Offensive Language Detection for Low Resource Language using Zero-Shot Learning

By

Noman Iqbal

(Registration No: 00000277461)

Supervisor

Dr Shibli Nisar

A thesis submitted to the faculty of Electrical Engineering Department Military College of Signals, National University of Sciences and Technology, Rawalpindi as part of the requirements for the degree of MS in Electrical Engineering

Jul 2022

DEDICATION

I wholeheartedly dedicated my work

MY BELOVED PARENTS (Haji Iqbal [Late], Parveen Iqbal), MY UNCLE (Brig-Munawar Hussain [Rtd], My Wife [Ambreen Noman],

RESPECTABLE TEACHERS AND DEVOTED FRIENDS

For their love, endless support and encouragement.

I would also like to eminently dedicate this to myself for not giving up and showing up right till the end

ACKNOWLEDGEMENTS

Thanks to ALLAH ALMIGHTY for showering His blessings.

I'm privileged to offer my heartfelt gratitude to my Supervisor Dr. Shibli Nisar for his abutment as well as counsel throughout my studies. His guidance and prolific exhortations were the beacon to complete my thesis in a timely fashion.

I’m equally honored to express my hat tip to Mr. Muhammad Hammad Iqbal. Without his sincere and bounteous contribution, I would not be able to achieve the milestone.

I would like to wage exceptional appreciations to GEC committee members Dr. Abdul Wakeel, Dr. Mehmood Alam, and Dr. Mir Yasir Umair for their great support and cooperation.

I’m thankful to Dr. Adil Masood Siddiqui, Dr. Muhammad Imran and Dr. Waseem Iqbal also for their uplift motivation and collaboration.

Last but not the least, I’m pleased to express my gratitude to all the individuals who have duly encouraged me for MS studies.

DECLARATION

*I, Noman Iqbal declare that this thesis titled "**Offensive Language Detection for Low Resource Language using Zero-Shot Learning"** has not been submitted before for any degree application at NUST or any other educational Institutes. This synopsis is presented as a result of my own original research*

_____

Noman Iqbal
(00000277461/MSEE24)

COPYRIGHT NOTICE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

JOURNAL PUBLICATION

ABSTRACT

Detection of cyberbullying on social media platforms is becoming a vital challenge for researchers recently as it is at its peak. Therefore, research work, to tackle this issue, is carried out in numerous languages around the Globe. People use different types of Social media platforms, in their native languages, to express their points of view. And if they are to express their anger or frustration, besides positive views, they often use abusive or offensive wording in their native language. Although some languages have an automatic monitor and block offensive content detection systems but unfortunately limited to Resource-rich languages very rare for low-resourced languages. The main reason is the non-availability of datasets for native/local languages.

In recent years, unethical behavior in the cyber-environment has been revealed. The presence of offensive language on social media platforms and automatic detection of such language is becoming a major challenge in modern society. The complexity of natural language constructs makes this task even more challenging. Until now, most of the research has focused on resource-rich languages like English. Roman Punjabi and Punjabi are two scripts of writing the Punjabi language on social media

To the best of our knowledge, no or very little work has been done on our topic. But an increasing amount of attention by computational linguistic community has given to detect offensive language and hate speeches from several online social media applications like YouTube [4-6], Twitter [7-9], Facebook [10] and blogs [11-12], in resource rich languages.

Our inspiration is the “Automatic Detection of offensive Language for Urdu and Roman Urdu” by Muhammad at. el [17] and “Hate-Speech and Offensive Language Detection in Roman Urdu”, by Hammad Rizwan, Muhammad Haroon Shakeel2020.emnlp-main.197

Similarly, we have a huge community of Punjabi speakers in Pakistan, India and Bangladesh. Cyberbullying is at peak via social media between them. This is a critical

issue which need to be addressed. We have already started to create the dataset for the proposed solution likewise in other low resource languages.

This research work proposes a model for “Punjabi”, a very low resource language, which automatically detects offensive language/words present. To create a dataset for roman Punjabi, we select 100 thousand and 1000 comments/feedback separately from different social media platforms and then the dataset of 1000 comments was labeled as offensive and non-offensive manually. The proposed model ZSL is a machine learning problem in which a beginner detects the samples from classes that didn't make the cut viewed in exercise and forecasts the category toward which they belong. The observed/seen and non-observed/unobserved categories are combined using zero-shot approaches, which use auxiliary information to represent observable differentiating features of objects. ZSL for true categorization is attained at 0.45, or 76 percent, of the threshold value. This unsupervised algorithm divided the datasets into two groups: offensive and non-offensive. The same threshold value and distance algorithm can be used to categories un-labeled datasets (UDS). Unsupervised algorithms can classify any amount of unlabeled data with a very astounding 76 percent accuracy for text classification. One of the fundamental steps in classic machine learning or deep learning algorithms is training the algorithm, and for deep neural networks, a massive amount of training data is needed. These algorithms require a lot of computation. Contrarily, unsupervised algorithms have higher classification accuracy while being computationally less expensive.

CHAPTER 1. INTRODUCTION

1.1 MOTIVATION

The genetics of social media has had a direct impact on mass communication methods and goals. [1]. It was initially governed by ethical and social norms before the nativity of social media. Mass communication was initially used for awareness and cultivation of knowledge, effectively. But the use of mass communication is intensively influenced by the beginning of social media platforms.

Now people are using social media platforms, in their local/native language, to express themselves very commonly. The reason for this is, that they feel safe and consider it as a natural flare, for communication and expressing themselves, to use their own language on social media. The other reason for using the local language on social media might be is that people can only speak/ communicate in their native language.

The different pillars of society can broadly be classified into two types i.e. positive and negative. We can see the use of abusive/insulting or offensive language to express their rage or frustration about anything from the negative pillars along with positive views from positive pillars of the society on social media. The content on social media may include content like verbal, non-verbal, pictorial, etc. but the non-verbal type of communication is generally preferred by the people. So, the term cyberbullying on social media comes into the beginning from the negative point of view/feedback from negative users.

An increasing rate of cyberbullying in society is because of the attractiveness of social media platforms such as TikTok, Snack Videos, Instagram, etc. As we cannot bear the increasing ratio of cyberbullying, therefore, it is a very significant issue in society and needed to be resolved

on time in the favor of individuals. Therefore, we chose to take the initiative with the traditional local language Punjabi.

1.2 CONTRIBUTION

Resource-rich languages have automatic detection systems for monitoring and blocking offensive content but are very rare for low-resourced languages because of the non-availability of datasets. Researchers have focused their intentions to address the above-mentioned issue of cyberbullying for resource-rich languages like English, French, German and Arabic, etc. on a priority basis because of the easy availability of datasets on the Web for these languages.

Now in this study, we'll focus on Punjabi, the poor-resource language, and also discuss the work that has been done for resource-rich languages as a model.

In the past few years, the ratio of Cyberbullying has significantly increasing trend in low-resource languages like Pashto, Urdu, Punjabi, and many other local languages because researchers around the world have neglected the resource poor languages with power recourses. In this project, we offer a model for Punjabi, a very low resource language, which automatically detects offensive language/word/phrase from the feedback. And to work with Punjabi, an unlabeled dataset of 100 thousand comments/feedback, in roman Punjabi, is created from different social media platforms.

1.3 GOALS

The main aim of this work is to highlight the outlying challenge of cyberbullying in local languages that arose due to the negligence and unawareness of the speaking community. To

attain the above-mentioned innovation, we decided to work with a very popular native language Punjabi.

1.4 OBSERVATIONS

As previously stated, social media has had a significant impact on the means and goal of mass communication. [17]. Times ago, mass communication was initially the best suite for the awareness and cultivation of knowledge. Initially, it was ruled by ethical and social norms.

Nowadays, social media avowed individuals to connect and communicate their perceptions about anything via platforms like Twitter, Facebook, Instagram, Snapchat, YouTube, TikTok, etc. [21]. The latest research about social media influencers has revealed that people have minimal tolerance for their emotions and conduct, which imparts aggression to their behavior and content [7]. As a result, people use language that antagonizes the feelings of others. There is no directly interaction between users, which makes individuals to share their judgement minus any fright. Here comes the parturition of cyberbullying, which is a big challenge for researchers these days.

Despite the tenet for content published by various social media platforms, it's quite difficult to encounter the violations, manually [13, 33]. The reason is the huge tally of data on social media platforms. The privilege to users of social media daises for expressing their feelings and opinions in native languages makes it more difficult to detect the violations. Therefore, disgust talking and offensive/abusive language finding on societal media has become an active area under exploration nowadays [6].

1.5 SOCIAL MEDIA AS A KEY INFLUENCER IN CYBERBULLYING

Social media daises, interactive technologies, are the main influencer in the parturition of cyberbullying globally as they deliver a focal tag of message for individuals of several geographic locations, religions, skin colors and cultures troll each other by using offensive/violent language [3]. To transcribe their ruling, response, or remarks regarding online goods, articles and videos, People feel comfortable and favor using their native language [22]. Comments with quarrelsome arguments give grounds for cyberbullying and should not be appreciable [14]. The global community is facing a big challenge regarding precautionary measures that have to be taken to control the uplifting graph of cyberbullying in local languages as no mechanism is implemented yet to encounter such comments/feedback from the negative users on social media.

