Exploring Sentiment Analysis in Low-resource Languages

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

Monil Gokani
2018114001
monil.gokani@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
December, 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Sentiment Analysis in Resource-poor Languages" by Monil Gokani, has been carried out under my supervision and is not submitted elsewhere for a degree.

January 9, 2024

Adviser: Prof. Radhika Mamidi

To my family and my friends

Acknowledgments

I would like to thank my research advisor, Prof. Radhika Mamidi, without whose support and guidance this thesis would not be possible. She encouraged me to explore this field at my own pace and was always taking the time to discuss ideas and directions with me, which guided my research journey. I would also like to express my gratitude to all of my professors from LTRC - Prof. Manish Srivastava, Prof. Dipti Misra Sharma, Prof. Vasudeva Verma, and many others - who guided my through the world of NLP and Computational Linguistics, and whose courses and discussions cultivated my perspective of the field.

I would also like to thank all of my family - my parents, who were a constant source of emotional strength and motivation, as well as my elder sister, who has been my biggest source of inspiration for as long as I can remember.

I also want to thank all of my friends - KV Aditya Srivatsa, Mukund Choudhary, Ishan Sanjeev Upadhyay, Tathagata Raha, Bharathi Ramana Joshi, Rishav Kundu, Nomaan Qureshi, Shivaan Sehgal, Gunjan Gupta, and many others - who have been a never-ending source of motivation, inspiration, and joy throughout my years at this institute.

Abstract

Sentiment Analysis is an important task for analysing online content across languages for tasks such as content moderation and opinion mining. However, state-of-the-art NLP modelling techniques often require a large amount of training data to achieve their results. Unfortunately, high-quality annotated data is often a rare commodity for many languages other than English, including most Indian languages. We attempt to tackle this data scarcity in this thesis in two ways - by creating additional resources, and by exploring more data-efficient modelling techniques.

Over the past few years, while some significant resources for Sentiment Analysis have been developed in several Indian languages, there do not exist any large-scale, open-access corpora for Gujarati. In this thesis, we present and describe the Gujarati Sentiment Analysis Corpus (GSAC), which has been sourced from Twitter and manually annotated by native speakers of the language. We describe in detail our collection and annotation processes and conduct extensive experiments on our corpus to provide reliable baselines for future work using our dataset.

We then explore modelling techniques that work well in a low-resource setting by experimenting with AfriSenti, a collection of sentiment analysis datasets in 12 African languages. We propose an XGBoost-based ensemble model trained on emoticon frequency-based features and the predictions of several statistical models such as SVMs, Logistic Regression, Random Forests, and BERT-based pre-trained language models such as AfriBERTa and AfroXLMR. We also report results from additional experiments not in the system and conduct an ablation study to observe the effects of different types of models and features on the final ensemble.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sentiment Analysis is the task of identifying the sentiment (i.e., the polarity) of a piece of text, such as a tweet, an article, a review, etc. It has received much attention in recent years thanks to the increasing penetration of the internet and the blooming of e-commerce and social media. Great progress has been made towards solving this task for languages such as English which have an abundance of linguistic resources (such as large, high-quality, annotated datasets, like the IMDb movie reviews dataset[1]), with multiple models achieving over 95% accuracy [2, 3, 4]. However, it still remains a challenging task for low-resource languages, which is generally the case for many Indian languages. In this thesis, I explore creating new resources for a resource-poor language and experimenting with various modelling methods on a large set of languages with small sentiment analysis datasets.

1.1 Motivation and Scope

The last decade or so has seen an explosion in the use of active internet users in India. Compared to 2010, when a mere 7.5% of the total population had access to the internet, over 45% in 2021 (see Figure 1.1).[1] With this drastic increase in internet accessibility, as well as the promotion of digital platforms by the Government of India through "Digital India" initiatives, there has been a corresponding rise in the number of active social media users as well as e-commerce platforms. The number of social media users increased by 500% since 2015, rising 142.23 million active users to a staggering 862.08 million active users in 2023.[2] In a similar time frame, the e-commerce market in India is expected to grow from US$38.5bn in fiscal year FY2017 to nearly US$188bn in FY2025, nearly a five-fold increase.[3] As a result, online platforms have become the largest available audiences for marketing, as well consumer surveying, making tasks like sentiment analysis, that can extract useful information such as customer preferences, complaints, etc. a vital component of this digital ecosystem.

[1] Data Commons 2023, *Data Commons*, viewed 17 Aug 2023, https://datacommons.org

[2] https://www.statista.com/statistics/278407/number-of-social-network-users-in-india/

[3] E-commerce Industry Report, May 2023, Indian Brand Equity Foundation https://www.ibef.org/industry/ecommerce

