrating the gender and capital city relation:

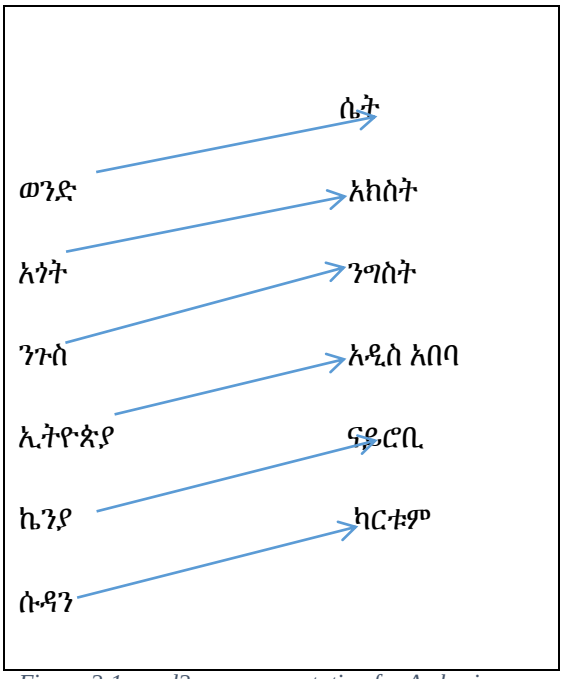


Figure 2-1 word2vec representation for Amharic

According to Mikolov, Skip Gram works well with a small amount of data and is found to represent rare words well.

Word2Vec - The Skip-Gram Model

Word embedding is one of the most popular representations of document vocabulary. It is capable of capturing the context of a word in a document, semantic and syntactic similarity, relation with other words, etc. The neural network is trained to do the following, given a specific word in the middle of a sentence (the input word), it looks at the words nearby and picks one at random. The network tells the probability of every word in the vocabulary of being the "nearby word" that is chosen[41]. When we say "nearby", there is actually a "window size" parameter to the algorithm. A typical windows size might be 5, meaning 5 words behind and 5 words ahead (10 in total). The output probabilities are going to relate to how likely it finds each vocabulary word nearby on the input word. For example, if you gave the trained network the input word "የሚያምር", the output probabilities are going to be much higher for words like "ደስ የሚል" and "አስደናቂ" than for unrelated words like "ለማውደምና" and "ለማዳከም". The neural network is trained to do this by feeding it word pairs found in the training documents. The below example shows some of the training samples (word pairs) we would take from the sentence "በብሄር ወይም በፖለቲካ ምክንያት ነው የምትደግፉ የምትቃወሙ ለምትሉ

የአዕምሮ ስንኩላን ነው።” We’ve used a small window size of 2 just for the example. The word highlighted in blue is the input word.

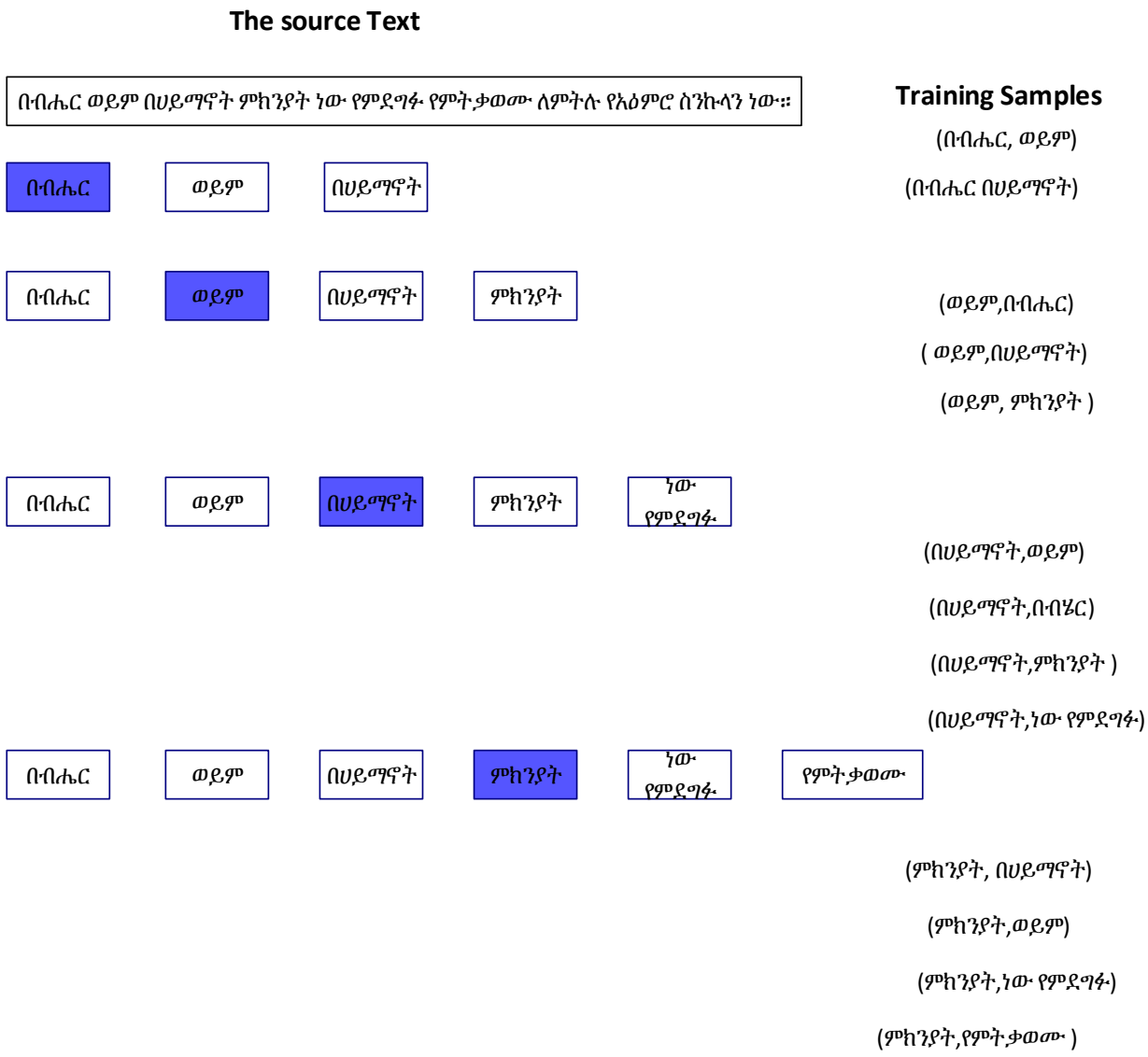


Figure 2-2 Example of how word2vec works

The network is going to learn the statistics from the number of times each pairing shows up. So, for example, the network is probably going to get many more training samples of (“የሚያምር”, “ደስ የሚል”) than it is of (“የሚያምር”, “ለማውደምና”). When the training is finished, if we give it the word “የሚያምር”, as input, then it will output a much higher probability for “ደስ የሚል” or “አስደናቂ” than it will for “ለማውደም”.

In a nutshell, Word Embedding turns text into numbers. This transformation is necessary because many machine learning algorithms (including deep nets) require their input to be vectors of continuous values; they just won't work on strings of plain text. In relation to this, TF-IDF is a word-document mapping (with some normalization). It ignores the order of words and gives nxm matrix (or mxn depending on implementation) where n is the number of words in the vocabulary and m is the number of documents. Word2Vec, on the other hand, gives a unique vector for each word based on the words appearing around the particular word. TF-IDF is obtained from straightforward linear algebra. Word2Vec is obtained from the hidden layer of a two-layered neural network. TF-IDF can be used either for assigning vectors to words or to documents. Word2Vec can be directly used to assign a vector to a word but to get the vector representation of a document further processing is needed. Unlike TF-IDF Word2Vec takes into account the placement of words in a document (to some extent).

2.6 Applications

Many information filtering systems have been developed in recent years for various application domains. Some examples of filtering applications are Personal email filters based on personal profiles, List servers or newsgroups or individuals, browser filters that block non-valuable information, filters designed to give children access them only to suitable pages, filters for e-commerce applications that address products and promotion to potential customers only. The most known application area is the recommender system. Recommender systems have become popular, which is a type of information filtering system that predicts the preference of the user. It gives importance to user interest and recommends an item.

Recommender systems can work in two ways[42]: Collaborative filtering and Content-based filtering.

The collaborative filtering system is mainly based on the user's preferences, actions and predicts what users will like based on his/her similarities to other users, whereas content-based filtering focuses on user interest and select items based on it. It suggests the best-matched item based on the previously chosen item. Our researches focus is content-based filter; text classification is an important part of content-based filtering. Content-based filtering also referred to as cognitive filtering, recommends items based on a comparison

between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items that have been seen by the user. It actually works well on the Machine learning-based text classifiers. In a Machine learning approach, it learns from training data and creates classifiers for the classification of the new dataset. The main task of text classification is to assign each text predefined category of text. The classification algorithms such as Support Vector Machines, Naive Bayes, Neural network, and Decision trees can be used for text classification...

2.7 Evaluation methods

Masahiro[43] evaluated the proposed information filtering method using the sub-string index method by a simulation using the collected data. Their system consisted of precision-recall curves of predicting interesting and not-interesting articles, obtained by varying the threshold scores that determine whether the articles are interesting or not. **Precision and recall** are two extremely important model evaluation metrics. In a text classification task, the precise definition of recall is the number of true positives divided by the number of true positives plus the number of false negatives. True positives are data points classified as positive by the model that actually is positive (meaning they are correct), and false negatives are data points the model identifies as negative that actually are positive. The formula to compute precision and recall as outlined below.

Recall = true positives / true positives + false negatives

Precision = true positives / true positives + false positive

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and the worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is as shown in the following equation:

F= 2* precision * recall/precision + recall

In a classification task, a precision score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labeled correctly) whereas a recall of 1.0 means that every item from class C was labeled as belonging to class C (but says nothing about how many other items were incorrectly also labeled as belonging to class C).

Sadaf[44] conducted their experiments with two different available data-sets for this research. The first data-set contains text such as video titles, descriptions, and comments given by different users on YouTube, which were collected by various researchers after crawling on the website. Later on, from that data-set, they chose around 200 different video titles/comments and carried out an online survey. The survey involved 20 participants and helped them in mining the opinions of the people, and labeling or categorizing the text as positive or negative.

The text was then parsed through the different document preprocessing steps and then was semantically analyzed to be ready for machine learning. The second data-set contained comments from different social sites. The comments were already classified as negative and positive classes. They selected 3875 comments randomly from the testing corpus to test the features, they engineered for classification.

Most research to date on social media text has used sentiment analysis, the automated extraction of expressions of positive or negative attitudes from texts posted on social media. But little attention was given to Negative posts that appear on Ethiopian social media. Different techniques, approaches, evaluation methods and etc. have never been discussed in relation to Ethiopian social media.

In our research, in order to evaluate the characteristics of text in different social media sources, we will assemble the datasets from across the spectrum of popular social media sites, varying in terms of document length, the number of authors/editors per document, and the level of text editing. Because of increasing importance to analyze multilingual data rather than unilingual data we will follow the sentiment analysis approach to perform social media analysis and evaluation. Amharic posts on Facebook social media will be our focus while designing the Model.