1.6 NATIONAL INTERESTS AND BENEFITS

Let's take the Pakistani community as an example of the effectiveness of the above-mentioned social virus. an increase of 83% is recorded in cybercrimes in Pakistan in the last three years is claimed by The News On the 28th of August, 2021, The statistics of social media subscribers in Pakistan show an increase of 24% and having a total stack of around 46 million according to a report published in February 2021. And according to the latest report published on 16th February 2022, there were 71.70 million social media consumers in Pakistan in January 2022. Therefore, Punjabi speaking community must have contributed significantly to the above facts and figures regarding social media being the 1st largest speaking language in Pakistan and the 10th largest speaking language in the world. The mentioned 83% increase is surely backed by cyberbullying on social media platforms. Punjabi speakers are much popular on TikTok and other social media platforms these days.

1.7 RESEARCH OBJECTIVE

According to the above-discussed facts and figures, social media aspects in our country are demanding an effective contribution to NLP (Natural Language Processing) for local languages in Pakistan.

Punjabi is the most popular and generally vocal languages in Punjab among the people. There are about 80.5 million people who speak Punjabi in Pakistan which is approximately 39% of the population. And about 122 million speakers around the world. Therefore, nowadays the Punjabi-speaking community is facing great challenges on cyber grounds and also facing the consequences of the uncontrolled and unsupervised beginning of social media platforms. As a result, an automatic method to detect, restrain, or prevent harsh language or divisive remarks before they are published online is required.

Here in this research work, we have proposed different models to tackle the issue mentioned above for roman Punjabi comments/feedback by Punjabi speaking community on social media platforms and will provide grounds to address the uncontrolled cyberbullying.

1.8 LITERATURE REVIEW

Researchers have focused to address the highlighted issue of cyberbullying in the resourceful languages in the last decades and a lot of work has been done in this regard because of available resources in the resource-rich languages. Most of the researchers, till now, have come across the above-mentioned challenge for several resource-rich languages i.e. English, Arabic, German and Indonesian, etc. [9, 11, 31].

In the past few years, ML techniques for Natural Language Processing (NLP) have been used extensively by researchers for offensive language and abusive comments from social media users [5, 8, 15].

Reference [1], has used the N-gram features extraction technique and ML models in Arabic to find the aggressive/violent language comments on YouTube. Similarly, the same technique and model have been used in the Indonesian language to recognize belligerent comments from social media [12].

Schneider et al. [25], for the German Language, come across the rising issue of cyberbullying by using the convolutional networks technique to detect the antagonistic comments on Twitter.

In 2019, G. I. Sigurbergsson and L. Derczynski used LSTM and Logistic regression techniques, for Danish and English Language, to utter the challenge of unethical and offensive comments detection [5].

Pelicon et al. (2021) presented a detection system for In the English language, there is a zero-shot cross-lingual offensive language and hate speech classification.

Akhter et al. (2020) used separate and joint n-gram techniques to find categories at character-level and word level. To detect objectionable language in Urdu and Roman Urdu text comments on social media, they used seventeen classifiers from seven machine learning algorithms.

Hammad et al. (2020) work with the Roman Urdu language and developed a dataset of 10,012 tweets in Roman Urdu. They proposed CNN-gram, a novel deep learning architecture, for the detection of hate speech and offensive language from Twitter.

All of the work done in the literature review discussed above has great significance in the field of (NLP). This work has imparted grounds to peruse the same for local languages. We did the

research work for Punjabi, a very popular and historical local but resource-poor language, and focused on the above-discussed challenge of cyberbullying.

YouTube is a highly used video website with millions of users around the world [4]. People watch YouTube content and share their points of view in the comment section. On YouTube, the Punjabi-speaking group appeared to be highly active. Cyberbullying traces can Punjabi speakers' content and comments on the YouTube platform can also be recognized. YouTube is the second-most-visited website behind Google. YouTube has a user base of over 2 billion people and billions of hours of video. Punjabi speakers give a fair amount to the viewer's stack. On the other side Facebook, suggestion a popular platform for individuals all over the world to communicate and share their material. [24].

TikTok is playing a vital role in the uplifting graph of cyberbullying, especially, in Pakistan and other countries like India nowadays [23]. According to a survey of January 2021, there are about 61.34 million people on internet and about 46 million people on social media subscribers in Pakistan [2]. In Asia, Punjabi is the most often spoken language. 39% people in Pakistan and especially in Punjab, use it as their native language. It is the world's 9th most widely spoken language (with 93 million speakers) and the Indian Subcontinent's third most spoken native language. Cyberbullying is on the rise in the Punjabi-speaking population. This is a social calamity that needs to be handled for the Punjabi-speaking community. People prefer to express themselves using English alphabets rather than Punjabi alphabets.

The issue of Cyberbullying is at its peak among the people of Pakistan. Therefore, we took responsibility to initiate the WAR against the social virus in the form of cyberbullying for the Punjabi-speaking community. We do our best to formulate an instinctive (automatic) model to control the virus of cyberbullying on the platform of social media in roman Punjabi. We

attained very attractive results for each suggested model in this study and encountered the aggressive/offensive comments in our dataset very efficiently.

1.9 THESIS STRUCTURE

This work comprises mainly 4 chapters. In chapter 1, we briefly introduce our research topic, motivation, contributions, goals, observations, National interests and benefits, and literature review. The chapter is summarized by discussing the thesis structure.

In chapter 2, we present the methodology that we have followed to achieve the milestone. This chapter includes the process of data collection, flow diagrams, and a brief discussion about the data classification and purification techniques.

In chapter 3, we talk about the proposed models and also discuss the outcomes of each model along with and comparison of their outcomes. Tables and graphs of comparison of the results of each model are also presented in this chapter.

4th and the very last chapter consists of a detailed conclusion to this spectacular achievement and recommendations for future contributions in this regard.

CHAPTER 2. METHODOLOGY

The complete proposed model is discussed in this chapter.

In first section, Collection of Data set of 100,000 words Roman Punjabi was done for offensive text classification. Then it was preprocessed and converted into a structured form. In preprocessing, all HTML tags, links, URLs, unnecessary characters, digits, and some stop words e.g., “is”, “he”, “we” etc. were removed as they do not contribute in the classification of offensive/abusive text. The basic objective of our thesis is to find the offensive language detection thorough zero shot learning this method usually work by correlating observed classes and through some additional information, non-observed classes. So we have implement this technique on the dataset, after we have applied following techniques on it as mentioned

The complete pipeline of the projected model is shown in Fig. 1.

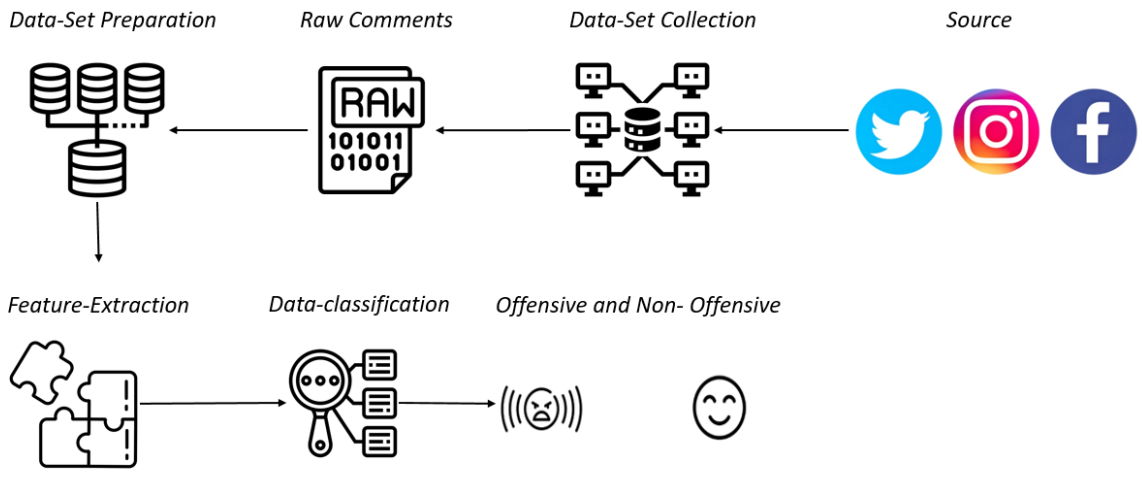


Figure 1. *Projected Model*

2.1 DATASET COLLECTION AND PREPROCESSING

Dataset is the key to proceed with the solution model to the subject. The first challenge that we faced while doing this research work was the collection of the Roman Punjabi dataset as online there was no such data set available.