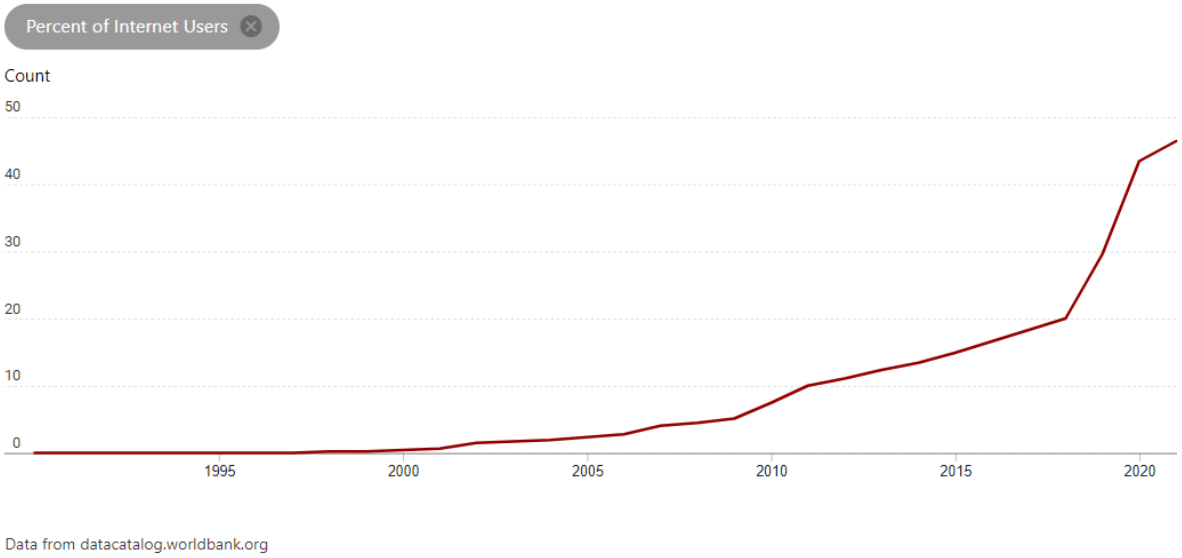

Figure 1.1: Growth of internet users in India (in percentage of population)

With increasing accessibility allowing the use of different languages on the internet, a large part of this new ecosystem requires processing information in these different languages. Sentiment analysis as a task is vital for gaining insight into the minds of these users. It can be used to coax out a plethora of valuable information from the content created by users on online platforms, such as identifying issues with a newly launched product by analysing reviews containing a negative sentiment or gauging the reaction of the user base to specific features or policies by analysing reviews mentioning those specific subjects. Doing these tasks in languages native to the users would allow us to target the needs or grievances of specific consumer groups.

Unfortunately, outside of English and a few other languages, there is a significant lack of high-quality annotated data for supervised machine-learning approaches. These languages are hence called resource-poor languages. Indian languages, including Hindi, the most widely spoken language in India, fall in this category. This lack of resources is the main motivation behind my work in this thesis.

This thesis attempts to address the complete absence of reliable, open-access sentiment analysis corpora in Gujarati by creating a new dataset for Gujarati sentiment analysis by annotating Twitter data. It then explores using different configurations of ensembled classifiers (2.1.3) for a set of African languages, another resource-poor domain, with the aim to explore how different features and algorithms interact with each across a varied collection of datasets.

1.2 Thesis Contribution

We explore the task of Sentiment Analysis, particularly in low-resource environments such as when working in Indian and African languages. We explore both resource creation and modelling approaches in this thesis. The major contributions of this thesis are:

- Creating a new dataset for sentiment classification in Gujarati sourced from Twitter, manually annotated by native speakers, which is the first such open-access dataset available for Gujarati.
- Experimenting with the Gujarati dataset using multiple different types of techniques to establish reliable baselines for future work.
- Experimenting with different ensembling configurations and incorporating emoticon features for sentiment classification in 12 different African languages.

1.3 Thesis Layout

- Chapter 1 describes the motivation behind the problems tackled in this thesis and the major contributions made in it.
- Chapter 2 describes the background of Sentiment Analysis and Text Classification in which this thesis is set, providing additional context and explanations for the various approaches and methods used in the subsequent chapters.
- Chapter 3 describes GSAC, the new gold-standard Twitter dataset for Gujarati Sentiment Analysis, and establishes baseline results on the dataset.
- Chapter 4 describes our experiments with exploring Emoticon features and ensembling multiple models across 12 different African languages.
- Chapter 5 is the conclusion of the thesis, elaborating on shortcomings in our work and discussing potential future work.

Chapter 2

Background and Related Work

This chapter provides a brief background about work done for sentiment analysis in recent years, focusing specifically on elements relevant to this thesis. Sentiment Analysis is a task where the objective is to determine the polarity of the tone of a particular piece of text, such as categorising it as positive or negative. It is generally treated as a text classification task, with the sentiment labels being the ”classes”. Hence, many of the techniques covered in this chapter are highly relevant to any text classification task in general. Since this thesis deals entirely with sentiment analysis at the sentence level, this chapter also does not cover aspect-based sentiment analysis, which deals with identifying opinions about specific parts (or ”aspects”) of the input.

2.1 Modelling Approaches for Sentiment Analysis

The earliest methods for sentiment analysis involved rule-based models, generally using some form of knowledge base such as a sentiment vocabulary. Since sentiment analysis is essentially a text classification task, machine learning-based approaches took over as feature-vector-based classification models like Naive Bayes, Logistic Regression, SVMs, etc., became more popular. This eventually evolved into deep learning models such as NNs, RNNs and LSTMs. Finally, With the introduction of the transformer by Vaswani et al. [5] in 2017, pre-trained language models based on the transformer architecture fine-tuned for text classification have become the new standard for text classification (including sentiment analysis), achieving state-of-the-art performance on a diverse array of classification tasks.

This section briefly explains some of these approaches, focusing specifically on those directly relevant to this thesis.