Chapter 3 : Related work

3.1. Introduction

There had been many efforts done previously in creating information filtering techniques that can be related to the research challenges that we will be addressing in our study, which is information filtering of social media Amharic texts based on sentiment analysis. There are a number of studies on sentiment analysis and a few are closely related to our study. We focused on two aspects for selecting studies of our discussion, namely the language family (Amharic language sentiment analysis using rule-based versus supervised machine learning) and social media Information filtering.

There is a growing number of research works on social media information filtering and development and refining the automated techniques of sentiment classification and analysis. Many researchers have worked on sentiment analysis techniques via different approaches (Lexical, Machine Learning, and Hybrid) however, in-depth analysis and review of the latest literature on sentiment analysis with machine learning algorithms for under-resourced language like Amharic is still required.

3.2 Sentiment Analysis for Amharic Language

We encountered only little sentiment analysis study on Amharic language by Gebremeskel[9] and Wondwossen Mulugeta[8].

Gebremeskel[9] followed the term counting approach to sentiment analysis. In the model the author proposed, once sentiment words and valence shifter words are detected using a sentiment lexicon, terms are assigned a weight by considering the effect of diminisher, intensifier or negation terms. The total polarity weight of a review is calculated by adding the polarity weight of individual sentiment terms. Although the term counting approach may be considered as a valuable alternative for underdeveloped languages like Amharic which are facing challenges in building a corpus, systems developed using this approach are not easy to scale up. Besides, machine learning performs with less human intervention.

A multi-scale sentiment analysis model for Amharic online posts written in Ethiopic scripts was studied by Wondwossen Mulugeta[8] using a supervised machine learning approach. The objective of the study was to determine a multi-scale sentiment sentence based on the

polarity weight value of sentiment words. To achieve the objective, the author has prepared a sample corpus that contains 608 posts. The corpus was collected from social media sources such as Facebook, Twitter, DireTube and Ethiopian reporter websites. The author employed preprocessing activities in the corpus dataset before the actual sentiment classification. After the preprocessing activities were done, the corpus was manually annotated by giving polarity values and sentiment intensity scale values. The author adopted two-scale schemes which are scale positive sentiment further as +1.

Zewdie [45], developed an apache spark based model to classify Amharic Facebook posts and comments into hate and not hate. Authors employed Random forest and Naïve Bayes for learning and Word2Vec and TF-IDF for feature selection. Tested by 10-fold cross-validation, the model based on word2vec embedding performed best with 79.83%accuracy. The proposed method achieved a promising result with a unique feature of spark for big data. Their article focused only on hate speech detection from social media posts and comments. It didn't analyze the different aspects of a category of hate, either hate with politics, ethnicity, religion, and socio-economy.

Sentiment Analysis for other Non-English Languages

Most sentiment analysis techniques are designed for English. In recent years, however, several works focused on non-English languages. Atteveldt[6] Used machine learning techniques to determine the polarity of political news stories in Dutch. They extracted lexical and syntactic features besides three different clustering of similar words based on annotated material. Ghorbel and Jacot[5] devised a supervised learning strategy using linguistic features obtained through part-of-speech tagging and chunking as well as semantic orientation of words obtained from the SentiWordNet sentiment analysis tool to classify the polarity of movie reviews in French. Since SentiWordNet is for English, the authors translated the French words into English before getting their semantic orientation. Zhang[46] addressed the challenges that are unique to the Chinese language. They evaluated a rule-based polarity classification approach against different machine learning approaches. In literature, the research on sentiment analysis in the Amharic language is limited, to the best of our knowledge. Our work differs from [8] as we use four machine learning algorithms comparatively to check the performance difference on social media Amharic

texts, and our dataset is relatively larger. The objective of the research was to Assign document sentiment whereas our main goal is filtering offensive contents from social media by detecting document sentiments.

3.3 Filtered wall to filter unwanted messages from social media

Various researchers across the globe have been doing their research on message (information) filtering. In this section, we will discuss researches done by different scholars to create a Filtered wall by using several approaches.

M. Vanetti[47] proposed a system enforcing content-based message filtering conceived as a key service for On-line Social Networks (OSNs). The system allows OSN users to have direct control of the messages posted on their walls. This is achieved through a flexible rule-based system, that allows a user to customize the filtering criteria to be applied to their walls, and a Machine Learning based soft classifier automatically producing membership labels in support of content-based filtering. They have presented a system to filter out undesired messages from OSN walls. The system exploits an ML soft classifier to enforce customizable content-depended filtering rules. Moreover, the flexibility of the system in terms of filtering options is enhanced through the management of Blacklists (BLS). The proposed system may suffer from problems similar to those in the specification of privacy settings in OSN. As future work, the authors directed their willingness to integrate contextual information related to the name of all the groups in which the user participates, appropriately weighted by the participation level. The authors also indicated the importance of stress contextual information which is related to the environment preferred by the user who wants to post the message; thus, the experience that social media user can try using the proposed system.

V. Subramaniyaswamy[48]enforced powerful techniques to achieve the filtering of messages in OSN user walls using the Naive-Bayesian algorithm. In this research, the authors considered Filtering as a process of removing unwanted material. Likewise, in OSN the filtering is used to remove the unwanted messages, comments from being posted on the user's wall. In this system, the people who are featured in the "Blocked List" cannot post on the user's wall. Technically, the OSN uses the Naive Bayesian and rule-based text classification along with "Bag-Of-words" (document with enormous words). Here, the

collection of offensive, abusive and vulgar words is framed as a document through which the first stage of filtering gets completed by checking the tokenized words (separated words that are taken from posted messages) against the document. If it is not present in the document, then it goes for "Analysis by experts" (Third-party consultation).

Allu S.[49] Proposed and experimentally evaluated an automated system, called Filtered Wall (FW), able to filter unwanted messages from OSN user walls. The authors exploited Machine Learning (ML) text categorization techniques to automatically assign with each short text message a set of categories based on its content. The authors incorporated the neural model within a hierarchical two-level classification strategy. In the first level, the RBFN categorizes short messages as Neutral and Non-neutral. In the second stage, Non-neutral messages are classified producing gradual estimates of appropriateness to each of the considered categories. Besides classification facilities, the system provides a powerful rule layer exploiting a flexible language to specify filtering rules (FRS), by which users can state what contents, should not be displayed on their walls. FRS can support a variety of different filtering criteria that can be combined and customized according to the user needs. More precisely, FRS exploit user profiles, user relationships as well as the output of the ML categorization process to state the filtering criteria to be enforced. in addition, the system provides the support for user-defined blacklists (BLS), that is, lists of users that are temporarily prevented to post any kind of messages on a user wall and provide the spam tab, then the system allows page admin to set up a keyword moderation blocklist and enable a profanity blocklist that filters wall posts and comments by users into the page wall's spam tab. Admin can configure the list from the manage permissions tab of the page admin interface.

There are many more other researches tried to design a filtered wall that can remove unwanted message from online social networks. But these researches do not cover various languages across the world. Their systems are limited for filtering languages like Amharic.

Filtering social media contents by using machine learning techniques

In this section, we will exploit researches done on Machine Learning (ML) text categorization techniques to automatically assign with each short text message a set of categories based on its content. Some exploited Machine Learning (ML) text categorization

techniques based soft classifier automatically labeling messages with the help of content-based filtering, decision trees[50], naive-Bayes, rule induction, neural networks[50], K-nearest neighbors(KNN)[50], and support vector machines. Although many approaches have been proposed, automated text classification is still a major area of research primarily because the effectiveness of current automated text classifiers is not faultless and still needs improvement[29].

Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness[51]. However, its performance is often degraded because it does not model text well. Support vector machines (SVM), when applied to text classification provide excellent precision, but poor recall. One means of customizing SVMs to improve recall is to adjust the threshold associated with an SVM. Shanahan and Roma [52] described an automatic process for adjusting the thresholds of generic SVM with better results.

Johnson[53] Described a fast decision tree construction algorithm that takes advantage of the scarcity of text data and a rule simplification method that converts the decision tree into a logically equivalent rule set.

Pang[54] Compared the performance of different machine learning techniques on movie reviews taken from the IMDB movie database. The SVM classifier is shown to yield the best performance. Their features included unigrams, bigrams, part of speech information, and the positions of the terms. Among these, the unigrams were found to yield better performance.

Alexander and Patrick[3] used a corpus of 300000 text posts from a tweeter, collected and split automatically into three sets of text, they use emoticons to split the text into +ve / –ve emotions or objective (which contains no emoticons). Scheible and Schutze[55] present a novel graph-theoretic method for the initial annotation of high-confidence training data for bootstrapping sentiment classifiers. They introduce a new semi-supervised sentiment classifier that integrates lexicon induction with document classification. Their experiments are on customer reviews of multi-domains [12]. David and his group [3] use a supervised K-Nearest Neighbor (KNN) as a classifier. They work on different topics of English tweets. They used hashtags and smileys as features in the classification process. Also, Barbosa and Feng[4] applied their approach to tweets data and use an SVM classifier to classify the

sentiment of tweets using abstract features. Abbasi[13]noted that one of the bottlenecks in applying a supervised learning approach is the manual effort involved in annotating a large number of training data. Riloff and Wiebe[14,15] try to solve the problem by using a bootstrapping approach to label training data automatically. Their approach performed well, achieving 77% recall with 81% precision.

Tony Mullen[30]introduced an approach to sentiment analysis that uses support vector machines (SVMs) to bring together diverse sources of potentially pertinent information. The authors' approach emphasizes the use of a variety of diverse information sources, and they used SVMs to provide the ideal tool to bring these sources together. The authors described the methods used to assign values to selected words and phrases and introduced a method of bringing them together to create a model for the classification of text. This research has allowed methods of assigning semantic values to phrases and words within a text to be exploited in a more useful way than was previously possible, by incorporating them as features for SVM modeling, and to explicit topic information to be utilized, when available, by features incorporating such values. While doing experimentation, the authors divided the dataset into two classes, training data, and testing data. In training data, reviews corresponding to a below-average rating were classed as negative and those with an above-average rating were classed as positive. The first dataset consisted of a total of 1380 Epinions.com movie reviews, approximately half positive and half negative. The second dataset consists of 100 record reviews from the Pitchfork Media online record review publication,3 topic-annotated by hand.

Ahmed[56] proposed a method that improves the performance of kNN based text classification by using a well-estimated parameter. Some variants of the kNN method with different decision functions, k values, and feature sets were proposed and evaluated to find out adequate parameters.

During machine learning text classification, as Emmanouil K[56] indicated, the level of difficulty of text classification tasks naturally varies. As the number of distinct classes increases, so does the difficulty, and therefore the size of the training set needed. In any multi-class text classification task, inevitably some classes will be more difficult than others

to classify. Reasons for this as discussed by the authors are Very few positive training examples for the class, and/or Lack of good predictive features for that class.

Though there is voluminous literature stating the capabilities of different types of text classification techniques, the spread of these techniques in advanced fields like Artificial Intelligence (AI)/Machine Learning (ML) is seldom reported.

Other than designing the Filtered wall system for social media information filtering, researchers also tried to detect and prevent SPAMs in OSN. In this section, we will be discussing researches done for online social network SPAM detection. OSN can also be an effective mechanism for spreading attacks. Popular OSNs are increasingly becoming the target of phishing attacks launched from large botnets. In most researches, the detection of spam is carried out by applying the Machine learning techniques to provide secure online social networks to the user[45]. Girisha[45] introduced a machine learning-based spammer detection model for social network(Twitter). The solution considers the user's content and behavior feature, and they applied them into KNN based algorithm for spammer classification. The authors worked on live data. In this research, the authors outlined different types of spam.