From more than 800 videos and posts available on YouTube, Facebook and TikTok etc., we collected roman Punjabi comments manually by copy pasting, as there is no such tool available to extract commented data from any social media platform so far in our knowledge. We collect data sets of 1000 comments and 100 thousand comments separately.

After the collection we had a datasets in raw form consisting of undesirable punctuations, html codes/commands, HTTP links and emoji's etc. To refine our datasets we remove unwanted contents by applying preprocessing techniques. Our datasets are now ready to be labeled after successful cleaning.

The dataset of 1000 comments was labeled manually as offensive and non-offensive in terms of 'TRUE' for offensive and 'FALSE' for non-offensive. Therefore, the results generated in this work are 100% correct and the accuracy achieved is duly verified, as the annotation was done by reading each document (human approach and understanding). as the labeling approach was totally based on personal understanding of offensive content therefore, labeling of each comment as offensive or not was itself a challenging task. Hence, it is the result of a critical contextual analysis of each document in the corpus.

The challenge of labeling the dataset was fulfilled by analyzing each comment critically to declare it as TRUE or FALSE. Our analysis is so deep and precise that we also considered the contexts of each comment. We watched the video content with keen attention to understand the scenario that is being presented. Then, we followed the comments section and thoroughly examined and explored each comment of the users and then able to label them properly. We

are confident about the labeling of comments because it was done by human effort and understanding, based on keen observations.

Besides the cleaning and labeling of the data, we also addressed the repetitions of same comments. It was done intentionally, so that we could have as unique dataset as we could to enhance the effectiveness and accuracy of our models.

Character N-gram and Word N-gram

For sentence tokenization in a sequence of words (Word N-Gram) or tokenizing a word in an order of characters (Character N-Gram), the Character N-gram and Word N-gram techniques were utilized (Character N-Gram). The speech was also broken down into phonemes. In natural language processing (NLP), sequenced words were employed as features, whereas phonemes were utilized in speech processing. It assigned probability values to the sequenced words or characters utilized in the categorization process. This model can predict the next item in a series and complete sentences automatically. For text or audio categorization, classifiers employed token probabilities. The N-Gram approach is extended further in the following models. The next item in a series is predicted using an n-gram model.

It's a commonly used model in NLP, and one of its applications is sentence completion. There are two sorts of n-gram models, as the title suggests, and they are separated and explained in the tables below.

Table 1. *N-grams designing from roman Punjabi*

N-grams	**Roman Punjabi**
Sentence	Kia hal ha dost
Unigram	‘kia’, ‘hal’, ‘ha’, ‘dost
Bigram	‘kia hal, ‘hal ha’, ‘ha dost’
Trigram	‘kia hal ha’, ‘hal ha dost’

2.2 ZERO-SHOT LEARNING

Zero-Shot Learning (ZSL) is a ML problem in which an initiator detects samples from classes that were not viewed during training at test time and must predict which class they belong to. In general, these approaches work by associating observed and non-observed classes with auxiliary information that encodes distinguishing observable attributes of objects..[1]

For example, a set of images of animals along with some auxiliary textual descriptions of what an animals look like is given for classification. An artificial intelligence model that has been trained to recognize horses can still recognize a zebra even though zebra has never been given but it also knows that zebras look like striped horses.

2.2.1 Introduction

The capacity to accomplish a task without any training examples is recognized as zero-shot learning. Consider the instance of recognizing a sort of Words without ever seeing or trained on Words.

Recently, researchers of artificial intelligence have made significant development in building AI systems that can learn from large volumes of labelled data. This supervised learning paradigm has a record of producing specialized models that excel at the task for which they were trained. Unfortunately, supervised learning alone can only take artificial intelligence so far.

Building more intelligent generalized models that can execute various tasks and learn new skills without vast quantities of labelled data is hampered by supervised learning. Practically speaking, labelling everything in the world is difficult. Some jobs, like as training translation systems for low-resource languages, simply do not have enough labelled data. If AI systems can get a deeper, more complex understanding of reality beyond what is defined in the training

data set, they will be more beneficial and eventually bring AI closer to human-level intelligence.

Generalized understanding of the world, or common sense, is thought to make up the majority of biological intelligence in humans and animals. Humans and animals both have this common sense ability, but it has remained an open challenge in AI research since its creation

2.2.2 What is Zero-shot text classification?

Text classification, as we all know, is a natural language processing task in which the model must predict the classes of text data. We must utilize a huge amount of labelled data to train the model in the usual procedure, and they cannot predict using unseen data. The combination of zero-shot learning and text classification has pushed natural language processing to new heights. The basic goal of any zero-shot text classification model is to classify text documents without utilizing any labelled data or having seen any tagged text. Zero-shot classification implementations are mostly seen in transformers.

Another term that comes to mind when we talk about zero-shot text classification is few-shot classification, which is similar to zero-shot classification but uses a smaller number of labelled samples during training. We can also utilize the Flair for zero-shot classification, under the package of Flair we can also utilize various transformers for the NLP procedures like named entity recognition, text tagging, text embedding, etc.

2.2.3 How Zero Shot Learning Work?

The subsequent is a slope of the information that was used in Zero-Shot Learning:

Seen Classes:

The data classes that were used to train the deep learning model are known as seen classes.

Unseen Classes:

These are the data classes that the deep model must generalize to. These classes' data were not used in training.

Auxiliary Information:

To resolve the Zero-Shot Learning problem, some auxiliary information is compulsory because no tagged examples feel right to the hidden modules are accessible. Auxiliary information should provide descriptions, semantic information, and word embedding for all unseen classes.

2.2.4 How to Choose A Zero-Shot Learning Method?

The following are the two most frequent techniques to solving zero-shot recognition problems:

1 Classifier Based Methods

2 Instance Based Methods

2.2.4.1 Classifiers based method

For training the multiclass zero-shot classifier, most existing classifier-based approaches use a one-versus-rest solution. They train a binary one-versus-rest classifier for each unseen class. We divide classifier-based approaches into three subcategories based on the approach used to build classifiers.

Correspondence Method

The goal of correspondence techniques is to make the classifier for unseen classes by corresponding the binary one-versus- Each class has a rest classifier and a corresponding class prototype. Each class has only one associated prototype in the semantic space. As a result, this prototype might be considered the class's "representation." The goal of correspondence techniques is to learn a function that connects these two sorts of "representations."

Relationship Method

The purpose of these techniques is to create a classifier for unseen classes based on their inter- and intra-class relationships. Binary one-versus-rest classifiers for seen classes can be learned using data in the feature space. Meanwhile, determining the relationships between the seen and unseen classes can be done by computing the associations among related prototypes. Relationship techniques strive to create the classifier for the unseen classes using these learned binary observed class classifiers and these class relationships. Meanwhile, determining the relationships between the seen and unseen classes can be done by computing the associations among related prototypes.

Combination Method

Combination methods explain how to build a classifier for the unknown classes by combining classifiers for the basic parts that make up the classes.

Combination methods are said to have a list of "fundamental ingredients" that make up the classes. Each data point, in the observed and unobserved classes, is made up of these fundamental parts.

2.2.4.2 Instance-based methods

Instance-based approaches' try to obtain labelled instances for the unobserved classes first, and then train the zero-shot classifier with these instances. Based on the source of these instances, existing instance-based techniques can be classified into three groups, as shown below.:

Projection Method

Projection methods can create labelled examples for unseen classes by combining feature space instances and semantic space into a shared space.

In the feature space, there are labelled training instances belonging to the visible classes. Meanwhile, both the seen and unseen classes have prototypes in the semantic space. Instances and prototypes are vectors in the feature and semantic spaces, which are real number spaces. The prototypes can also be considered labelled instances in this perspective. As a result, we classified instances in two different areas (the feature and semantic spaces).

Synthesizing Methods

Synthesizing methods are used to generate labelled instances for unknown classes by synthesizing pseudo-instances using various ways.

In some ways, occurrences of each segment are presumed form of circulation in order to create the phony-instances. To begin, the unknown classes' distribution parameters must be approximated. Then, instances of previously unseen classes are created.

2.3 Words Embedding

Word embedding is a technique for presenting a word's characteristics. When words are employed in similar ways, word embedding, a sort of word representation, allows them to have comparable representations. In embedding, each word is represented as a real-valued vector in a predefined vector space. When a word is used in a text, the words that surround it determine its position within the vector space. A high-dimensional real-valued vector represents each word.

2.3.1 Word Embedding Algorithms

Technique of showing the structures of a word. Word embedding, a type of word demonstration, agrees words to have similar demonstrations when they are used in similar ways. Each word is encoded as a real-valued vector in a predetermined vector space during

embedding. When a word is employed in a text, its position in the vector space is determined by the words that surround it. Each word is represented by a high-dimensional real-valued vector.

2.3.2 Embedding Layer

A word embedding layer is a word embedding trained in conjunction with a neural network model for a specific natural language processing task, such as language modelling or document categorization. It necessitates document text preparation and cleaning in order for each word to be encoded in a single pass. The model specifies the dimensions of the vector space, such as 50, 100, or 300. The vectors are started with small random numbers. In a supervised fashion, the embedding layer is applied to the front end of a neural network and fitted using the Backpropagation method.