2.1.1 Statistical ML models

Early approaches to using ML algorithms for text-related tasks first required converting the text into a feature vector, before using that vector to train or use an ML model. Hence, this section is divided into two parts - feature extraction and models.

2.1.1.1 Feature Extraction

While feature extraction is an entire research domain in itself, this thesis mainly uses two types of feature extraction algorithms - Bag-of-Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF).

Bag-of-Words (BoW) is a feature extraction method that treats a document as a "bag" of its individual words, disregarding word order. It creates a vector where each dimension represents a unique word and its count in the document. While BoW efficiently captures word frequency information, it ignores word order and thus misses out on some context. Despite that, it is still a computationally efficient method that achieves competent results for text classification.

Term Frequency - Inverse Document Frequency (TF-IDF) is a feature extraction technique that emphasizes words that are distinct to a document by combining their frequency in a document (TF) with its rarity across all documents (IDF). High TF-IDF values are assigned to words that appear frequently in a document but infrequently in the entire corpus, indicating their significance to that document. This makes TF-IDF particularly suited to classification tasks, where the objective is to find characteristics that distinguish one document (or sample) from all other documents.

Another popular feature-extraction approach before transformers was to train neural networks for language modelling and use the internal representations of those networks as the vector representation of the corresponding individual tokens. There were several popular algorithms to train these representations from raw corpora, such as Word2Vec[6], GloVe[7], and fastText[8]. However, these representations are significantly more expensive to generate than BoW or TF-IDF, and not as good as those generated from transformer-based architectures. Hence, we decided not to experiment with these in this thesis.

2.1.1.2 Classification Algorithms

While transformer-based architectures have generally outperformed older, simpler classification algorithms covered in this section, these algorithms also require exponentially lesser amounts of computational resources and time. This can lead to interesting applications and experiments, such as combining multiple lighter, distinct models to increase the system's overall performance, as discussed in Section 2.1.3. Additionally, they also provide useful baselines for comparing the performance of other models. We use the following models in the experiments in this thesis:

- **Naive Bayes Classifier** is a probabilistic classifier based on Bayes' theorem. Using the input features, it calculates the probability of each class and then assigns the class with the highest probability. It makes the "naive" assumption that the input features are all independent of each other, which significantly simplifies calculations. Despite not capturing complex dependencies well due to this assumption, it is a useful choice for situations where computational resources are not easily available.

- **Logistic Regression** models the probability of an instance belonging to a certain class using the logistic function. It computes a weighted sum of input features, applying a sigmoid function to yield a value between 0 and 1, representing the likelihood of the positive class. If the probability exceeds a threshold, the instance is classified as positive; otherwise, negative. It can be extended to handle multiple classes by having multiple classifiers that train to distinguish between every pair of classes (one-vs-one approach) or by having multiple classifiers that train to distinguish one class from every other class combined (one-vs-rest approach).

- **Support Vector Machines** are a class of binary classification algorithms that aim to classify data by finding the best decision boundary (called a hyperplane) that can separate the data points in the vector space (where each sample is a vector of its features) into the two classes. It tries to find the best such hyperplane by trying to maximise the distance between the closest positive and negative samples to the hyperplane. These can also be extended to a multi-class classification problem using the same one-vs-one or one-vs-rest approach as Logistic Regression.

- **Random Forests** is a type of classifier model based on an ensemble of **decision trees**. Decision trees are hierarchical structures that make sequential decisions based on features, enabling data classification or regression by splitting it into subsets using if-then rules. During training, the ensemble constructs multiple decision tree models trained on a random subset of the data and features and combines their outputs through voting or averaging, which reduces the chances of overfitting and improves model performance.

- **Multi Layer Perceptron** is a simple feedforward artificial neural network architecture. It consists of input, hidden, and output layers, with fully connected nodes. Training occurs through standard forward and backpropagation algorithms. It is the simplest type of neural network but can still capture more complex dependencies than earlier models.

2.1.2 Deep Learning Approaches

Neural networks revolutionised the field of machine learning, including NLP, with their ability to capture significantly more complex relationships between input features. Unfortunately, they were still very limited in their ability to capture information from sequential data, like text. To tackle this, the Recurrent Neural Network (RNN) [9] was proposed, which introduced the concept of a hidden state, which contained the combined outputs of the previous states of the system and enabled it to process sequential input.

RNNs still suffered from losing context information for long-range dependencies due to the vanishing gradient problem, where older states of the cell would be lost due to the continuous addition of newer information. Long Short-Term Memory [10] networks (LSTMs) emerged as a modification of the basic RNN architecture to address this vanishing gradient problem. They incorporated specialized memory cells, maintained and updated separately from the hidden state, allowing them to capture and remember