In order to detect and prevent spammers in social networks, several methods have been proposed and developed by many researchers. Gianluca[57] created 900 profiles on Facebook, MySpace, and Twitter, 300 on each platform. The purpose of these accounts was to log the traffic (e.g., friend requests, messages, invitations) they receive from other users of the network. They call these accounts honey-profiles. After having created their honey-profiles, they ran scripts that periodically connected to those accounts and checked for activity. They decided that their accounts should act in a passive way. Therefore, they did not send any friend requests but accepted all those that were received. In a social network, the first action a malicious user would likely execute to get in touch with his victims is to send them a friend request. After having acknowledged a request (i.e., accepted the friendship on Facebook and MySpace or started following the user on Twitter), they logged all the information needed to detect malicious activity. Their scripts ran continuously for 12 months for Facebook and for 11 months for MySpace and Twitter periodically visiting each account. The visits had to be performed slowly (approximately one account visited every 2

minutes) to avoid being detected as a bot by the social networking site and, therefore, having the accounts deleted. During their study, they received a total of 4,250 friend requests Overall, they observed 85,569 messages. They started to manually check all the profiles that contacted them. During this process, they noticed that spambots share some common traits, and formalized them in features that they then used for automated spam detection.

During our review, it is seen that spam detection in social networks uses Decision Tree, SVM, Random Forest and Naïve Bayesian approach which is highly effective and a combination of spam prevention filters will give higher accuracy.

A research done by Bo Pang, Lillian Lee and Shivakumar Vaithyanathan [54], focuses on classifying different movie reviews into three categories namely; positive, negative and neutral. They used different machine learning methods namely naïve Bayesian method, support vector machines, and maximum entropy method. In their research, they collected a set of proposed negative and positive words for a movie review from an audience and used the three methods to classify whether a review is in one of the three categories. The reviews were taken from the internet movie database (IMDb). With all the methods by changing the size of the word list, they were able to achieve a peak accuracy of 82.9%.

3.4. Summary

In this chapter, gaps of different researches on sentiment analysis and information filtering are investigated and analyzed. From the analysis of the related works, we came up with the following major problems of present-day information filtering and sentiment analysis as a social media monitoring tool for Amharic.

Although English has been the target language in most sentiment analysis research, recent efforts extend the focus to other languages such as Amharic. Basic machine learning techniques as simple as Naive Bayes have been used to achieve baseline results [8],[9][10]. However, these systems require lots of feature engineering work prior to applying any machine learning method. There is a lack of widely available benchmark for comparing different machine learning algorithms for Amharic Sentiment Analysis of social media text.

Most of the researches on sentiment analysis of social media has focused on the English language and European languages. Only a few researches, try to solve the problems of morphologically rich languages such as Amharic Language. Despite its large number of

speakers, the Amharic language hasn't got much attention from researchers. Furthermore, social media is a source of raw data which includes (and is not limited to) the following metrics: Shares, Likes, Conversations, Comments, mentions, impressions, and most important status updates. It is a place where unstructured information exists. Because sentiment essentially relates to feelings; attitudes, emotions, and opinions of peoples, it is recognized as the most effective social media content filtering technique. Simply put, for the purposes of social media, it is the process of determining the author's opinion conveyed in a post. With social media monitoring, sentiment analysis is much harder because there isn't a defined contextualization process. People talk about everything and anything under the sun and their feelings and opinions toward certain topics are almost impossible to contextualize for a computer. The absence of Amharic text sentiment analysis of social media content pushes us to build an Amharic content filtering system.

Information filtering for major languages like English, Japanese and Spanish has been conducted since the 1960s. But no attempt was made in Information filtering of under-resourced language like Amharic. The complexity here is that all the natural language processing approaches that have been applied to most languages, is not valid for applying to Amharic language directly. The text needs much manipulation and pre-processing before applying these methods. There are few works done on the Amharic language sentiment-based social media content filtering system. Since the Amharic language is, a language with its own script has been the official language of Ethiopia since then and still holds this position.

Our research is concerned about Amharic sentiment analysis for the social media content filtering system. Though Amharic plays several roles, it is considered one of the less-resourced languages. This is because it lacks basic tools and resources like sizable, clean and properly tagged corpora for carrying out natural language processing research and applications[58], [59][60] and this is because Amharic is morphologically rich and the boundary of the syntactic word in an orthographic word is unclear in most cases[60]. Even if it is possible to use some techniques of collecting a big corpus from the web, the basic natural language tasks like POS tagging or Normalization, and semantic analysis would be challenging if we base our analysis on orthographic words[59]. These activities suffered

from the lack of Amharic speech and Amharic text corpus suited for the development of information filtering of social media posts.

Our work is fundamentally different from the other works since we have proposed using different supervised machine learning algorithms with word2vec for sentiment analysis. It is hard to know right at the start which algorithm will work best for Amharic social media texts. It is usually best to work iteratively. Amongst the ML algorithms we identified as potential good approaches, we threw our data into them, run them all in either parallel or serial, and at the end, we evaluated the performance of the algorithms to select the best one(s). We have also employed an information filtering system based on the predicted tag of sentiment classifier. To the best of our knowledge, this is the first information filtering system to categorize Amharic posts according to their sentiments into politically, socially and religiously offensive contents and then filters offensive messages from social media user walls on the basis of message content.

Chapter 4: The proposed social media information filtering model for Amharic

4.1 Introduction

In this chapter, we will discuss the overall design of our proposed social media information filtering model, the main components and the interaction between them. First, we will illustrate the general overview of the proposed Filtering system architecture in Figure 4-1, then we will describe how data preprocessing is performed, how supervised machine learning algorithms classified document-level sentiment components along with its sub-components from the perspective of the Models flow of operations in the form of processes and how to implement each of the components will be defined.

4.2 Architecture of filtering model

The design of the proposed architecture has five major components; Amharic sentence tokenizer, pre-processor/Normalization step, Morphological analyzer, Feature selector, and Sentiment Classifier. We will begin by showing the architectural design of the Filter then we will be discussing the main components of the Filter along with the resources needed for each module and in what way they will interact with each other.

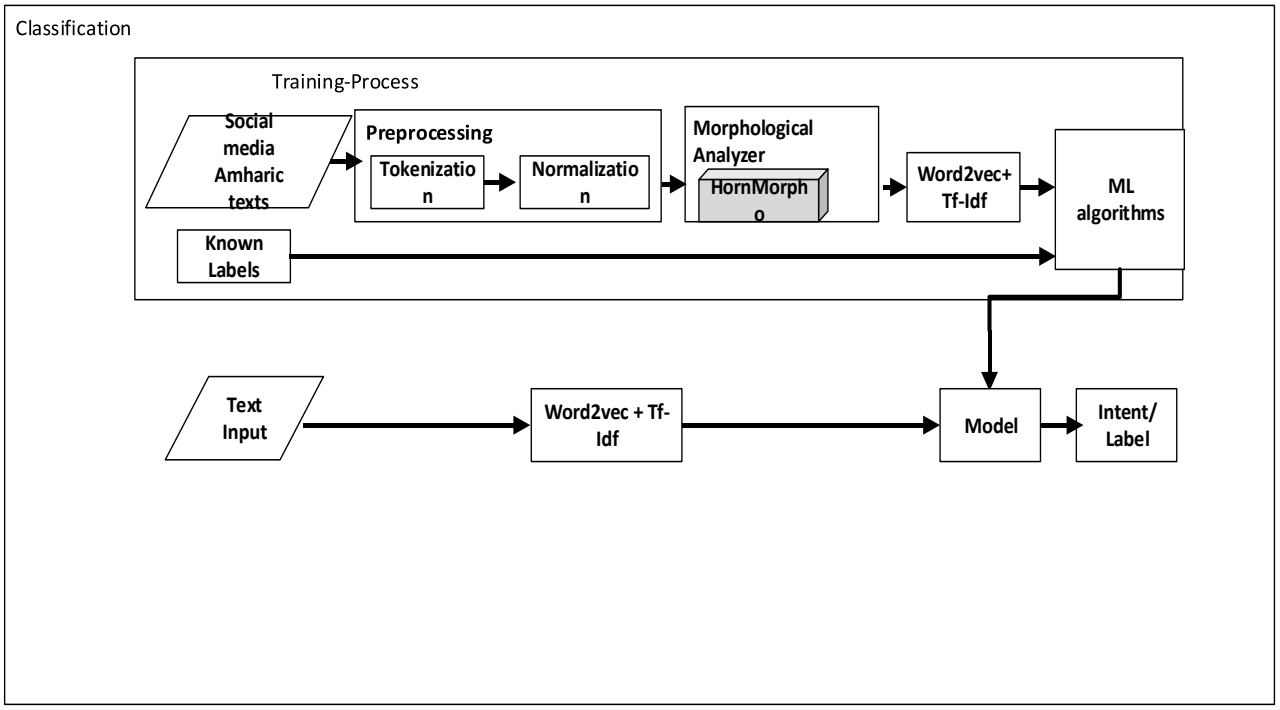


Figure4-1 Sentiment Analysis Architecture for social media Amharic texts

4.3. Components of information filtering model

4.3.1 Normalization/ Preprocessing

The first pre-processing step that was carried out on the collected data was filtering out non-Amharic content. This is particularly important when dealing with data from the "Web" or "social media". The preprocessing of input texts from social media involves the conversion of raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set. At first, all the words in the data are tokenized to their individual words and as the next step, all the stop words are removed from the training data. Stop words are the commonly occurring words in the language that do not contribute any meaning that would benefit the text classification process. Removing numbers, accent marks, whitespaces and other diacritics are main preprocessing steps that are included during the design and development of our model. The normalization of Amharic characters is also carried out in the preprocessing phase. The normalization issue that we gave attention is replacement of homophonic characters with their common forms; ሐ and ኀ are replaced with ሀ, ሠ with ሰ, ዐ with አ, and ፀ

with ጸ, normalization of punctuation marks, different styles of punctuation marks exist in texts from social media. For instance, double quotation marks, "።, ፣ « or » "are available. Thus, the normalization of punctuation marks is a nontrivial matter. Text normalization includes[61]: Removing numbers, punctuations, accent marks and other diacritics, and removing stop words.

4.3.1.1. Tokenization

We took Amharic text and break it up into its individual words. These tokens are then used as the input for normalization, feature selection, and sentiment classification. Example pseudo-code is described in Algorithm 4.1

Input (Si)

Output (Tokens)

Begin

Step1: Collect Input sentences (SEi) where i=1, 2, 3…. n;

Step2: For each input SEi; split Word (EWi) = SEi; // apply split() word process for all documents i=1, 2, 3…n in //

Step3: For each EWi; Stop Word (SWi) =EWi; // apply Stop word elimination process to remove all stop words like ["የ", "በ", "ለ", "ስለ", "ላይ","ነው","ነበር", "ብቻ", "እኔ", "እሷ" , " እሱ", "እነርሱ"]

Step4: Tokens (Ti) will be passed to Normalizer.

End

Algorithm 4-1Tokenization Algorithm

Removing stop words

"Stop words" are the most common words in a language like prefixes "የ", "በ", "ለ", "ስለ", "ላይ" and suffixes "ን", "ም", "ና", "ዎች". These words do not carry important meaning and are usually removed from texts. These words are commonly occurring words in the language that do not contribute any meaning that would benefit the text classification process. It is not possible to remove Amharic stop words using Natural Language Toolkit (NLTK), we designed our own stop words because NLTK doesn't recognize Amharic stop words.