2.3.3 Word-2-Vec

It's a statistical method for quickly and efficiently learning a single word encoding from a text body. It's a way to make neural-network-based embedding training more effective. It has become the industry standard for pre-trained word embedding.

Two different learning models can be used as part of the word-2-vec strategy for learning word embedding:

- CBOW Mode
- Continuous Skip Model
- Continuous Skip Gram Model

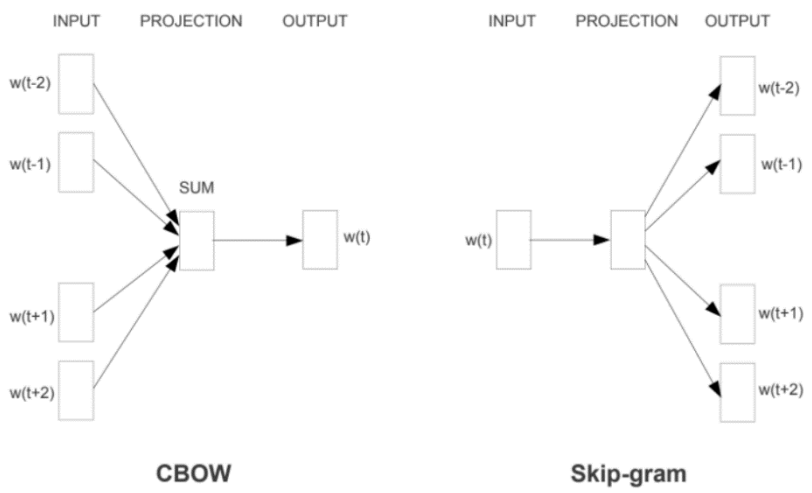


Figure 2. *CBOW and Skip-gram*

The embedding is learned by the CBOW model predicting the current word based on its context. When the current word is given, the continuous skip-gram model learns embedding by predicting the adjacent words.

2.4 Glove Method

It is an unlabeled learning method for gaining word vector symbols. The training is done on amassed global word-word co-occurrence statistics from a corpus. The ensuing representations highlight the interesting linear sub-structures of the word vector space.

2.4.1 Nearest Neighbor

The Euclidean sphere is a useful tool for determining how semantically or linguistically similar two words are. This figure can occasionally discover unusual but important terms that were not in the average person's vocabulary. The following words are alternatives for the lexical item frog:

0. *Frog*
1. *rana*
2. *lizard*
3. *eleutherodactylus*
4. *frogs*
5. *toad*
6. *litoria*
7. *leptodactylidae*

(shown in figure 2)

			
1. rana	3. eleutherodactylus	6. litoria	7. leptodactylidae

Figure 3. *Nearest Neighbor*

2.4.2 Linear substructures

For nearest neighbor evaluations, the similarity the relatedness of two words is measured by a single scalar produced using metrics. Because two words almost often have more intricate links than a single number can express, this simplicity might be problematic. Man and woman, for example, are identical in how they both describe humans; but, the two words are frequently seen as opposites since they highlight a center point along which humans differ.

A model must link more than one statistic with the word pair in order to catch the detail required to quantitatively distinguish man from woman. The difference in vectors between both the two word vectors is an obvious and simple choice for a larger set of discriminative numbers.

CBOW and Skip–Gram are two local context window approaches. In the previous post, we went through these in depth. CBOW trains several times faster and has somewhat greater

accuracy for frequent words, whereas Skip-gram performs well with minimal quantities of training data and represents even rare words.