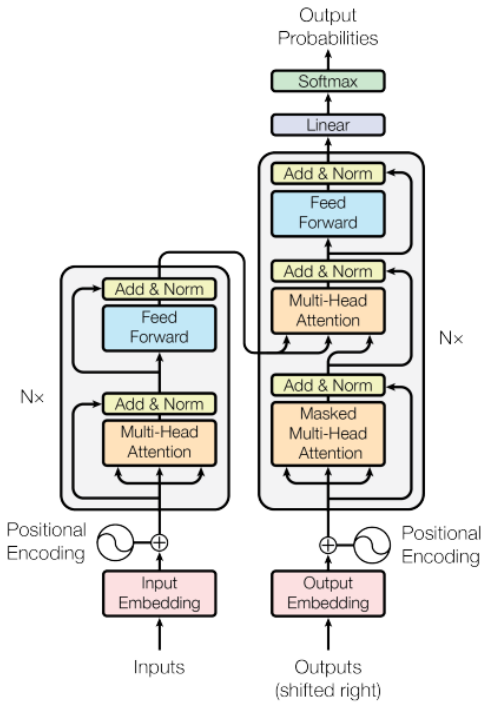

Figure 2.1: The Transformer model architecture

information over longer sequences than RNNs. However, LSTMs were still limited in their ability to capture very long-range dependencies and constricted by the need to process input one token at a time, which made them very slow to train. These two major issues were addressed with the introduction of the transformer architecture by Vaswani et al. [5]. Since then, it has become one of the most popular and best-performing approaches for a wide range of NLP tasks, including sentiment analysis.

Transformers had two significant advantages over previous sequence-to-sequence architectures - the ability to process entire sequences at once, which made them computationally efficient, and their ability to weigh different parts of an entire sequence through a self-attention mechanism. Because of this, they are very effective for capturing long-range dependencies, a long-standing limitation of previous neural architectures, making them particularly well suited for handling large amounts of text data. Figure 2.1 provides an overview of the model architecture.

This new architecture gave rise to a new generation of language models, which significantly outperformed older models at many tasks, including sentiment analysis. This section describes some of these models that are relevant to this thesis.

- **BERT** (Bidirectional Encoder Representations from Transformers) [11] is one of the first and most widely used transformer-based language models introduced by Google AI. It uses a bidirectional transformer architecture trained on 13 GB of raw English text, achieving state-of-the-art performance on eleven NLP tasks.

The pre-training process for BERT comprises of two training tasks - Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM involved replacing 15% of the input tokens with a 'mask' token and training the model to predict the masked tokens. NSP trained the model to predict if a given pair of sentences was in the correct order, which should let the model capture sentence-level dependencies.

This pre-trained model can then be fine-tuned for specific tasks, such as sentiment analysis, by training the model on smaller, task-specific datasets. This lets the model capture task-specific elements while leveraging the extensive language-specific knowledge that it captured during the pre-training process.

mBERT was a multilingual version of BERT. It was trained using the same architecture and methods but was trained specifically for handling multilingual use cases. The training data for mBERT consisted of the Wikipedia dumps of 104 languages with the largest Wikipedias.

- **RoBERTa**[12] used the same basic architecture as BERT but introduced key differences and used significantly more training data to outperform BERT. It removed the NSP task from pre-training and introduced dynamic masking for the MLM task. Additionally, compared to the 13GB of text used by BERT, RoBERTa uses 160 GB of data and is trained for 100 epochs against BERT's 40.

  XLM-RoBERTa [13] is a multilingual variant of RoBERTa that is trained on over 100 languages, using 2.5TB of CommonCrawl data.

- **ALBERT** [14] is based on the original BERT architecture but with two key differences. First, it factorised the large embedding matrix of the model, which significantly improved training speed and reduced memory consumption. Cross-layer parameter sharing lets the model dramatically reduce its number of trainable parameters. Both of these let the model achieve comparable performance to BERT while using significantly fewer computational resources.

  IndicBERT[15] is a multilingual ALBERT model pre-trained exclusively on 11 Indian languages - Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, and Telugu - and English. The training corpus contains a combined total of nearly 9B tokens, with Hindi being the largest component at 1.84B tokens. It achieves comparable or better performance than models like mBERT or XLM-RoBERTa while having significantly few parameters due to using the ALBERT architecture. It is one of the most popular models for Indian languages.

The models used in this thesis are either variants of these models fine-tuned for different languages or pre-trained using the same methods as them.

2.1.3 Ensembling

Ensembling methods are a class of learning algorithms that attempt to combine outputs from multiple classifiers to increase the robustness of the overall system. The main idea behind these algorithms is that

by ensuring the internal classifiers are trained to capture different characteristics of the training data, we can combine them into a system that generalizes better than any individual classifier could [16]. Recent experiments with ensemble learning methods to combine different classifiers have demonstrated their ability to increase the robustness and performance of systems [17, 18].

Some of the techniques used to combine classifiers include:

- **Voting** combines multiple classifiers by simply taking a majority vote of the outputs from each classifier and using that as the final label. For regression tasks, the average is taken instead of voting.

- **Bagging** or bootstrap aggregating trains multiple classifiers of the same type on different random subsets of the training data and combines their outputs using averaging or majority voting, depending on the task. By training each classifier only on subsets of the data, this approach reduces variance, which reduces the chances of the model overfitting on the data. The Random Forests model described in Section 2.1.1.2 is a type of ensemble model that combines multiple Decision Tree classifiers and is one of the most commonly used ensembling techniques.

- **Boosting** is a method that iteratively trains a set of classifiers, with each new classifier focusing on classifying samples that were misclassified by the previous classifiers. **XGBoost** (Extreme Gradient Boosting)[19] is one of the best models of this class and is widely used in both academia and real-world applications due to its performance, scalability, and speed.

- **Stacking** is a way of ensembling models where the outputs from a set of classifiers are used as input features to another meta-classifier. This idea is very powerful as it lets us combine diverse models learning from different algorithms into a single robust system. This is primarily how we use ensembling in this thesis in Chapter 4, where we investigate how different kinds of models and features interact in an ensembling environment.