Identification of stop words enables language users to retrieve information fast and makes the language more powerful for information processing. For example, in addition to suffixes and prefixes, Amharic words like "ይጠበቃል" (expected) "አስታውቀዋል" (declared), ብለዋል (said), "ነው" (is), "ነበር" (was), etc, are also considered as stop words[62]. some examples of stop words: ["ነው","ነበር", "ብቻ", "እኔ", "እሷ"," እሱ", "እነርሱ"]. Algorithm 4.2 illustrates pseudo-code of preprocessing steps, stop-word removal, punctuation, and other diacritics.

```
INPUT: string characters of words and numbers [0-9], stop words (S), diacritics (symbols with Amharic alphabets)
OUTPUT: string of alphabets
   Procedure:
       for r in words:
            if r in stop words, punctuations and characters, ascii and numbers, :
          Re.sub(S,[0-9],[!"#$%&'()*+,-:;<=>?@[\]^_`{|}~]
End
```

Algorithm 4-0-2 Algorithm to remove stop-words, punctuation, and numbers

4.3.2 Morphological Analyzer

In our system, the morphological analyzer is used to analyze words in a given training data to identify root or stem form and component morphemes so that required features and root or stem word are extracted to build a classifier. We adopted HornMorpho, which is an open-source, morphological analyzer for Amharic, Afan Oromo, and Tigrigna languages. Mikael gasser's morphological analyzer basically finds all possible root forms and suffixes of a given word. By using HornMorpho we analyzed the morphology of an input word; the analyzer read the inflected surface form of each word in a text and provided its lexical form. We applied Morphological analysis for the completion and also to improve the performance of the classifier, as the number of words in the training dataset can be represented by a single root word. If the word is taken as it is, the probability of occurrence on the training set may be zero for most of the words. This lowers the performance of the system.

```
Input: word
Take a word from the preprocessed texts or user input sentence
```

For each word in each sentence:

    Call HornMorpho

    Select the root word from the output of HornMorpho

    Return the root word

End for

Output: root word

Algorithm 4-3 Morphological analyzer algorithm

4.3.3 Feature extraction from text

The main step in a machine learning text classifier is to transform the text into a numerical representation, usually a vector. The process of automatic feature extraction uses preprocessed text as an input. The input is large labeled data, and this data is tokenized, preprocessed and features are extracted before it's used for training. This stage is where the dataset is transformed into a vector of numbers. The output from this stage is a fixed-size vector representation for each word. TF-IDF and Word2vec with skip-gram model is used for generating word vectors. In the case of TF-IDF, each component of the vector represents the frequency of a word or expression in a predefined dataset. There are certain words that are considered slang or abusive words but are so common that people use them a lot in their daily posts. So, if there is a post containing these words, it is intended as offensive. To draw the line between whether a slang/abusive word is intended as offensive or not, we tried the TFIDF approach since it allocates a low weightage to such words. So, instead of the Bag of Words model where we have a count of how many times that word occurred in the post, this will have a TFIDF weight instead. After having this vector representation of the text using tf-idf and word2vec, let's see how word2vec with SVM works. Firstly, word2vec prunes the words whose occurrence frequency in the corpus is less than five before training and features selected based on part-of-speech from hornMorpho, are utilized to select valuable features from word vectors. This method selects valuable features according to the part-of-speech of words. The different choices of part-of-speech can lead to different results[2]. In this method, after part-of-speech selection, we keep verbs, and nouns, which are two of the most common words in the documents. And then word2vec generates vector representation of the corpus and creates a model file that contains a list of frequent words and their

corresponding vector representations with one dimension. By taking this feature vectors it feeds SVM with data. After the training data is fed, the model building process starts to form a model considering the features.

Input:

D: {d1, d2……dn},the documents of the training set

S: {s1, s2,……sn},Input sentence S contains n words

T: {t1, t2,……tn},the unique terms in all documents

R: {r1, r2,……rn}, input root words R containing n root words

Output: Selected features ready for model training

Word vectors of root word R

Procedure:

Segment D into sentences S1, S2, . . . ;

Word vector ← words set preprocessed;

Document vector ← If keyword Wi appear in sentence Si ;

Cosine similarity array ← word vector;

For i in range (iteration), set the number of iterations, epoch, batch-size:

Sort W S(Vi);

Choose top T key words after sorting;

End

End

Algorithm 4-4 Feature selection Algorithm

The main steps of the algorithm 4.3 are as follows.

1) Divide the given Document D into complete sentences, i.e. D = {S1, S2, · · ·, Sm}, where D is divided into m independent sentences.

2) For each sentence Si ∈ D, segment the sentences into individual words, and then perform the preprocessing and morphological analysis. Only reserve words with specified parts of speech, such as nouns, verbs, adjectives, so that Si = {wi,1, wi,2, · · ·, wi, n}, where wi,j are reserved keywords, and n is the number of keywords.

3) Construct a candidate key words graph G = (V, E), where V is a set of nodes that consists of the candidate keywords generated by step (2), and then use the co-occurrence relationship to construct an edge between any two points. Two edges exist between nodes only if their corresponding vocabulary co-occurs in a window of length K, where K represents the size of the window. That is, the number of the maximum co-occurrence words is K.

4) Calculate the weight of each node by iterating. We divided our dataset of 2000 paragraphs into batches of 16 then it took 125 iterations to complete 1 epoch. Since our dataset is small it more likely will benefit from more iteration.

5) Sort the importance of each node weight value, and then obtain the candidate keywords.

4.3.4 Sentiment Classifier

Sentiment classifier is the main part of our architecture. The classification process uses different machine learning algorithms to build a model. The input for the Sentiment classifier is the output of feature extractor which is a file containing the word vector of words from a given plain Amharic text. Apart from Artificial neural network classifiers used for sentiment classification by researchers, and a lot of classification algorithms available, depending on the application and nature of our available data set, we applied the classification with the help of four traditional ML algorithms, SVM, decision tree, logistic regression, and Naïve-Bayes. Since our research focus is on document-level sentiment analysis, the whole document is analyzed and we figured out the polarity of the whole document. For example, a post contains texts of only one issue about politics, the system calculates whether the whole post expresses an overall offensive or non-offensive opinion about politics. In our research, the document expresses opinion on a single entity. It is not applicable to multiple political, social, religious situation entities.

In the training step, we used our data to incrementally improve our model's ability to predict whether a given post is offensive or not. In this process of training the ML model, we

provided an ML algorithm (that is, the learning algorithm) with training data to learn from. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that we want to predict), and it outputs an ML model that captures these patterns. We used the ML model to predict new data for which we do not know the target. In this case, known offensive and non-offensive Amharic texts are used as the training data. When the classifier is trained accurately, it can be used to detect an unknown Amharic text.

Our model is designed to identify offensive and non-offensive words including insult, such as "አስጠሊታ", "ቆንጆ", "መንጋ", "አስቀያሚ", "ነፍጠኛ", etc. Our Sentiment classifier receives an input, a set of vector representation of texts (e.g., posts, comments or messages from social media) discussing a particular entity (e.g., massacre of ethnic group in certain parts of Ethiopia, making fun of famous people, attacking certain religion, etc) analyses an incoming message and tells whether the underlying sentiment is Politically-offensive, Socially-offensive, Religiously-offensive or Non-offensive. Furthermore, our model analyzes the different aspects of the category of social media posts, either offensive posts with politics, ethnicity, and religion.

Pseudo code TRAIN NAIVE BAYES(D, C)

returns log P(c) and log P(w|c)

for each class c ∈ C # Calculate P(c) terms

Ndoc = number of documents in D

Nc = number of documents from D in class c

logprior[c]← log Nc/ Ndoc

V←vocabulary of D

bigdoc[c]←append(d) for d ∈ D with class c

for each word w in V # Calculate P(w|c) terms

count(w,c)←# of occurrences of w in bigdoc[c]

loglikelihood[w,c]← log count(w, c) + 1/ $\sum w^j$ in V count (w^j , c) + 1

return log prior, loglikelihood, V

function TEST NAIVE BAYES(test doc, log prior, loglikelihood, C, V) returns best c

for each class c ∈ C

sum[c]← logprior[c]

for each position i in test doc

if word[i] ∈ V sum[c]←sum[c]+ loglikelihood[word[i],c]

return argmaxc sum[c]

Algorithm 4-5 Pseudo code of Naïve-Bayes algorithm

As we can see in algorithm 4.4, the Multinomial Naïve Bayes algorithm reads the training dataset and predicts the class of testing dataset using conditional probability. Because naive Bayes is a kind of classifier that uses the Bayes theorem, it predicts membership probabilities for each class such as the probability that a given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. Each event in classification represents the occurrence of a word in a document.

INPUT: k,m,q,C,Y and termination criterion

OUTPUT: optimal value for SVM parameters and classification accuracy

Begin

Initialize k solutions

Call SVM algorithm to evaluate k solutions

For i=1 to m do

Select S according to its weight

Sample selected S

Store newly generated solutions

Call SVM algorithm to evaluate newly generated solutions

End

T = Best(Sort s1,……sk+m, k)

End

End

4. Calculate the likelihood for each class;

5. Get the greatest likelihood;

END

Algorithm 4-6 Pseudo code of the SVM algorithm

Algorithm 4.5 shows how SVM finds the optimal value of parameters by using weight and calculates the likelihood of each category.

The algorithm creates a line or a hyperplane that separates the text into classes. At first, approximation what SVMs do is finds a separating line (or hyper plane) between data of our classes. It takes preprocessed Amharic text as an input and outputs a line that separates those classes. According to the SVM algorithm, we find the points closest to the line from all the classes. These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The hyper plane for which the margin is maximum is the optimal hyper plane.

INPUT: Training dataset T, Testing dataset Td, initial no of components, Attr(set of descriptive attributes)

Tree(T, Td, Attr)

OUTPUT: A class of testing dataset

Step:

1. Read the training dataset T
2. create a root node for the tree;

Place the best attribute of the dataset at the **root** of the tree.

Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

4. Calculate the likelihood for each class;
5. Get the greatest likelihood;

END

Algorithm 4-7 Pseudo code of Decision-tree algorithm

The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label. In algorithm 4.6 decision Tree Algorithm Pseudocode, Places the best attribute of the dataset at the root of the tree, then Splits the training set into subsets and it repeats on each subset until it finds leaf nodes in all the branches of the tree.

4.3.5 Filter

This component removes inappropriate or offensive information from social media using the results of the sentiment classifier. It first checks for the value of an output from sentiment classifier and do actions based on that.