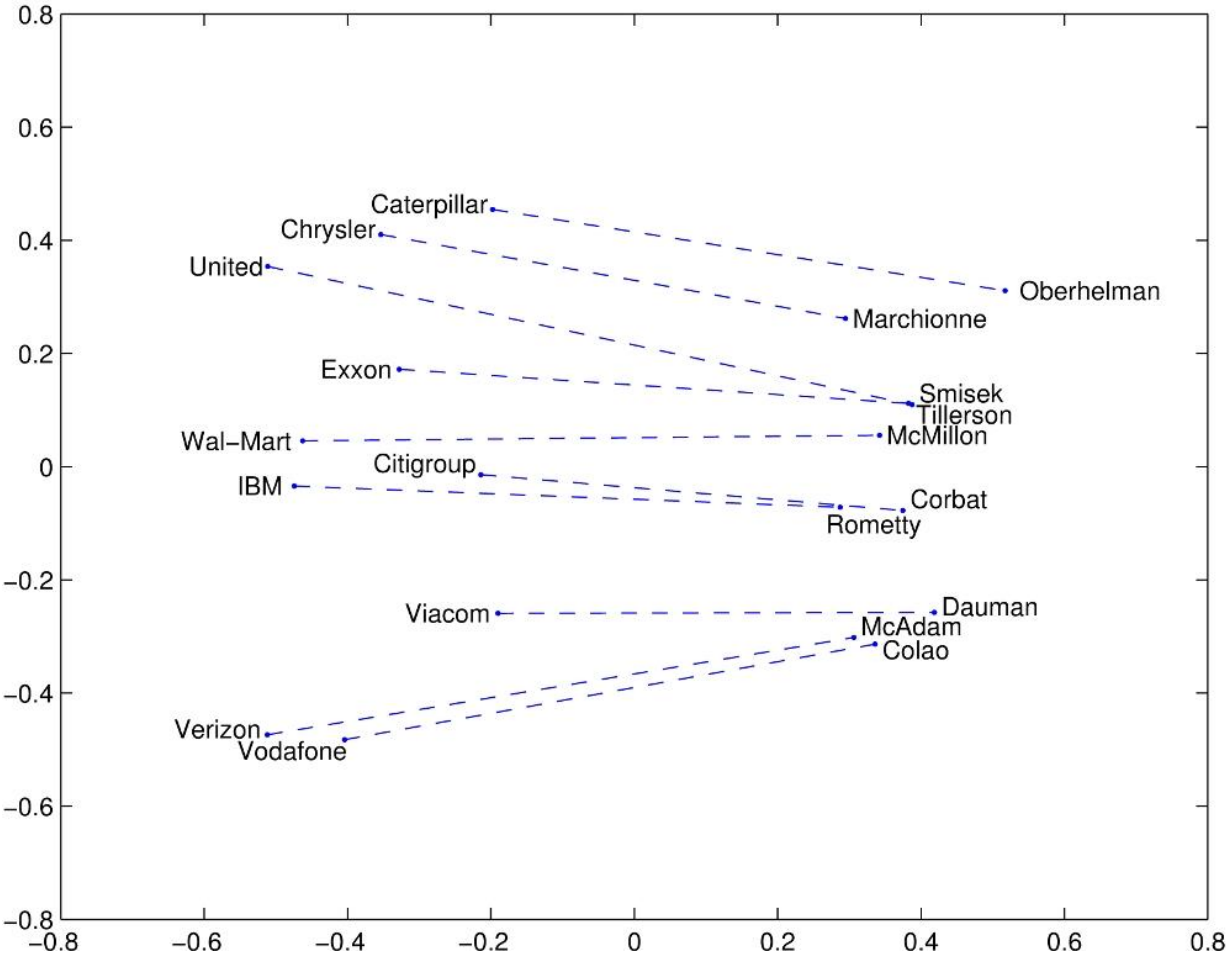


Figure 4. *Linear Substructures (Company)*

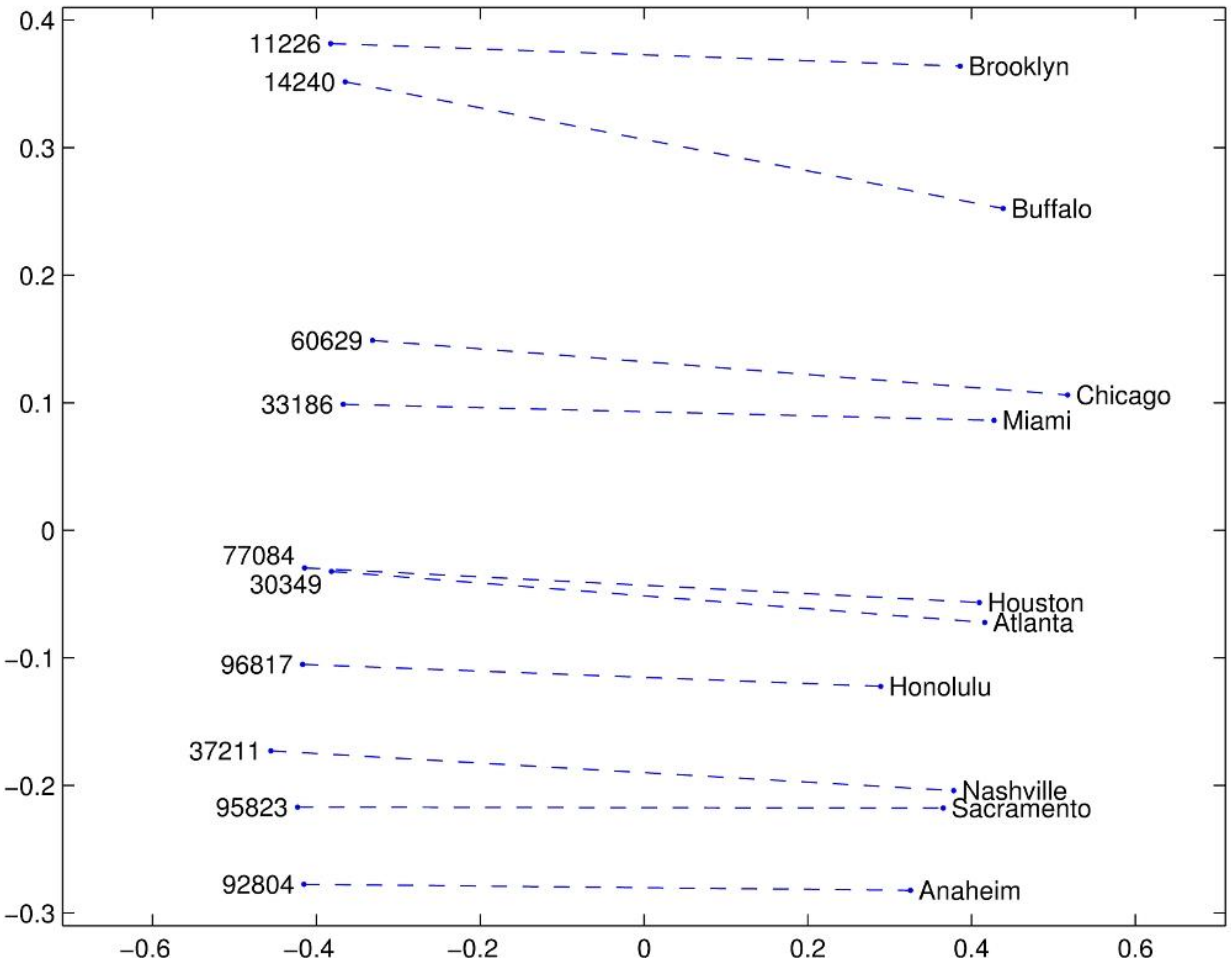


Figure 5. *Linear Substructures (ZipCode)*

2.4.3 Training

The GloVe model is skilled on the non-zero entrances of a global word-word co-occurrence matrix. It arranges how recurrently words happen at the same time with one extra within a given corpus.

The tools included with this bundle simplify the collection and preparation of co-occurrence data for use as input in algorithms. These pre-processing stages are separated from the main training code. It can be done out itself.

2.4.4 Model Overview

GloVe is a log bi-linear model with a weighted least squares objective. The model's core notion is built on the ratios of word-to-word co-occurrence probabilities, which can be used to encrypt meaning. Consider the likelihood of the target terms ice and steam co-occurring with a variety of vocabulary probing words. Here are some actual probabilities based on a corpus of 6 billion words:

Probability and Ratio	***K=solid***	***K=gas***	***K=water***	***K=fashion***
P(k|ice)	2.9×10^{-4}	4.6×10^{5}	3.1×10^{-4}	1.8×10^{-6}
P(k|steam)	2.3×10^{-4}	7.9×10^{-3}	2.2×10^{-3}	1.8×10^{-6}
P(k|ice)| P(k|steam)	*8.10*	8.4×10^{-1}	*1.46*	*0.97*

Table 2. E*xamples of designing*

2.5 Fast Text

It's a word2vec model that has been modified. Instead of generating vector for words directly, FastText presents each word as just an n-gram of characters. The fastText encoding of the

phrase "artificial" with n=3 is ar, art, rti, tif, ifi, fic, ici, ial, al>, with angle brackets indicating the start and the end of the word.

FastText is unusual in that it can create word vectors from morphological properties for unknown words or vocabulary concepts. Because morphology refers to a structure or syntax of words, FastText works more effectively for such tasks, while word2vec performs better for semantics duties.

2.6 Distance Metrics

To compute the similarity between data points, distance metrics are used. Whether for classification or clustering, an efficient distance measure increases the performance of our machine learning model.

Let's imagine we wish to tackle a classification or regression problem by means of the K-Means Gathering or k-Nearest Neighbor approach to construct clusters. What criteria will you use to compare different observations? What does it mean to claim that two points are similar?

Isn't that what will happen if their looks are similar? When we map these spots, they will be closer in distance to each other.

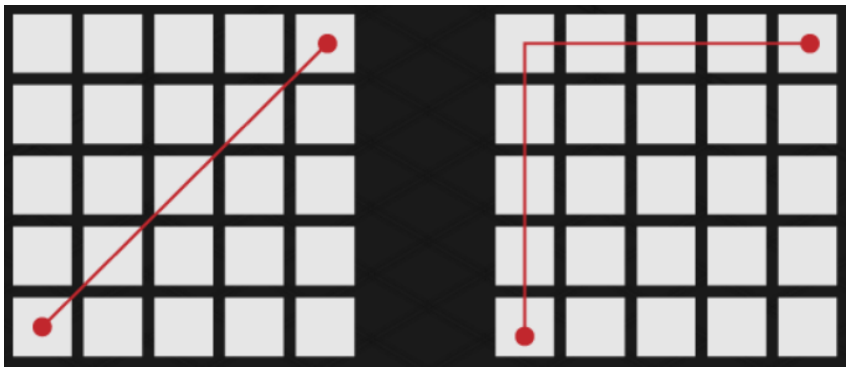
Distance Metrics Types in Machine Learning

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Hamming Distance

Let's start with the distance metric that is used in our thesis, Euclidean Distance.

2.6.2 Euclidean Distance

Euclidean Distance is defined as the shortest distance between two points. Most machine learning algorithms, including K-Means, employ this distance metric to calculate how similar observations are. Assume we have the two points listed below.



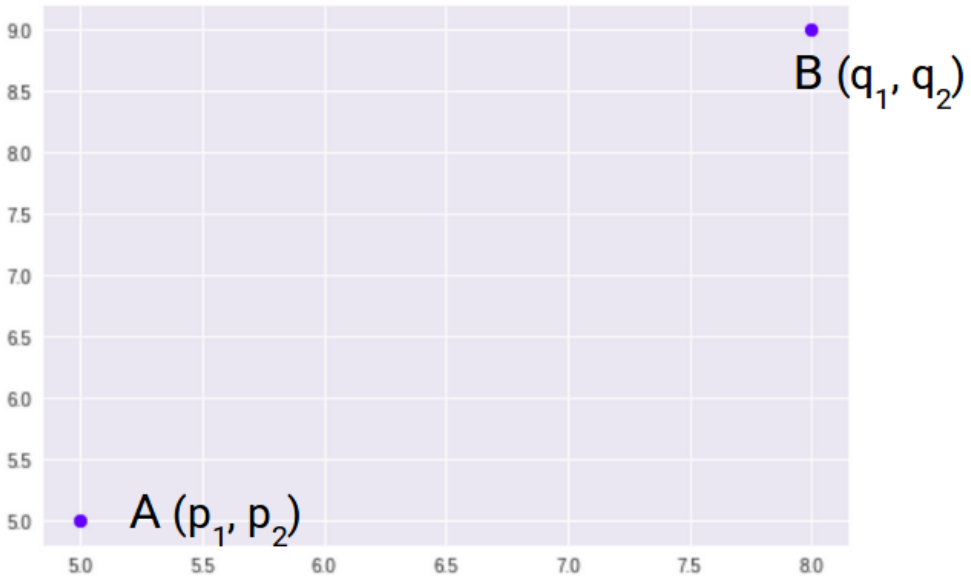


Figure 6. *Euclidean Distance between two points*

The formula for Euclidean Distance is:

$$d = ((p_1 - q_1)^2 + (p_2 - q_2)^2)^{1/2}$$

When working with two dimensions, we use the formula above. It can be used to every n-dimensional space. as:

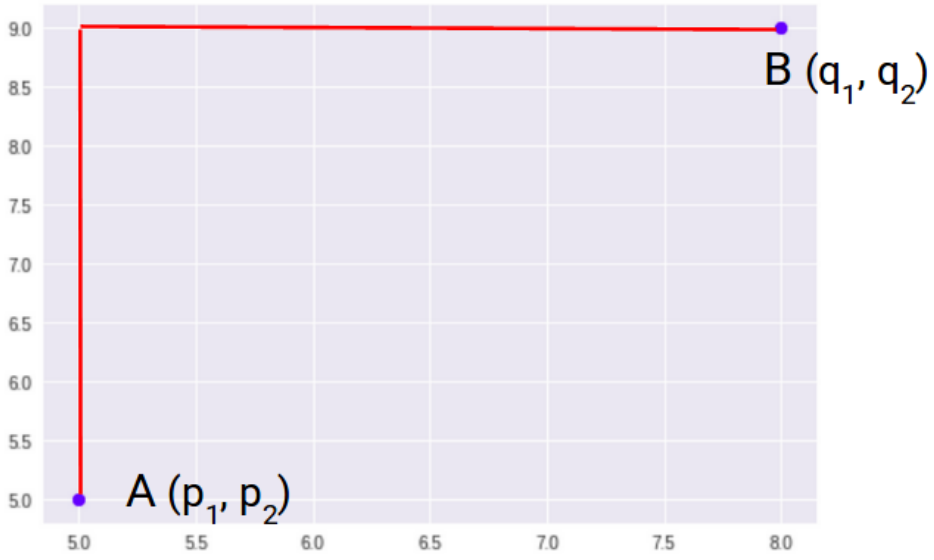


Figure 7. *Two dimensional Euclidean Distance*

$$D_e = \left[\sum_{i=1}^{n}(p_i - q_i)^2\right]^{1/2}$$

Where,

- n = quantity of dimensions
- pi, qi = main data points

2.6.3 Manhattan Distance

Total difference between two places in all dimensions is Manhattan Distance. Manhattan Distance can be expressed as:

Because the above representation is two-dimensional, we use the formula above while working with two dimensions. It is applicable to any n-dimensional space.$d = |p_1 - q_1| + |p_2 - q_2|$

And the generalized formula for an n-dimensional space is given as:

$$D_m = \sum_{i=1}^{n}|p_i - q_i|$$

Where,

- n = quantity of dimensions
- pi, qi = main data points

2.6.4 Minkowski Distance

The generalized version of Euclidean and Manhattan Distance is Minkowski Distance.

The Minkowski Distance formula is as follows:

$$D = \left[\sum_{i=1}^{n} |p_i - q_i|^p\right]^{1/p}$$

2.6.5 Hamming Distance

The Hamming Distance is a measure as to how similar strings of the same length are in terms of the number of points in which the respective character differ.

Let's consider the example to help you understand the concept. If we have two strings, say "Euclidean" and "Manhattan," we may compute the Distance Measure since their length are equal. We'll match strings one by one, character by character. Both strings have different first characters (e and m, respectively). Also, all strings' second characters (u and a) were different, and so forth.

2.7 Word Embedding of labeled and Unlabeled dataset

Word embedding is used to convey the features of a word in both labelled and unlabeled data in this section. When words are employed in similar ways, word embedding, a sort of word representation, allows them to have comparable representations. In embedding, each word is represented as a real-valued vector in a predefined vector space.

2.7.1 Labeled Data

When a word is used in a text, the words that surround it decide where it is classified in the vector space. A high-dimensional real-valued vector represents each word.

In a conceptual sense, word embedding reduces the gap between similar words compared to words with really no semantic link.

The second important aspect of word embedding is that regardless of how many unique words are in the text corpus, we will have the same amount of columns in the input matrices for developing models. This is a significant advantage over one-hot encoding, where the number of columns is usually equal to the number of unique words in a text.

2.7.2 Unlabeled Data

By dispersed word illustrations to encode word semantics, word embedding has aided a wide range of text analysis applications. Word representations are usually acquired through modelling native words, with the assumption that words with similar surrounding words are semantically related. We contend that in unsupervised word embedding learning, local contexts can only partially explain word semantics. Global contexts, which refer to larger semantic units like the document or paragraph in which the word appears, can capture many aspects of word semantics and complement local contexts.

2.8 Average Metrics

It's simple to score a model with binary classification using scoring measures like precision, recall, and F1-score. It becomes more difficult in the case of multi-class classification. Macro and micro averaging strategies are used to address the issues mentioned above. These methods are implemented by the Python Sklearn module. Later sections provide examples to

demonstrate this. The model's true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs) are used to determine the micro-average precision and recall score.

2.8.1 When should you employ micro- and macro-averaging scores?

Use the micro-averaging score when every event or forecast must be equally weighted. Use the macro-averaging score to evaluate the classifier's overall results in terms of the most prevalent class labels when all classes must be considered equally. Use a weighted macro-averaging score if there are class imbalances (different number of instances related to different class labels). The weighted macro-average is calculated by multiplying the score of the each classifier by the number of true instances while computing the average.

2.8.2 Micro-Average & Macro-Average Accuracy Scores for Multi-class Classification

Micro-average exactness scores for multi-class organization problems can be defined as the sum of true positives for all classes divided by all positive predictions. This is how it would appear mathematically.

$$Precision\ Micro\ Avg = \frac{(TP_1 + TP_2 + \cdots + TP_n)}{(TP_1 + TP_2 + \cdots + TP_n + FP_1 + FP_2 + \cdots + FP_n)}$$

The arithmetic mean of all precision scores from distinct classes is the macro-average precision score. The following is how it would appear mathematically:

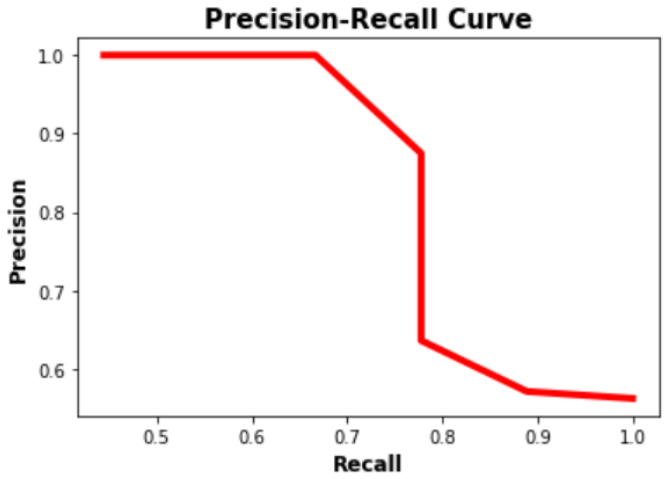
$$Precision\ Macro - Avg = \frac{\left(P_{rec_1} + P_{rec_2} + \cdots + P_{rec_n}\right)}{n}$$

2.8.3 Micro-Average & Macro-Average Recall Scores for Multi-class Classification

Micro-average recall notches can be distinct as the totality of true positives for all classes divided by actual positives in a multi-class classification issue (and not the predicted positives). Mathematically, it would look like this:

$$Recall\ Micro\ Avg = \frac{(TP_1 + TP_2 + \cdots + TP_n)}{(TP_1 + TP_2 + \cdots + TP_n + FN_1 + FN_2 + \cdots + FN_n)}$$

Macro-average The arithmetic mean of all memory scores from different classes is the recall score. The following is how it might look on a mathematical scale:



$$Recall\ MacroAvg = \frac{(Recall_1 + Recall_2 + \cdots + Recall_n)}{n}$$

Figure 8. *Precision-Recall Curve*

2.9 Text Classification

Text classification is a ML method for classifying textual data into a set of pre - defined categories. Text classifiers can organise, arrange, and classify almost any type of text, including documents, medical studies, and files, as well as internet text.

2.9.1 How Does Text Classification Work?

In manual text categorization, a human annotator evaluates the text's content and allocates it to the correct section. Even though this process can produce excellent results, it takes costs and time money.

Automated text categorization uses ML, NLP and other AI-guided technologies to categorize text more quickly, accurately, and consistently.

There are several ways to automatically classify text, but they always fall into one of three categories:

- Rule-based systems
- Machine learning-based systems
- Hybrid systems

2.9.2 Rule-based systems

Rule-based approaches sort text into categories by implementing a set of different language rules. These rules direct the system to use conceptually pertinent elements to identify appropriate categories based on a statement's content. Each act shall of a projected category and an antecedent or pattern.

Rule-based efficiency can be made over time to become more human-readable. There are certain drawbacks to this strategy, though. These systems call for substantial subject knowledge to be get started. Furthermore, they take a lot of time because creating rules for a complex system can be difficult and frequently calls for substantial research and testing. Rule-based systems are particularly difficult to scale and maintain since adding new rules can alter how previously applied rules operate.

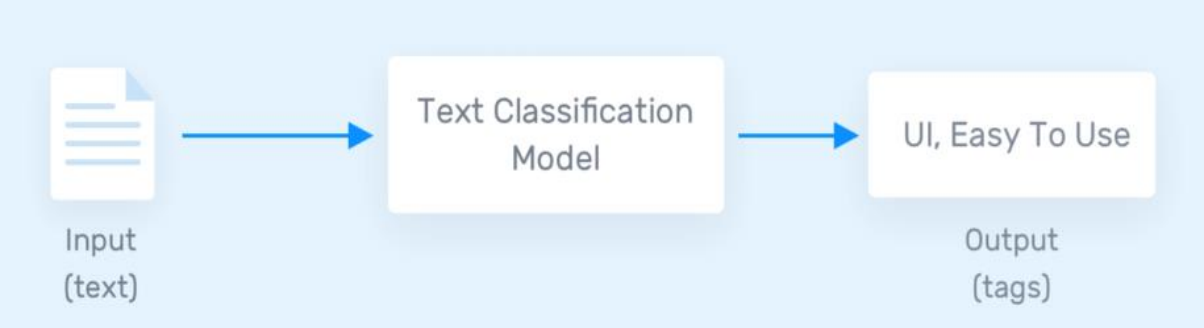


Figure 9. *Rule Based Systems*

2.9.3 Machine learning based systems

ML text categorization learns to produce categories based on prior observations rather than relying on rules developed by humans. Machine learning algorithms can understand complex links between text segments and how a specific output (in this case, tags) is predicted for a particular input by using pre-labeled examples as training data (i.e., text). Any text can be categorised or classified into a "tag," which is a predefined classification.