2.2 Sentiment Analysis Datasets

Several datasets for sentiment analysis have been created for Indian languages in the past few years. The earliest effort for SA in Indian languages was SAIL (Sentiment Analysis in Indian Languages), a shared task for sentiment analysis in three different languages - Hindi, Bengali, and Tamil [20]. More recently, the most prominent dataset for Hindi is the IITP Movie and Product Reviews Dataset [21]. It contains two datasets - the product reviews dataset containing 4509 product reviews and the movie reviews dataset - containing 2152 movie reviews. These are all classified into 4 categories - positive, negative, neutral and conflict (when both positive and negative sentiments are present in the sample). In Marathi, another close geographical and typological neighbour of Gujarati, the L3CubeMahaSent dataset [22] contains approximately 18k tweets labelled positive, negative and neutral. For Telugu, the ACTSA corpus [23] labelled 5.8k sentences extracted from news articles into the same three classes.

Figure 2.2: Countries and languages represented in the AfriSenti data collection (Amharic, Algerian Arabic, Hausa, Igbo, Kinyarwanda, Moroccan Arabic, Mozambican Portuguese, Nigerian Pidgin, Oromo, Swahili, Tigrinya, Twi, Xitsonga, and Yorùbá).

They also included annotator agreement information as well as conflict resolution schema in their annotation guidelines. Another significant Teluga SA resource is the Sentiraaama corpus [24], containing 4 datasets from different domains for over 1000 total documents with 46k sentences, classified into positive or negative sentiments. SentiNoB [25] is a dataset in Bengali sourced from social media comments containing 15k samples in three classes. BanglaBook[26] is another recent effort for SA in Bengali that contains 158k book reviews in Bengali annotated by the user score on the review. Apart from these, efforts have also been made in other languages like Tamil [27] and Malayalam [28].

In African languages, which are also significantly resource-poor, AfriSenti [29] is the main dataset that we experimented with. It contains over 110k tweets across 14 African languages from across the continent (refer Figure 2.2 [29]) classified into 3 classes, with significant variance in dataset sizes, class distribution and scripts. This made it the perfect dataset to experiment with, as it let us study the performance of our models across varying distributions of training data. The dataset is a compilation of new datasets for most of the languages and existing datasets for Amharic [30], and Hausa, Igbo, Yoruba, and Naija datasets, all from [31]. The data collection, filtering and annotation schema are also described in detail, which made them a good reference for creating our own dataset.

Chapter 3

GSAC: A Gujarati Sentiment Analysis Corpus from Twitter

As established in the Chapter 2, sentiment analysis as a task has received significant attention in recent years, with ever-increasing internet access and social media usage, even in Indian languages such as Hindi [21, 32] and Marathi [22] which are typologically and geographically close to Gujarati. However, there is hardly any work done in Gujarati itself.

Gujarati is a very prominent language in Western India, with over 55 million first-language speakers and a significant presence in the states of Gujarat, Maharashtra, and Rajasthan [33]. It is also the official language of the state of Gujarat. Despite a large online community active on social media and a significant mainstream media presence, there is a lack of large-scale, publicly available resources for sentiment classification (see Section 3.1).

Hence, we describe a new, gold-standard, manually annotated Gujarati Sentiment Analysis Corpus (GSAC) for monolingual sentiment classification. The dataset is sourced from Twitter and labelled by native speakers. We describe our annotation process and also run extensive experiments on the dataset using feature-based and deep-learning architectures to establish a reliable baseline for GSAC and compare the performances of various model architectures. The dataset is available on GitHub.[1]

Work(s)	Source	Size	Annotation	Open Access
Joshi and Vekariya(2017) [34]	Twitter	40	Manual	No
Mehta and Rajyagor(2021) [35]	Poems	300	Manual	No
Gohil and Patel(2019) [36]	Twitter	1120	Manual	No
Shah and Swaminarayan(2021) [37]	Movie Reviews	500	Manual	No
Shah and Swaminarayan(2022) [38], Shah et al.(2022) [39]	Movie Reviews (Gujarati + translated from English)	2085	Automated, based on website rating	No
GSAC	**Twitter**	**6575**	**Manual**	**Yes**

Table 3.1: Comparison of previous datasets on Gujarati Sentiment Analysis with our dataset - GSAC

[1] https://github.com/MG1800/gsac

3.1 Related Work

Significant work has been done on sentence-level and aspect-based sentiment analysis (SA) in various Indian languages. Datasets have been created for SA in Hindi [21, 32], Telugu [23], Marathi [40], Bengali [25, 32] and Tamil [27], and Tamil and Malyalam [41]. However, SA in Gujarati has been scarcely explored, and no standard, publicly available dataset exists.

One of the earliest works in SA in Gujarati was by Joshi and Vekariya [34], who used a POS tag-based feature set for an SVM classifier on a small sample of 40 tweets. Since then, Gohil and Patel [36] developed and experimented with a Gujarati SentiWordNet to classify tweets, creating a Twitter dataset with 1120 samples. Other approaches included scraping movie-review websites to create a dataset [37], even translating reviews from English to Gujarati to expand the dataset [38, 39]. Mehta and Rajyagor [35] attempted classifying a set of 300 poems into nine different emotional categories using machine learning-based approaches. However, none of the datasets used in these experiments have been released to open access, which makes it difficult to reproduce any of these results or compare the performance of new models with them.