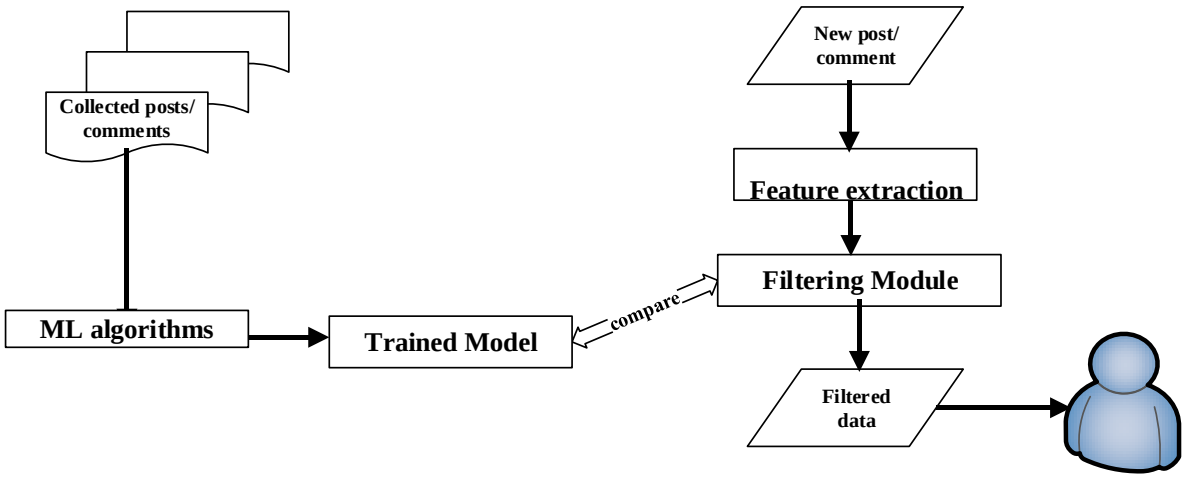


Figure 4.2 Information filtering Architecture

The architecture presented in Figure 4.2 shows information filtering procedure. Whenever new text comes, its feature is extracted to make the text understandeable by ML algorithms. The filtering module compares the text with trained model to determine its corresponding category. After determining the intent of the text, toxic content is hidden and the user can see filtered data.

```
if pipeline.predict(msg) == Pol_off:
      print("ይሄ ጽሁፍ ፖለቲካዊ ይዘት ስላለው እና ሌሎችን ሊያስከፋ ስለሚችል እንዳይታይ ተደርጓል")
elif pipeline.predict(msg) == Rel_off:
      print("ይሄ ጽሁፍ  የሌሎችን ሃይማኖት የሚነቅፍ ስለሆነ እንዳይታይ ተደርጓል")
elif pipeline.predict(msg) == Sol_off:
       print("ይሄ ጽሁፍ የሰውን ስብዕና የሚነካ ይዘት ያለው ጽሁፍ ስለሆነ እንዳይታይ ተደርጓል")
else:
         print(msg)
```

Algortihm 4-8 filtering algorithm

After categorizing text into one of the four categories, the filter will discard offensive words and hide it.

Our Filtering model primarily comprises of the following modules:

(1) Preprocessor: This phase involves the removal of stop words, numbers and characters. Analyzes incoming data and represents it according to the requirements of the Filtering module, Amharic texts on social media wall are collected and preprocessed in this module;

(2) Feature extractor: Feature extraction is a process of dimensionality reduction by which an initial set of raw data from social media is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set. Used for elimination of irrelevant variables to enhance the generalization performance

(3) Sentiment classifier: A sentiment classifier built using supervised machine learning algorithms is trained on given corpus on the list of offensives, non-offensive documents and Classifies the document into one of four categories

(4) Filtering and Hiding module: Takes the categories assigned by the classifier and then filters out inappropriate/ offensive classes of the different categories and determines the relevancy of the content. This module also learns and hides the offensive content of multiple categories. Each of these modules represents the prominent tasks of the information filtering model.

Chapter 5: Evaluation and implementation of the proposed model

In this chapter, the experimental results of the developed prototype system, setups/procedures, the evaluation parameters, and discussions are presented. The lack of readymade available resources such as Amharic Stop words, Amharic word net, data sources, and well-defined tools made conducting the experiment challenge.

For sake of research and experimental purposes, we collect data on four different categories namely; politically offensive, socially offensive, religiously offensive and Non-offensive. All the data about each category are collected from Facebook and blogs, methods for collecting data will be discussed in the following sections.

5.1 Amharic Sentiment Data Collection

As indicated in the Methodology section, we have considered the Facebook data domain as a major domain for conducting the experiments. The main challenge we faced during data collection from the social media data domain is due to the lack of readily available data written in Amharic language, datasets are built manually with help of an expert from the law department on data annotation. At first, we wrote python code to scrape Facebook data and we were collecting by scrapping but 90% of the data we were collecting was Non-offensive data, which created imbalanced data set, as we have four categories including Non-offensive.

Data source

All of the datasets (social media posts/comments) used for conducting the experiment from Facebook pages with the number of followers greater than 1000.

On the collected data, we performed text preprocessing including method to normalize character level mismatch such as ጸሀይ and ፀሐይ, Normalizing words with Labialized Amharic characters such as በልቱዋል or በልቱአል to በልቷል, replacing any existence of special character or punctuation to null, removing URLs, in general for our particular data set, our text cleaning step includes HTML decoding, removing stop words, punctuation, bad characters, and so on. As should be obvious, document-level analysis is carried out on the sentences from posts and comments. This approach is just a process of handpicking

paragraphs or sentences from Facebook pages sources with the goal of populating a dataset with offensive/inappropriate sentences. This manual approach is very time consuming and needs expert evaluation as offensiveness is subjective.

Our dataset was previously labeled to indicate offensiveness for each of the posts. The classes represent the criteria by which the accuracy of the experiments was analyzed. As we can see from the graph 5.1 our exploratory analysis showed an imbalance between the four sentiment classes. The analysis also showed a higher number of politically offensive instances for each of the posts. The serious imbalanced data set has an adverse effect on classification results. In the next section, we will define our classes.

Defining our classes

Defining what we mean by politically-offensive, Socially-offensive, Religiously-offensive, and Non-offensive is another challenge to tackle in order to perform accurate sentiment analysis. As in all classification problems, defining our categories is one of the most important parts of our problem. What we mean by politically-offensive, Socially-offensive, religiously-offensive, or Non-offensive does matter when we train sentiment analysis models. Since tagging data requires that tagging criteria be consistent, a good definition of the problem is a must.

Here are some ideas on what a Non-offensive tag contains:

1. Objective texts. So-called *objective* texts do not contain explicit sentiments, so we included those texts into the Non-offensive category.
2. Texts containing wishes. Some wishes like ከቃል በላይ ነው እረጅም እድሜ ይስጥልን, እንዲህ ነው የእናትነት ፍቅር, and የኢትዮጵያ እናቶች እጅግ በጣም ትሁት ናቸው ረጅም እድሜና ጤና ይስጥልን are generally Non-offensive.

Posts/comments are Politically-offensive when posts/comments are considered disrespectful or objectionable to a particular group of people. Online political bullying, shaming, supporters of one group calling supporters of others, racist, xenophobes and a variety of other labels are considered as politically-offensive content.

Religiously-offensive comments/posts consist of, harassment regarding the religious preference of a person, unwelcome and pervasive comments or behavior regarding

someone's religion that create a hostile or abusive situation, it would become a criminal offense to use threatening, abusive or insulting words or behavior.

Mocking, insulting, harassing children, adolescents and adults with disabilities, Spreading malicious rumors or gossip about famous people or insulting someone, Making negative comments about a person's appearance, lifestyle, family, or culture are categorized under Socially-offensive.

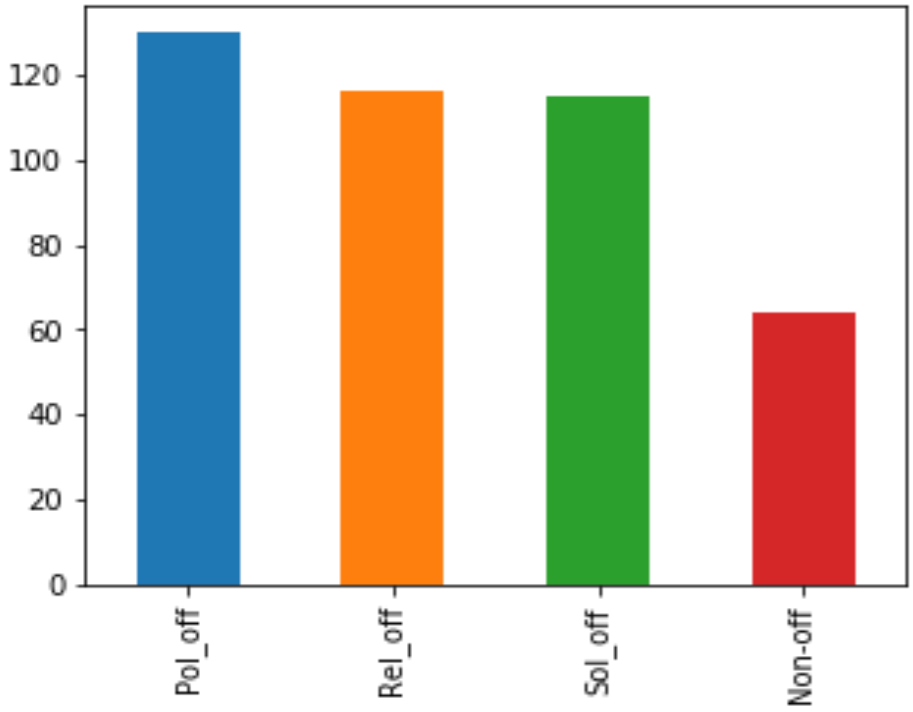


Figure 0-1 Data distribution graph

5.2 Manual classification

This activity is concerned with labeling the posts/comments for experimental purposes. All 2000 paragraphs are manually categorized by an independent individual from the domains into predefined categories: Non-offensive, Sol-offensive, Pol-offensive and Rel-offensive. The manually classified posts/comments helped us in crosschecking with the results obtained from our prototype system: sentiment analysis model for social media Amharic texts.

5.3 Using different classifiers

First, social media data were collected and preprocessed with basic natural language processing techniques like word tokenization, stop word removal and Normalization. The residual tokens were arranged as per their frequencies or occurrences in the whole documents set. Then different feature selection methods were utilized to pick out top n-ranked discriminating attributes for training the classifiers. The number of selected features (n) was varied from very small to very large. Four machine learning-based classifiers were applied to evaluate the effectiveness of different feature selection methods on the basis of the performance of the sentiment classification task.

After we have our features, we trained a classifier to try to predict the tag of a post. We started with a

- Naive Bayes classifier, which provides a nice baseline for this task. To make the vectorizer => transformer => classifier easier to work with, we used a Pipeline class that behaves like a compound classifier.
- SVM using the kernel trick transforms our data and then based on these transformations it finds an optimal boundary between the possible outputs. Simply put, it does some complex data transformations, and then figures out how to separate our data based on the labels or outputs we've defined. We chose from several different kernels when we create our support-vector classifier object (SVC). We applied SVM using a parameter tuning, to look for those changes which bring us the highest accuracy, because of its high performance on small as well as high dimensional data spaces.
- We have used multinomial logistic regression to predict our target classes, by calculating the probabilities for each target. Once the probabilities are calculated, we needed to transfer them into one hot encoding and use the cross-entropy methods in the training process for calculating the properly optimized weights. The inputs to the multinomial logistic regression are the features we have in the dataset. Politically-offensive, religiously-offensive, socially-offensive and Non-offensive parameters are our features. The expected output after training the multinomial logistic regression

classifier is the calculated weights. Later the calculated weights will be used for the prediction task.

- Decision Tree: When training a dataset to classify a variable, the idea of the Decision Tree is to divide the data into smaller datasets based on a certain feature value until the target variables all fall under one category. A DT algorithm splits the dataset based on the maximum information gain, Gini, Chi-square, entropy, etc.
- Word2vec: we used word2vec representation of each word of a text as input data for SVM for sentiment classification. We just passed them as input to our classifier just the same way as we would do with any sparse high-dimensional word representations where each feature is a binary indicator of a word (or a word counter, or tf-idf).