Feature extraction, which involves employing a technique to convert each text into a numeric representation in the form of a vector, is the first step in training a computational modeling NLP classifier.

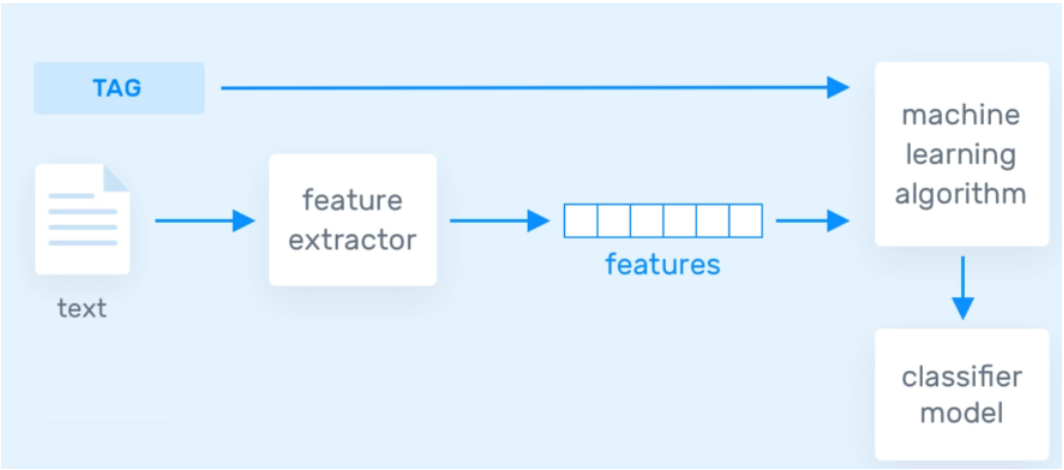


Figure 10. *Machine Learning Based*

2.9.4 Hybrid Systems

Hybrid techniques combine a machine learning-trained base classifier with a rule-based approach for even better outcomes. Special criteria for conflicting tags that the underlying classifier hasn't fully explained can be easily added to these hybrid systems.

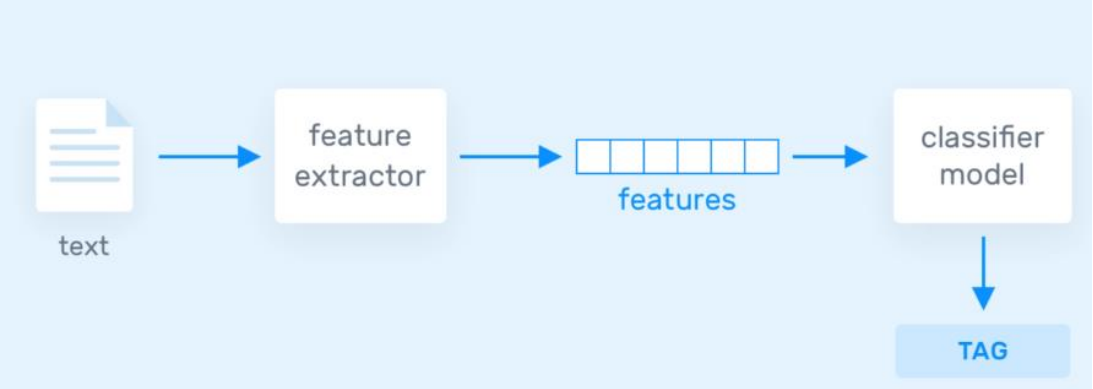


Figure 11. *Hybrid System*

CHAPTER 3. RESULT DISCUSSION

In this research work, we have used zero shot learning method for the classification of offensive comments in Roman Punjabi language. This method of text classification is an unsupervised learning method. We have gathered unlabeled dataset (UDS) of Roman Punjabi comments from social media. This data set consists of a corpus of 10,000 documents. To test the efficiency of the proposed algorithm, we have collected a separate dataset of 1000 documents and manually annotated it as True or False. It is called labeled dataset (LDS).

First, UDS has been preprocessed to remove all the unwanted content such as HTML tags, links, digits, icons, and punctuation marks etc. Then all the documents are tokenized in sequence of numbers to get document vectors. Converting texts into sequence of numbers retain the order of occurrence of words in sentences. Machine learning algorithms deal with the numbers only, therefore conversion is also necessary. Each vector has variable word length. To fix number, maximum length of words in vector is taken and all other documents are post padded with zeros to make it of same length. Next, these vectors of same length are passed through Word Embedding's layer which is a deep neural network model. We have set embedding's dimension as 32 which means that for every word in the document , embedding layer will calculate 32 related words in its embedding space. As a result, we have got a 3-dimensional vector of size 10,000 x 75 x 32 whereas 10,000 represent number of documents is dataset , 75 is the length of words in each document and 32 is the embedding's of each word. After that, we have taken column wise average of all embedding's for a single vector/document which resulted in 10,000 x 32 vector after taking average of all the documents.

We have created a small dictionary of Punjabi offensive words which contain 38 words. These words are collected from offensive comments of LDS and the words are most frequently used

in different offensive Punjabi comments. Offensive words are passed through same process as explained above to get the embedding's. It results in a vector of size 38 x 32 whereas 38 is the length of offensive words and 32 are embedding's for each offensive word. Next, we have taken column wise average of all the embedding's to get the centroids of offensive words. The final centroid vector of offensive words has dimension of 1 x 32.

Third step is to pass the labeled dataset (LDS) of 1000 documents and get its embedding's. It resulted in a vector of size 1000x 32. All the labels for LDS (1 for true and 0 for false) are stored in a buffer. Final step is to calculate the Euclidian distance of each vector in LDS with the centroids vector of offensive words. It results in 10000 distances and are normalized in range of 1 and 0. All the distances are stored in a buffer. Any distance value near to 1 implies a non-offensive comment while distance value closer to 0 implies offensive comment because it's closer to the vector of offensive words. We have set a threshold value and assign labels to 1000 documents based on threshold value. These calculated labels are compared to already stored labels (in buffer) to check the accuracy of algorithm.

To get proper classification accuracy, algorithmic parametric tuning is required. Thus, we have tested the performance of algorithm of 40 successive values of threshold, iteratively in a loop starting from 0.4 as shown in the table. The maximum accuracy of algorithm is obtained as 75.16 or 76% on the threshold value 0.45. It means that distance of any document from offensive vector more than 0.45 is classified as non-offensive while any document with distance lesser than threshold value is classified as offensive. The accuracy is obtained by comparing the assigned labels with the real labels in the labeled dataset.

S. No	Threshold Value	Efficiency
1	0.41	74.9
2	0.42	74.9
3	0.43	75.16
4	0.44	75.03
5	**0.45**	**75.16**
6	0.46	74.76
7	0.47	74.5
8	0.48	74.23
9	0.49	73.97
10	0.5	73.57
11	0.51	73.04
12	0.52	72.5
13	0.53	72.24
14	0.54	71.97
15	0.55	71.04
16	0.56	70.25
17	0.57	69.85
18	0.58	69.45
19	0.59	69.05
20	0.6	68.39
21	0.61	67.86
22	0.62	67.46
23	0.63	66.79
24	0.64	65.6
25	0.65	64.67
26	0.66	64.27
27	0.67	63.47
28	0.68	62.28
29	0.69	62.01
30	0.7	62.01
31	0.71	60.82
32	0.72	59.76
33	0.73	57.5
34	0.74	56.3
35	0.75	54.98
36	0.76	54.05
37	0.77	52.05
38	0.78	50.99
39	0.79	50.19
40	0.8	48.6