Additionally, Gujarati was a part of the set of languages included in the training data for XLM-T [42], a highly multilingual effort for creating a unified Twitter-based language model for sentiment classification. However, Gujarati was not a part of the monolingual evaluation reported by the authors. Additionally, Gujarati has been included in some research on multilingual lexical level sentiment classification [43, 44].

Efforts in dataset creation for Sentiment Analysis have been varied. We mainly focused on Twitter datasets or datasets in Indian languages for reference when deciding our annotation process. Mukku and Mamidi [27] created a Twitter-based emotion classification dataset in Tamil and English and used a set of emotion words as queries for collecting tweets, an approach that we also use for collecting our data. [23] classify sentences from a news corpus into three sentiment categories - positive, negative, and neutral, similar to what we aim for, and hence are a good source of reference for annotation guidelines. We also refer to Muhammad et. al. [45], which is a more recent effort at creating a sentiment classification dataset for resource-poor languages, collecting and annotating a dataset for 4 African languages with multiple human annotators.

Table 3.1 compares our dataset to the existing SA datasets in Gujarati.

3.2 Dataset Creation

The dataset was created in two main steps - collecting and sampling the dataset from Twitter to create a subset for annotation and getting the data annotated by native speakers, which included creating the annotation guidelines and training them for the task.

3.2.1 Collection

We source our data from Twitter, which has a large active user base of Gujarati speakers. We scraped the initial dataset using Twitter API [2], which supports filtering the results for Gujarati using the language tag. We also used the API parameters to exclude retweets and quotes, to reduce the number of duplicates in our dataset. To ensure we had a desirable mix of sentiments in the dataset, the search queries were based on a hand-picked subset of sentiment words [3] based on a machine-translated English sentiment lexicon [46]. We chose a subset so as to remove words that were either not translated or translated incorrectly in the list, selecting ∼250 words. The start times are varied to ensure the tweets are spread out over time, with the final set having tweets ranging from August 2010 to February 2022. We then preprocessed, filtered, and sampled from this large dataset to generate subsets for each of our annotators to label. The complete process we followed is described below:

1. Create a list of prompts by hand-picking samples from machine-translated sentiment vocabulary.
2. Scrape tweets using these prompts using Twitter API, using the API parameters to ensure collected tweets are in Gujarati script, spread out over several years, and do not include any retweets or quotes.
3. Preprocess these tweets, normalising white-spaces and newlines, lower-casing, and replacing all user mentions and URLs with the tokens `@user` and `<url>` respectively.
4. Drop any tweets with identical text or fewer than 10 tokens after preprocessing. This step eliminated a significant number of gibberish tweets that were not useful for the task, such as the one shown in row 4 of Figure 3.1.
5. Randomly sampled 10% of the tweets for each prompt to create a subset of approximately 22,000 samples from the larger set that retained the same distribution as the original set.
6. From this smaller representative subset, we randomly sampled 7,000 tweets for annotation based on the annotation resources available to us.

The statistics for this process are provided in Table 3.2. We labelled approximately 7000 tweets from the representative set, with the final dataset containing 6,575 tweets after dropping undesirable samples as described in Section 3.2.2.

3.2.2 Annotation

We first developed the annotation schema and tested it by annotating a small sample of the dataset ourselves. Once the dataset was finalised, we recruited four annotators and trained them over several rounds of labelling and discussion before providing them with independent subsets to annotate.

[2] https://developer.twitter.com/en/docs/twitter-api

[3] https://www.kaggle.com/datasets/rtatman/sentiment-lexicons-for-81-languages

ID	Text	English Translation	Label
1475024518670286849	હર એક સવાર તમારા માટે નવો દિવસ લઈને આવે છે, ઉઠો અને તમારા સુંદર સ્વપ્ન પૂર્ણ કરવા દોડવા લાગો	Every morning brings a new day for you. Wake up and start running to finish your beautiful dreams	positive
1051082975377469440	આ જાહેરાત થી હિન્દુઓ ની લાગણી ને ઠેસ પહોંચાડી છે. @user માફી માંગે નહિતર, આ છાપા નો બહિષ્કાર કરીશું. #ધિક્કાર_છે_દિવ્યભાસ્કર @user @user @user <url>	this advertisement has caused harm to the feelings of hindus. @user ask for forgiveness or this newspaper will be boycotted. #divyabhaskar_is_hate @user @user @user <url>	negative
1412564898148724738	ટ્રકમાં દવાના બોક્સ નીચે સંતાડેલો 20.25 લાખનો દારૂનો જથ્થો કબજે <url>	Liquor stash worth 20.25 lakhs captured from being hidden inside medicine boxes of trucks <url>	neutral
1278888857199493129	@user વચન.. નમન.. કથન.. કઠણ.. રમણ.. વતન.. સરસ.. સરળ.. શરણ.. હરણ.. જતન.. ધમણ.. બરડ.. કડક.. શરત.. ખપત.. પવન.. પતન.. ફરજ..	@user promise.. bow.. statement.. hard.. ramana .. hometown.. nice.. easy.. refuge.. deer.. preservation.. a lot.. strength.. solid.. bet.. shortage.. wind.. downfall.. duty	unfit - random list of words
1336584570989387776	સેંસેક્સ 4600 ને પાર ફિર એક બાર મોદી ને કિયા ચમત્કાર 🤘 <url>	senex beyond 4600 once again Modi has done a miracle 🤘 <url>	unfit - Hindi typed in Gujarati script

Figure 3.1: Some samples from the GSAC dataset

3.2.2.1 Annotation Schema

We classified each tweet in our dataset as `positive`, `negative`, or `neutral`. We also gave our annotators an `unfit` tag for tweets that they think cannot be used for the task. We define each of the labels as follows:

- `positive` - Tweets were classified as positive if they expressed a positive sentiment about some subject (a product or a movie, for example) or if they showed support for a subject, such as a person or a policy. Tweets about events inherently associated with positive sentiments (such as reporting a sports team's victory) are also labelled positive.