5.4 Experimentation result

Like the rest of classifiers, SVM doesn't take much training data to start providing accurate results. Although it took more computational resources, SVM achieved more accurate results. In short, SVM takes care of drawing a "line" or hyperplane that divides a space into two subspaces: one subspace that contains vectors that belong to a class and another subspace that contains vectors that do not belong to that class. Those vectors are representations of our training texts and a class is a tag we have tagged our texts with. word2vec is used as an input to SVM model for sentiment analysis task. we used word representations to initialize the features for our SVM model. We augmented our feature set with the sentence/document representation computed by (weighted) averaging of representations of words present in it.

We designed a prototype that accepts raw Amharic texts and displays a message about the input text if it is Politically-offensive, socially-offensive or religiously-offensive and it displays the original message if it is Non-offensive. The application developed shows the user interface and output of the classifier. It allows the user to insert Amharic text and checks the category of the input text. After the inputs are feed into the classifier, hiding offensive contents is done. Simple Chat application is shown in fig 5.1.

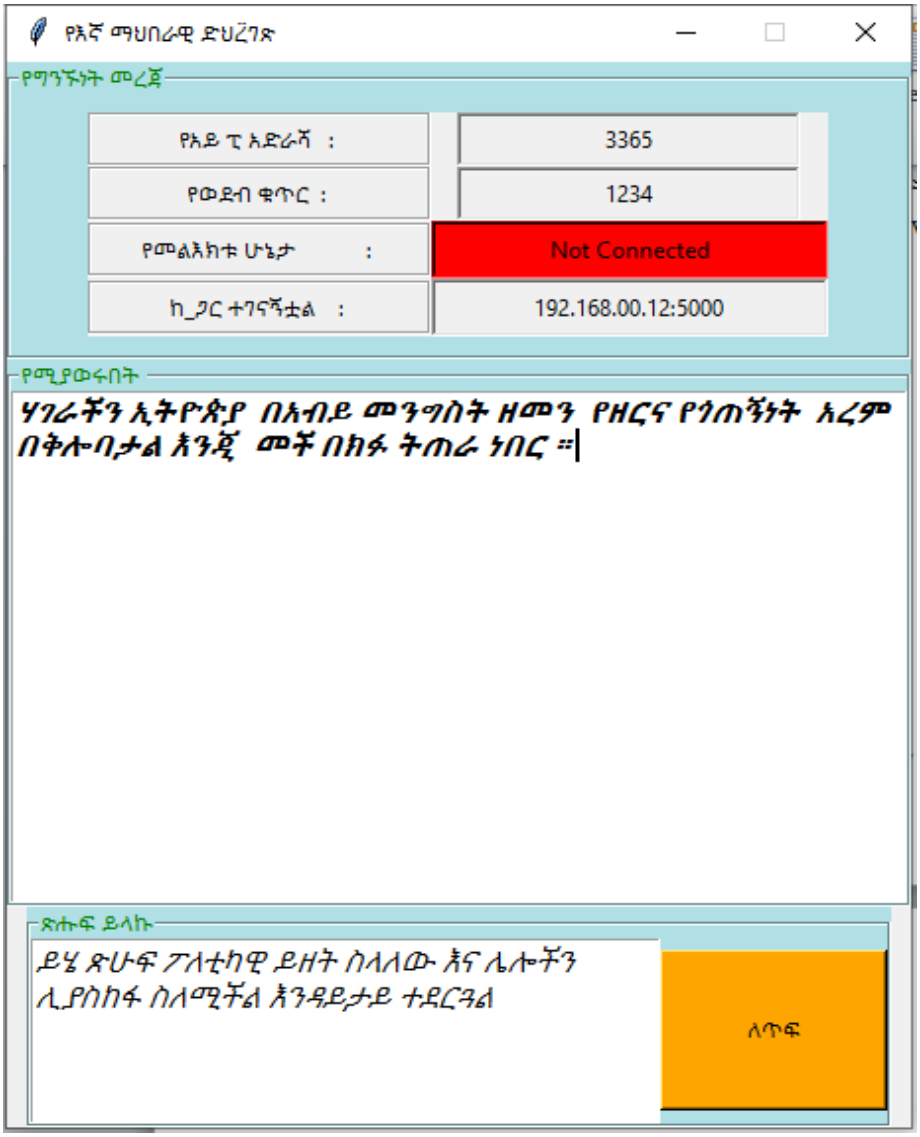


Figure 5-2 Simple chat application for social media information filtering

In order to test the proposed model, we have developed a prototype. Our task is to predict whether a given social media post is offensive/inappropriate or not based upon four attributes of the post/comment i.e. words contained within the sentence in accordance with the country's law, well known taboo expression in the society, words that has any textual or verbal practice that implicates issues of discrimination or violence against people in regard to their race, ethnicity, nationality, religion, sexual orientation, and gender identity and is kind of insult, provocation, abuse and aggression.

All social media posts/comments are used for conducting the experiments. Each post was classified by the system prototype according to the procedures described earlier and all the results were recorded. Then the results of different machine learning algorithms (SVM, Decision trees, logistic regression, and Naïve Bayes classifiers) were compared with the manually labeled classifications. As a result, the results obtained by each classifier are given below in table 5.1.

Table 5. 1 results obtained by each classifier

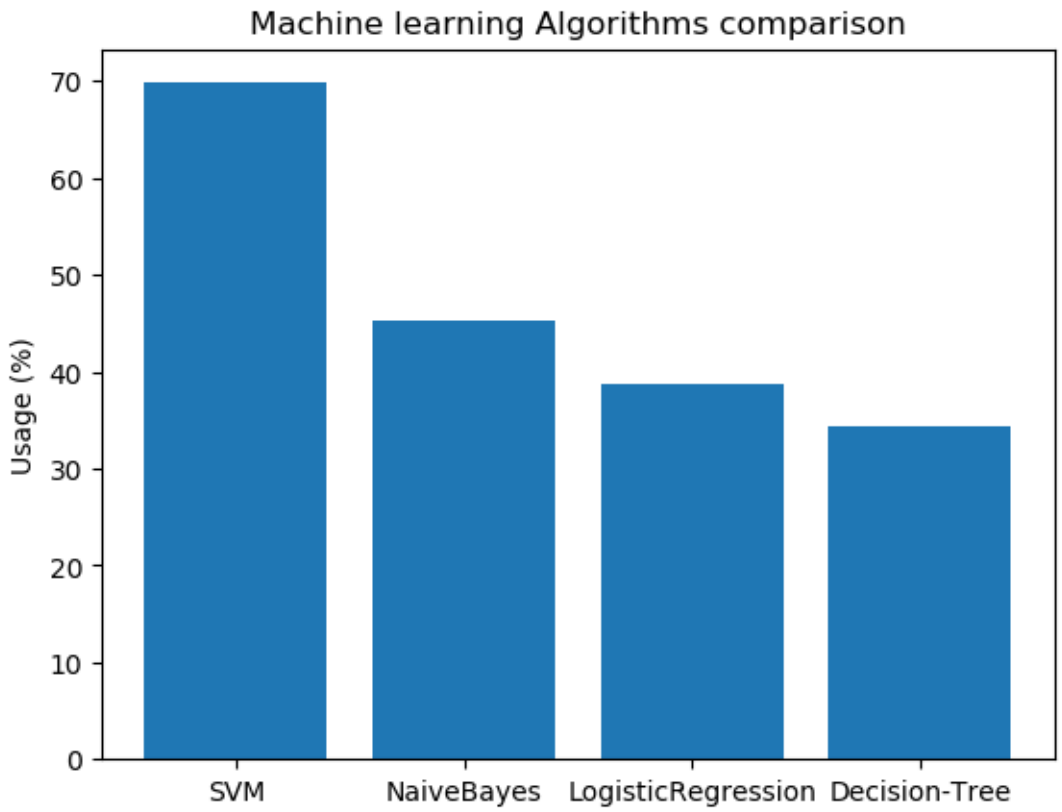
Model	Precision	Recall	F1-score	Class
SVM	**0.88**	**0.65**	**0.75**	**Non-off**
	0.60	**0.61**	**0.61**	**Pol-off**
	0.66	**0.59**	**0.62**	**Rel-off**
	0.55	**0.74**	**0.63**	**Sol-off**
Naive-Bayes	**0.65**	**0.48**	**0.55**	**Non-off**
	0.45	**0.60**	**0.51**	**Pol-off**
	0.45	**0.48**	**0.46**	**Rel-off**
	0.50	**0.45**	**0.48**	**Sol-off**
Decision-Tree	**0.49**	**0.45**	**0.47**	**Non-off**
	0.48	**0.27**	**0.35**	**Pol-off**
	0.49	**0.40**	**0.44**	**Rel-off**
	0.40	**0.64**	**0.49**	**Sol-off**
Logistic-Regression	**0.72**	**0.70**	**0.71**	**Non-off**
	0.62	**0.60**	**0.61**	**Pol-off**
	0.57	**0.55**	**0.56**	**Rel-off**
	0.56	**0.62**	**0.59**	**Sol-off**

As shown in Table 5.1, the classification report details how the algorithms performed for each sentiment class. We can see that SVM is easily identifying Non-off classes (f1-score of 75) and Sol-off (f1-score of 63) with an accuracy score of 65%.

The NB model obtained an overall classification accuracy of 50% on average. Precision and F-score are varied from 65% to 55%. Naïve-Bayes performs very poorly when features are highly correlated which is limitations of NB classifier because our dataset contains highly correlated features.

Since Logistic regression is much more robust to correlated features; our features f1, f2, f3, and f4 are perfectly correlated, it will simply assign quarter the weight to w1, quarter the weight to w2, quarter the weight to w3 and a half to w4. Thus we have many correlated features in our data; logistic regression assigned a more accurate probability than the rest of the classifiers. Thus, the logistic regression accuracy score is the second-highest score next to SVM.

Because our dataset is small, and decision Tree creates a training model that can be used to predict class or value of target variables by **learning decision rules** inferred from prior data (training data), the size of our corpus is the reason why decision tree classifier scored lowest accuracy.



5.5 Discussion of the results

As shown above, the experiment is done with different machine learning algorithms, experimental setups and has shown us promising results. These different experimental setups are the reasons for the variations of experimental results.

Table 5.1 shows a comparison between the training of Naive Bayes, logistic regression, decision tree and SVM classifier over our dataset. From the learning curves, we can clearly notice that NB classifier will not benefit from adding more training data; since it converges quickly even with the small social media dataset. Whereas in the case of SVM classifiers, there is room to further improve our results by increasing the size of data size and the system prototype performs relatively well with SVM classifiers than with the rest three classifiers mentioned above. This is mainly due to the complex nature of Amharic texts, and small datasets. The detailed performances of all classifiers are reported in Table 5.1. For the purpose of method comparison, we follow the conventions in reporting our classifiers results. We use the measures recall, precision, F-measure, and macro-accuracy.

When writing Amharic comments/posts in our prototype for prediction, for example: in a post.

“አይ የሚኒልክ ሰፋሪ ማጥፊያቹ ደርሷል መሰለኝ ኡኡታ አበዛቹ”,

SVM: Pol-off the right expressed sentiment.