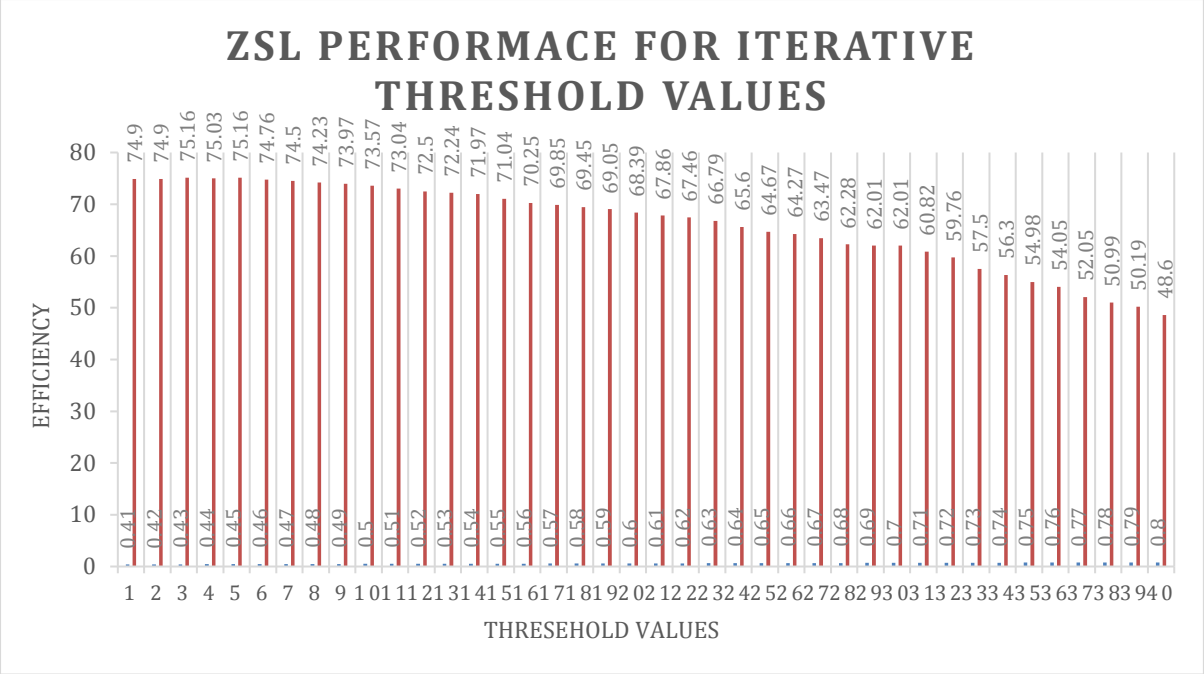


Fig: Graphical representation of efficiency value at different values of threshold

Above table shows that maximum efficiency of ZSL for true classification is achieved for the threshold value 0.45 i.e., 76 %. This unsupervised algorithm as resulted in two clusters of datasets i.e., offensive, and non-offensive. Un labeled dataset (UDS) can be classified using the same threshold value and distance method. 76% accuracy for text classification using unsupervised algorithm is very impressive and it can be used to classify any length of unlabeled data. In traditional machine leaning or deep learning algorithms, training the algorithm is one of the necessary steps and for deep neural networks, huge bulk of training data is required. Such algorithms are computationally intensive too. On contrary, unsupervised algorithms are computationally less costly with great classification accuracy.

CONCLUSION

This paper proposes an offensive text detection model for the roman Punjabi language. To the best of our knowledge, no dataset for the roman Punjabi language is available online. The dataset is compiled by gathering the comments from YouTube, Twitter, and Facebook. A total of 100,000 thousand comments were gathered for the creation of the roman Punjabi corpus. One of the main contributions of this work is the collection and compilation of the roman Punjabi dataset. The corpus created will be made available for the researcher working in this domain. This research work proposes a model for "Punjabi", a very low resource language, which automatically detects offensive language/words present. To create a dataset for roman Punjabi, we select 100 thousand and 1000 comments/feedback separately from different social media platforms and then the dataset of 1000 comments was labeled as offensive and non-offensive manually. The proposed model ZSL is a machine learning problem in which a beginner detects models from classes that were not viewed through training and predicts the category toward which they belong. The observed/seen and non-observed/unobserved categories are combined using zero-shot approaches, which use auxiliary information to represent observable differentiating features of objects. ZSL for true categorization is attained at 0.45, or 76 percent, of the threshold value. This unsupervised algorithm divided the datasets into two groups: offensive and non-offensive. The same threshold value and distance algorithm can be used to categories un-labeled datasets (UDS). Unsupervised algorithms can classify any amount of unlabeled data with a very astounding 76 percent accuracy for text classification. One of the fundamental steps in classic machine learning or deep learning algorithms is training the algorithm, and for deep neural networks, a massive amount of training data is needed. These algorithms require a lot of computation. Contrarily, unsupervised algorithms have higher classification accuracy while being computationally less expensive.

REFERENCES

[1] G. I. Sigurbergsson and L. Derczynski, “Offensive Language and Hate Speech Detection for Danish,” pp. 1–13, 2019.

[2] Hammad Rizwan, Muhammad Haroon Shakeel, Asim Karim Hate-Speech and Offensive Language Detection in Roman Urdupages 2512–2522, November 16–20, 2020.

[3] H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection,” IEEE Access, vol. 6, pp. 13825–13835, 2018.

[4] F. Noor, M. Bakhtyar, and J. Baber, “Sentiment Analysis in E-commerce Using SVM on Roman Urdu Text,” vol. 285. pp. 213–222, 2019.

[5] A. Alakrot, L. Murray, and N. S. Nikolov, “Towards Accurate Detection of Offensive Language in Online Communication in Arabic,” Procedia Comput. Sci., vol. 142, pp. 315–320, 2018.

[6] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard, “Common sense reasoning for detection, prevention, and mitigation of cyberbullying,” ACM Trans. Interact. Intell. Syst., vol. 2, no. 3, 2012.

[7] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, “Detecting offensive language in social media to protect adolescent online safety,” Proc. - 2012 ASE/IEEE Int. Conf. Privacy, Secur. Risk Trust 2012 ASE/IEEE Int. Conf. Soc. Comput. Soc. 2012, pp. 71–80, 2012.

[8] H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection,” IEEE Access, vol. 6, pp. 13825–13835, 2018.

[9] R. Sunil, “Understanding support vector machine (svm) algorithm from examples (along with code),” Analytics Vidhya.

[10] J. M. Schneider, R. Roller, P. Bourgonje, S. Hegele, and G. Rehm, “Towards the Automatic Classification of Offensive Language and Related Phenomena in German Tweets,” no. Konvens, 2018.

[11] M. Bouazizi and T. Otsuki, “A Pattern-Based Approach for Sarcasm Detection on Twitter,” IEEE Access, vol. 4, pp. 5477–5488, 2016.

[12] G. IngiSigurbergsson and L. Derczynski, “Offensive Language and Hate Speech Detection for Danish,” arXiv e-prints, p. arXiv:1908.04531, Aug. 2019.

[13] T. Ishisaka and K. Yamamoto, “Detecting nasty comments from BBS posts,” PACLIC 24 - Proc. 24th Pacific Asia Conf. Lang. Inf. Comput., pp. 645–652, 2010.

[14] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, “Offensive language detection using multi-level classification,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6085 LNAI, pp. 16–27, 2010.

[15] R. Pelle, C. Alcântara, and V. P. Moreira, “A classifier ensemble for offensive text detection,” 2018, pp. 237–243.

[16] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, “A lexicon-based approach for hate speech detection,” Int. J. Multimed. Ubiquitous Eng., vol. 10, no. 4, pp. 215–230, 2015.

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

[18] B.-S. Kwon, R.-J. Park, and K.-B. Song, “Short-term load forecasting based on deep neural networks using lstm layer,” Journal of Electrical Engineering & Technology, vol. 15, no. 4, pp. 1501–1509, 2020.

[19]S. Lewandowsky, M. Jetter, and U. K. Ecker, “Using the president’s tweets to understand po-litical diversion in the age of social media,” Nature communications, vol. 11, no. 1, pp. 1–12, 2020.

[20]K. Buchanan, L. B. Aknin, S. Lotun, and G. M. Sandstrom, “Brief exposure to social me-dia during the covid-19 pandemic: Doom-scrolling has negative emotional consequences, but kindness-scrolling does not,” Plos one, vol. 16, no. 10, p. e0257728, 2021.

[21]J. W. Patchin and P. D. S. Hinduja, “Tween cyberbullying.”

[22]D. Ali and L. Xiaoying, “The influence of content and non-content cues of tourism information quality on the creation of destination image in social media: A study of khyber pakhtunkhwa, pakistan,” Liberal Arts and Social Sciences International Journal (LASSIJ), vol. 5, no. 1, pp. 245–265, 2021.

[23]S. Kiritchenko, I. Nejadgholi, and K. C. Fraser, “Confronting abusive language online: A sur-vey from the ethical and human rights perspective,” Journal of Artificial Intelligence Research, vol. 71, pp. 431–478, 2021.

[24]H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection,” IEEE access, vol. 6, pp. 13 825–13 835, 2018.

[25]A. Bisht, A. Singh, H. Bhadauria, J. Virmani et al., “Detection of hate speech and offensive language in twitter data using lstm model.”

[26] A. Balayn, J. Yang, Z. Szlavik, and A. Bozzon, “Automatic identification of harmful, aggres-sive, abusive, and offensive language on the web: A survey of technical biases informed by psychology literature,” ACM Transactions on Social Computing (TSC), vol. 4, no. 3, pp. 1–56, 2021.