- `negative` - Tweets that expressed a negative opinion about a subject (such as criticising a policy or an official) were labelled negative. Tweets talking about events with an inherently negative connotation - such as reporting the death of a celebrity or the loss of a sports team, and tweets containing any kind of derogatory remarks or threats towards a subject were also labelled negative.

- `neutral` - Tweets were labelled as neutral in two cases - if they contained no sentiment about the subject or if they contained a mix of both positive and negative sentiment about a subject (such as praising one aspect but criticising another of a product).

- `unfit` - Tweets were marked unfit if the annotator could not assign one of the three labels to it. This happened in several cases, such as cases where it was a different language tweet that was typed in Gujarati script, or there was not enough context in the tweet to label it (if it required a media attachment to understand, for example). Any tweets marked unfit by any of the annotators were dropped from the dataset.

Figure 3.1 illustrates some of the tweets and their labels, along with an approximate English translation of the tweet.

Stage	Count
Initial set from scraping	320,978
Filtering out duplicates	247,226
Dropping tweets with under 10 tokens	226,482
Representative Set after Sampling	22,630
Annotated	6,575

Table 3.2: Data collection statistics

3.2.2.2 Annotation Process

We manually annotated 7000 samples across four annotators. The annotators were linguistics students who were native speakers of Gujarati, aged between 19 and 23. The annotators were trained for the task over three rounds of annotation on small subsets of 50 tweets each, followed by a session of doubt clarification and discussion after every round. To measure the annotation quality, we calculate inter-annotator agreement using Fleiss' Kappa coefficient [47]. Over the three rounds of training, it improved from 0.48 to 0.52 and finally to 0.58, which suggests moderately strong agreement. The tweets used for these training rounds were discarded and not included in the final dataset. Each annotator then labelled data in subsets of 500 samples.

3.2.3 Statistics

Our final dataset contains a total of 6575 tweets after dropping the tweets labelled unfit. We divide the dataset into training, development, and test sets in a 70:10:20 ratio, respectively. Within the complete dataset, the `neutral` class has the highest representation, comprising about 45.12% of the total dataset, followed by `positive` at 30.05% and finally `negative` at 24.83%. Additional details about the class distribution are reported in Table 3.3.

The average word count for the combined dataset is 27.77, with a standard deviation of 13.86. The average word count (excluding whitespaces) is 136.07, with a standard deviation of 67.55, as shown in Table 3.4, which also reports the same values for each class. Figures 3.2 and 3.3 illustrate the class-wise and split-wise distribution of word counts in the dataset, respectively.

3.3 Experiments

We train two sets of models to test how different models perform on our dataset and to set baselines for it. The first set of models consists of feature vector-based models, of which we train two different variants based on different sets of features - Bag-of-Words and TF-IDF. The second set is a set of deep