Furthermore, SVM works extremely well (even better than logistic regression or Naïve Baye s) on small datasets or short documents. whereas, naive Bayes is easy to implement and very fast to train.

Chapter 6: Conclusion and Recommendation

6.1. Conclusion

As inappropriate/offensive content on social media continue to be a societal problem, the need for automatic detection systems becomes more apparent. In this report, we proposed a solution to the detection of offensive language on Facebook through machine learning using word2vec and TF IDF values. We performed a comparative analysis of Logistic Regression, Naive Bayes, Decision Tree, and Support vector machine on various sets of feature values and model parameters. The results showed that the support vector machine performs comparatively better with the word2vec approach.

In this paper, we proposed an Amharic text sentiment-based social media content filtering approach for the filtering or hiding of offensive/inappropriate content. Our methodology contained several classifiers implemented using the sci-kit-learn machine-learning library to predict the sentiment of Facebook posts. These classifiers include Naïve Bayes, Support Vector Machines, Logistic Regression, and decision tree. Each classifier was initiated and then the fit method was used to train the classifiers on the scaled oversampled training data. The predict method was then used to classify the test dataset, and the accuracy score method from the sci-kit learn's metrics library was used to determine the effectiveness of the test. Logistic Regression and SVM classifiers performed best as reflected in Table 5.1. We focused on cleaning the posts that we have in the collections and this pre-processing step gave us improved results with all the classifiers. This proved that cleaning the text after understanding the domain does result in better F1 scores for all the classifiers, as shown through experimentation. The classifier that we have developed based on Word2Vec with SVM gives us very good accuracy (0.72) when applied to a sample of 4 classes.

The results show that the filtering techniques based on sentiment analysis and combined with machine learning classification, perform well enough to be worth studying further as a research topic. This work focused on Facebook data collected posts/comments for the task of sentiment analysis. Evaluation is done to calculate the performance of the classifiers. Even if the results are good for sentiment classification, our research is far from perfect. We

have a lot of work ahead of us. We need to have a large dataset and evaluate the performance of machine learning algorithms.

Overall, we conclude that there is no single technique in the text classification domain that would contribute to the accuracy dramatically. Each component in our classification pipeline contributes to improving the accuracy and must be adjusted to fit the problem domain. While the accuracy of the classifier is vital, it should not come at the cost of large performance degradation. The use of platform tools to keep an eye on runtime performance and optimize it from time to time is equally important.

6.2. Recommendations and future work

To fully explore the problem of classifying posts/comments as well as the results of our experiment we have identified the following future work:

- To get a better feature selection for the classifier we propose using a larger corpus such as a larger social media dataset, entire datasets, to train the Word2Vec model. This will give more documents for the model to build the syntactic and semantic relationships for the words. Currently, by training on the training set itself, the model cannot generate any word vectors for terms not present in the training set. This is a big limitation of the system. It will be interesting to see which of these bigger corpora provides us the best classification accuracy empirically.
- We hope to optimize the accuracy of the sentiment analysis by potentially adopting deep neural networks and by analyzing other approaches that other researchers used. Also, the expansion of this work should explore more features of Word2Vec and take advantage of its vast modules to obtain more accurate results.
- As additional classes are added we expect the accuracy of the classifier to decrease as it has to distinguish across more classes. We need to investigate whether the accuracy actually decreases in such cases, and then experiment with approaches to classify large numbers of classes without the accuracy penalty, such as breaking up the multiclass classifier into one classifier for each broad category or using ensemble methods for voting based prediction across multiple classifiers.
- This research only focused on text content on social media; however, there is more content on social media which is offensive and inappropriate, like image, video,

image, and text in one. So, the research should be broadened to other social media content types as well in the future.

- In this research, due to time limitations, data collected manually is not enough to test it using Neural Networks. For the future, by collecting enough amounts of data, the Amharic text filtering system can be implemented using neural networks and the performance and accuracy of the content filtering system can be improved.

6.3. Contribution

- The main contribution of this work is the performance investigation of different sentiment classification and machine learning methods in terms of accuracy for Amharic text.
- We consider designing our own Amharic dataset as a contribution to this research.
- Designing Amharic text filtering architecture
- Designing a Prototype for our model.
- Applying word2vec and supervised machine learning algorithm to optimize the accuracy of Amharic texts and compare the performance.

In this work, our focus was on sentiment analysis in Amharic, which is an agglutinative language that makes the sentiment analysis a relatively more complicated problem. The contributions of our work are the following. We proposed a model for sentiment analysis in text written in Amharic, especially focusing on informal and noisy text found in social media. We customized Horn-Morpho a morphological analyzer Python program for analyzing and generating words in Amharic, Tigrinya, and Oromo[63], to make it work with Amharic text. We evaluated the performance of our model using a small dataset containing social media texts that are written in Amharic. The experimental results indicate a fairly good accuracy in predicting the polarity of Amharic text.

References

[1] W. Li, J. E. O'Brien, S. M. Snyder, and M. O. Howard, "Characteristics of Internet Addiction/Pathological Internet Use in U.S. university students: A qualitative-method investigation," *PLoS One*, 2015.

[2] B. Liu, "LIBRO_SentimentAnalysis-and-OpinionMining," no. May, 2012.

[3] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," pp. 1320–1326.

[4] B. Liu, "Sentiment analysis and subjectivity," in *Handbook of Natural Language Processing, Second Edition*, 2010.

[5] H. Ghorbel and D. Jacot, "Sentiment analysis of french movie reviews," in *Studies in Computational Intelligence*, 2011.

[6] W. van Atteveldt, J. Kleinnijenhuis, N. Ruigrok, and S. Schlobach, "Good news or bad news? Conducting sentiment analysis on Dutch text to distinguish between positive and negative relations," *J. Inf. Technol. Polit.*, vol. 5, no. 1, pp. 73–94, 2008.

[7] F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, and T. By, "Sentiment analysis on social media," *Proc. 2012 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2012*, no. June 2014, pp. 919–926, 2012.

[8] Wondwossen Mulugeta, "A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts," pp. 80–87.

[9] A. T. S. To, "SCHOOL OF GRADUATE STUDIES SENTIMENT MINING MODEL FOR OPINIONATED By : Selama Gebremeskel FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES SENTIMENT MINING MODEL FOR OPINIONATED AMHARIC TEXTS," p. Angeles, L., Advocacy, S., Location, O. (2002)., 2010.

[10] Z. Mossie and J. Wang, "S OCIAL N ETWORK H ATE S PEECH," pp. 41–55, 2018.

[11] K. Ganagavalli, A. Mangayarkarasi, T. Nandhinisri, and E. Nandhini, "Sentiment

analysis of twitter data using machine learning algorithm," *J. Comput. Theor. Nanosci.*, vol. 15, no. 5, pp. 1644–1648, 2018.

[12] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," in *Computational Intelligence*, 2006.

[13] V. M. and A. P. Francesco Colace, Massimo De Santo, Luca Greco, *Sentiment Analysis and Ontology Engineering An Environment of Computational*. 2016.

[14] T. H. E. Prosecution, O. F. Criminal, C. Committed, O. Social, and B. B. Zemedkun, "School of law," 2019.

[15] A. Brown, "What is hate speech? Part 1: The Myth of Hate," *Law Philos.*, vol. 36, no. 4, pp. 419–468, 2017.

[16] M. Y. Anis, L. S. Anggreni, and M. S. Yuliarti, "Hate Speech in Arabic Newspaper Cyber Law - Case Study In Al-Jazeera.Net Daily Newspaper," no. October 2018, pp. 615–620, 2018.

[17] K. M. Yilma, F. News, and C. Information, "Fake News and Its Discontent in Ethiopia," vol. 1, no. June, pp. 98–114, 2017.

[18] L. Gordon, S. Sullivan, S. Mittal, and K. Stone, "Ethiopia's Anti -Terrorism Law: A Tool for Dissent," pp. 1–22, 2015.

[19] S. Philosophy and N. Sep, "Review Reviewed Work (s): Offense to Others by Joel Feinberg Review by : Bernard Gert Published by : International Phenomenological Society," vol. 48, no. 1, pp. 147–153, 2020.

[20] V. K. T. Karthikeyan, "Web Content Filtering Techniques : A Survey," vol. 5, no. 03, pp. 203–208, 2014.

[21] U. Hanani, B. Shapira, and P. Shoval, "Information filtering: Overview of issues, research and systems," *User Model. User-adapt. Interact.*, 2001.

[22] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*. 2010.

[23] A. L. Kavanaugh *et al.*, "Social media use by government: From the routine to the

critical," *Gov. Inf. Q.*, 2012.

[24] H. Liang and J. J. H. Zhu, "Big Data, Collection of (Social Media, Harvesting)," in *The International Encyclopedia of Communication Research Methods*, 2017.

[25] J. Bright, *The use of social media for research and analysis : a feasibility study*, no. December. 2014.

[26] S. Walker, "The Complexity of Collecting Digital and Social Media Data in Ephemeral Contexts," 2017.

[27] S. Lomborg and A. Bechmann, "Using APIs for Data Collection on Social Media," *Inf. Soc.*, 2014.

[28] M. N. Injadat, F. Salo, and A. B. Nassif, "Data mining techniques in social media: A survey," *Neurocomputing*, 2016.

[29] M. Ikonomakis, "Text Classification Using Machine Learning Techniques," vol. 4, no. 8, pp. 966–974, 2005.

[30] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources UXW Ya ` cb d e ¦ f gihqpsrutVv8wyx ¨' "˜"" • ˆ ˆ ‰ • UXW Ya ` cb ' x ¨€ –□ i *f*,□,,, „ □,†" Y ˆ ˆ ‰‡—."

[31] B. M. and V. B., "Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis," *Int. J. Comput. Appl.*, vol. 146, no. 13, pp. 26–30, 2016.

[32] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "Training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 1992.

[33] S. Redhu, S. Srivastava, B. Bansal, and G. Gupta, "Sentiment Analysis Using Text Mining : A Review," vol. 4, no. 2, pp. 49–53, 2018.

[34] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics*. 2004.

[35] S. K. Murthy, "Automatic construction of decision trees from data: A multi-

disciplinary survey," *Data Min. Knowl. Discov.*, 1998.

[36] C. S. Diaries, "M A C H I N E L E A R N I N G F O R S E N T I M E N T A N A LY S I S O F," 2017.

[37] R. N. Waykole and A. D. Thakare, "International Journal of Advance Engineering and Research A REVIEW OF FEATURE EXTRACTION METHODS FOR TEXT," pp. 351–354, 2018.

[38] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," no. 2, pp. 1–5, 2014.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[40] F. Heimerl and M. Gleicher, "Interactive Analysis of Word Vector Embeddings," *Comput. Graph. Forum*, 2018.

[41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations ofwords and phrases and their compositionality," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2013.

[42] C. Lucchese, C. I. Muntean, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini, "Recommender systems," in *Mining User Generated Content*, 2014.

[43] M. Morita and Y. Shinoda, "Information filtering based on user behavior analysis and best match text retrieval," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 1994.

[44] S. Khurshid, S. Khan, and S. Bashir, "Text-Based Intelligent Content Filtering on Social Platforms," in *Proceedings - 12th International Conference on Frontiers of Information Technology, FIT 2014*, 2015.

[45] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE : Synthetic Minority Over-sampling Technique SMOTE : Synthetic Minority Over-

sampling Technique," no. February 2017, 2002.

[46] H. Zhang, Z. Yu, M. Xu, and Y. Shi, "Feature-level sentiment analysis for Chinese product reviews," in *ICCRD2011 - 2011 3rd International Conference on Computer Research and Development*, 2011.

[47] M. Vanetti, E. Binaghi, E. Ferrari, B. Carminati, and M. Carullo, "A System to Filter Unwanted Messages from OSN User Walls," vol. 25, no. 2, pp. 285–297, 2013.

[48] V. Subramaniyaswamy, R. Logesh, V. Vijayakumar, and V. Indragandhi, "Automated Message Filtering System in Online Social Network," *Procedia - Procedia Comput. Sci.*, vol. 50, pp. 466–475, 2015.

[49] A. S. Kumar, "A System for Filtering Unwanted Messages from On-line Social Network (OSN) User Walls Victimization Machine Learning Techniques," vol. 8491, pp. 248–250, 2014.

[50] V. N. Mandhala, D. Bhattacharyya, and T. Kim, "Eliminating Unwanted Messages in SNS using Decision Tree," *Int. J. Database Theory Appl.*, vol. 7, no. 3, pp. 121–130, 2014.

[51] S. B. Kim, H. C. Rim, D. S. Yook, and H. S. Lim, "Effective methods for improving naive Bayes text classifiers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002.

[52] J. G. Shanahan and N. Roma, "Improving SVM text classification performance through threshold adjustment," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2003.

[53] D. E. Johnson, F. J. Oles, T. Zhang, and T. Goetz, "A decision-tree-based symbolic rule induction system for text categorization," *IBM Syst. J.*, 2002.

[54] B. Pang and L. Lee, "Sentiment analysis using subjectivity summarization," 2004.

[55] C. Scheible and H. Sch, "Bootstrapping Sentiment Labels For Unannotated Documents With Polarity PageRank," no. 1997, pp. 1230–1234, 2002.

[56] A. Hassan, Q. Diao, D. Radev, A. Arbor, I. Corporation, and S. Clara, “Improved Nearest Neighbor Methods For Text Classification,” pp. 1–22, 2011.

[57] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, “Detecting spammers on social networks,” *Neurocomputing*, 2015.

[58] M. M. W. B, L. Besacier, and M. Meshesha, “A Corpus for Amharic-English Speech Translation : The Case of Tourism Domain A Corpus for Amharic-English Speech Translation : The Case of Tourism Domain,” no. July, 2018.

[59] W. Mulugeta and M. Gasser, “Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming,” no. May 2012, 2014.

[60] A. Nürnberger, “Contemporary Amharic Corpus : Automatically Morpho-Syntactically Tagged Contemporary Amharic Corpus : Automatically Morpho-Syntactically Tagged Amharic Corpus,” no. October, 2018.

[61] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Microsoft Word - Abdelouahab Moussaoui.doc - data-preprocessing-for-supervised-leaning,” vol. 1, no. 12, pp. 4091–4096, 2007.

[62] S. G. Miretie, “Automatic Generation of Stopwords in the Amharic Text Automatic Generation of Stopwords in the Amharic Text,” no. May, 2018.

[63] M. Gasser, “HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya,” *Conf. Hum. Lang. Technol. Dev.*, pp. 94–99, 2011.

Appendix 1
Machine learning Classifiers-Support Vector Machine

Appendix 1

```
pipeline.score(x_train,Y_train)
from sklearn.metrics import accuracy_score
Y_preds = pipeline.predict(x_test)
print('accuracy %s' % accuracy_score(Y_preds, Y_test))
```

```
accuracy 0.6079136690647482
```

```
from sklearn.metrics import classification_report
from sklearn.metrics import precision_score, recall_score
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import f1_score
pipeline.fit(x_train, Y_train)
print(classification_report(Y_test, pipeline.predict(x_test), digits=4))
```

```
              precision    recall  f1-score   support

     Non-off     0.7660    0.6000    0.6729        60
     Pol_off     0.5976    0.6282    0.6125        78
     Rel_off     0.6167    0.6066    0.6116        61
     Sol_off     0.5281    0.5949    0.5595        79

    accuracy                         0.6079       278
   macro avg     0.6271    0.6074    0.6141       278
weighted avg     0.6184    0.6079    0.6103       278
```

```
pipeline.predict(["አዲሱ አመት የሰላም የአንድነት  የመግባባት፣ እና የፍቅር ይሁንልን። መልካም በዓል ይሁንልን"])
```

```
array(['Non-off'], dtype='<U7')
```

```
pipeline.predict(["ከፈለግን ኦርቶዶክሶችን ከነቁልቦቿ ከነ ቤተክርስቲያኑ ከ ከሃገራችን አናጠፋለን። ባለፈው ሲጻፃ ላይ እንጻያረግነው"])
```

```
array(['Rel_off'], dtype='<U7')
```

```
pipeline.predict(["እኳንም አማራ ያልሆንኩ እንደ አንተ አይነቱ ነው የገዛ ወንድሙን የሚገለው"])
```

```
array(['Pol_off'], dtype='<U7')
```

```
pipeline.predict(["የዚህስ ባሠ ምን አይነቱ ጅል አንፎ ዝፍቅ ነው በአናታቹህ። ወይም አፉ ላይ ጻይፐር አሠሩለት።።"])
```

```
array(['Sol_off'], dtype='<U7')
```

```
from sklearn.externals import joblib
joblib.dump(clf, 'model.pkl')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15: DeprecationWarning: sklear
n.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directl
y from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled
models, you may need to re-serialize those models with scikit-learn 0.21+.
  warnings.warn(msg, category=DeprecationWarning)
```

```
['model.pkl']
```

```
clf = joblib.load('model.pkl')
```

In [20]:
```
1 pipeline.predict(["ከፈለግን ኦርቶዶክሶችን ከነቁልቶቹ ከነ ቡተኪርስቲያኑ ከ ከሃገራችን እናጠፋለን። ባለፈው ሲጻማ ላይ እንጸየረማከው"])
```

Out[20]:
```
array(['Rel_off'], dtype='<U7')
```

In [21]:
```
1 pipeline.predict(["እኒንም አማራ ያልሆንኩ እንደ አንተ አይነቱ ነው የዛ ወንድሙን የሚገለው"])
```

Out[21]:
```
array(['Pol_off'], dtype='<U7')
```

In [22]:
```
1 pipeline.predict(["የዚህስ ባሠ ምን አይነቱ ጅል አንጄ ዝፍቅ ነው በአናታቹህ። ወይም አፉ ላይ ጸይፐር አሠሩብት።"])
```

Out[22]:
```
array(['Sol_off'], dtype='<U7')
```

In [23]:
```
1 from sklearn.externals import joblib
2 joblib.dump(clf, 'model.pkl')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15: DeprecationWarning: sklear
n.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directl
y from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled
models, you may need to re-serialize those models with scikit-learn 0.21+.
  warnings.warn(msg, category=DeprecationWarning)
```

Out[23]:
```
['model.pkl']
```

In [24]:
```
1 clf = joblib.load('model.pkl')
```

Appendix 2

1]:
```
import pandas as pd
import numpy as np
np.random.seed(1000)
import gensim
import tensorflow as tf
tf.random.set_seed(1000)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\gensim\utils.py:1197: UserWarning: detected Windows; aliasing chunkize t
erial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

2]:
```
df = pd.read_excel(r"C:\Users\hiwot\Documents\Classification\category\sol_off.xlsx")
```

5]:
```
def get_mean(words, word2vec):
    vecs = []
    for word in words:
        if word in word2vec:
            vecs.append(word2vec[word])
#         else:
#             vecs.append(np.zeros(100))
    if len(vecs) == 0:
        return None
    else:
        vec = np.mean(np.vstack(vecs), axis=0)
        return vec / np.linalg.norm(vec)

```

Activate W
Go to Settings

[6]:
```
# columns Text Words Phrases Category
text = "Text"
words = "Words"
Phrases = "Phrases"
category = "Category"
```

[7]:
```
model_file = open('model-1.vec', encoding='utf-8')
line = model_file.readline()
n_words, vec_size  = line.split(' ')
vec_size = int(vec_size)
line = model_file.readline()
word2vec = {}
print("Vector Size", vec_size)
while line:
    line = line[:-1].split(' ')[:-1]
    word, vec = line[0], np.array([float(x) for x in line[1:]])
    if len(vec) != vec_size:
        print(word, vec_size, len(vec))
    word2vec[word] = vec
    line = model_file.readline()


```

```
Vector Size 100
```

```
:
def base_model():
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(4, input_dim=vec_size, activation='softmax', kernel_initializer="uniform"))

    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
model = base_model()
```

```
:
labels = list(set(df[category][:100]))
```

```
:
X = []
Y = []
for i in range(df[text].shape[0]):
    words = df[text][i].split(' ')
    vec = get_mean(words, word2vec)
    if vec is not None:
        label = labels.index(df[category][i])
        yl = [0]*len(labels)
        yl[label] = 1
        Y.append(yl)
        X.append(vec)

```

```
:
X = np.vstack(X)
Y = np.array(Y, dtype=np.int32)
```

Activate Windows
Go to Settings to activate Window

```
l4]:
indexes = np.arange(len(X), dtype=np.int32)
np.random.shuffle(indexes)
X = X[indexes]
Y = Y[indexes]
n_train = int(.8 * X.shape[0])
n_test = X.shape[0] - n_train
print(n_train, n_test)
```

```
1161 291
```

```
l5]:
x_train, y_train, x_test, y_test = X[:n_train], Y[:n_train], X[n_train:], Y[n_train:]
```

```
l6]:
def gen_data(X, Y, batch_size=32):
    indexes = np.arange(len(X))
    current = 0
    while True:

        bs = indexes[current:current + batch_size]
        batch_x = X[bs]
        batch_y = Y[bs]
        current += batch_size
        if current > len(X):
            np.random.shuffle(indexes)
            current = 0
        yield batch_x, batch_y

```

Activa
Go to Se

```
[17]:
batch_size = 16
steps_in_epoch = len(x_train) // batch_size
gen = gen_data(x_train, y_train, batch_size)
```

```
[18]:
model.fit_generator(gen, steps_per_epoch=steps_in_epoch, epochs=200, validation_data=(x_test, y_test))
```

```
Epoch 195/200
72/72 [==============================] - 1s 11ms/step - loss: 0.7467 - accuracy: 0.7048 - val_loss: 0.9109 - val_accuracy: 0.
6220
Epoch 196/200
72/72 [==============================] - 1s 11ms/step - loss: 0.7827 - accuracy: 0.6873 - val_loss: 0.9118 - val_accuracy: 0.
6151
Epoch 197/200
72/72 [==============================] - 1s 11ms/step - loss: 0.7826 - accuracy: 0.6978 - val_loss: 0.9137 - val_accuracy: 0.
6151
Epoch 198/200
72/72 [==============================] - 1s 11ms/step - loss: 0.7582 - accuracy: 0.6969 - val_loss: 0.9155 - val_accuracy: 0.
6186
Epoch 199/200
72/72 [==============================] - 1s 11ms/step - loss: 0.7686 - accuracy: 0.6952 - val_loss: 0.9141 - val_accuracy: 0.
6151
Epoch 200/200
72/72 [==============================] - 1s 11ms/step - loss: 0.7623 - accuracy: 0.6969 - val_loss: 0.9134 - val_accuracy: 0.
6151
```

Appendix 2