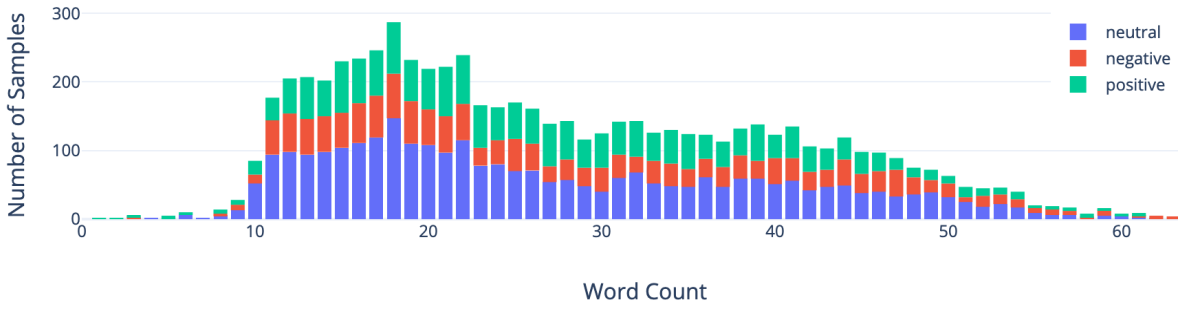

Figure 3.2: Class-wise distribution of word counts

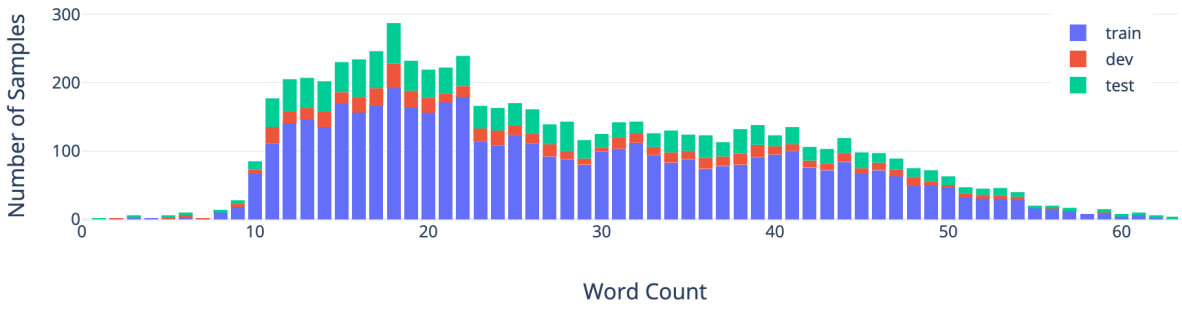

Figure 3.3: Split-wise distribution of word counts

contextualised models, where we fine-tune various transformer-based pre-trained language models for classification on this dataset.

3.3.1 Feature Vector Models

We train five classifiers - Naive Bayes, Logistic Regression, Support Vector Machines, Random Forests, and a Multi-Layer Perceptron - each on two different feature vectors - Bag-of-Words and TF-IDF for a total of 10 models.

Split	Positive	Neutral	Negative	Total Count
Train	1374	2100	1128	4602
Dev	201	287	163	651
Test	401	580	341	1322
Total	1976	2967	1632	6575

Table 3.3: Split-wise Class Distribution of Dataset

Split	Tokens	Characters
Positive	27.86 (11.65)	141.79 (58.92)
Neutral	27.25 (15.52)	132.39 (77.02)
Negative	28.60 (13.07)	135.62 (57.82)
Overall	**27.77 (13.86)**	**136.01 (67.55)**

Table 3.4: Mean Token and Character Counts for each label (brackets contain standard deviation)

Bag-of-Words (BoW) or Count Vectorizer represents a document (in this case, a tweet) as a vector of the counts of each word present in the document. Even though it ignores word order, bag-of-words features can still be useful as feature vectors for tasks such as text classification [48].

TF-IDF (Term Frequency - Inverse Document Frequency) [49] is a method to represent documents that factors in the relative frequency of a word across documents by calculating a score based on two parameters - term frequency, which is the frequency of a term in the current document, and inverse document frequency - which is based on the frequency of the term across all documents.

The models we train for each of these are:

- **Naive Bayes Classifier** - A Naive Bayes classifier is a simple classifier that estimates the probability of each label under the assumption of input features being conditionally independent, which has been shown to perform well on text classification [48]. We train the classifier for 200 epochs or until convergence.

- **Logistic Regression** - Logistic regression [50] is a classification algorithm that estimates a logistic function to calculate the probability of an input feature belonging to a certain class. We train an LR classifier over 100 epochs or convergence using a one-vs-all approach.

- **Support Vector Machine** - A support vector machine [51] is a classifier that tries to find the hyper-plane that most optimally divides the training data according to the labels. This is also trained using a one-vs-all approach, over 200 maximum epochs.

- **Random Forests** - Random Forests [52] are a type of ensemble classifier that use a large number of decision trees (set to 100 for our model), each using a subset of the input features and training data, to estimate the most likely label for the given input.

- **Multi-Layer Perceptron** is a simple feed forward neural network [53, 9]. Our model uses a single 100-dimension hidden layer, with a ReLU activation, for 300 maximum epochs.

We use the scikit-learn python library [54] to create feature vectors from the text and train and test this set of models.

3.3.2 Deep Contextualised Models

Multilingual transformer-based language models trained on multiple languages such as BERT [11] and RoBERTa [13] have been shown to perform well on downstream tasks [55]. We fine-tune the following language models on our dataset:

- **Multilingual BERT** - mBERT is a multilingual version of BERT [11], and is a language model trained on the top 100 languages with the largest Wikipedia corpora, which includes Gujarati. We use the `bert_base_multilingual_uncased` version of BERT.
- **XLM-RoBERTa** - XLM-RoBERTa is a multilingual version of RoBERTa [13], which is itself a more optimised version of BERT, trained on a larger dataset, and a modified training task. XLM-RoBERTa also includes Gujarati as a part of its training set. We fine-tune the `xlm-roberta-base` variant of the model.
- **XLM-T** - XLM-T [44]is a variant of XLM-RoBERTa that was trained on a Twitter dataset consisting of 198M tweets in a large set of languages, including over 10,000 samples in Gujarati. It was further finetuned for Sentiment Classification on a set of 8 languages, which included Hindi, which is closely related typologically to Gujarati. We further fine-tune the `twitter-xlm-roberta-base- sentiment` variant of the model on HuggingFace.
- **GujaratiBERT** - GujaratiBERT [56] is an mBERT (base variant) model that has been fine-tuned for Gujarati using publicly available monolingual Gujarati corpora. Since it is specifically fine-tuned for Gujarati, we expected it to perform better than mBERT and XLM-RoBERTa.
- **IndicBERT** - IndicBERT [15] is an ALBERT [14] model pre-trained on a combined corpus of 12 different Indian languages (including Gujarati), which has been shown to achieve state-of-the-art performance on multiple downstream tasks in several Indian languages on the IndicGLUE benchmark [15], including sentiment analysis in Hindi [21] and Telugu [23]. We fine-tune this model for classification on our dataset.

All of our transformer models are trained for 5 epochs, with a learning rate of 4e-5 and batch size of 8. We set up our training and testing scripts using the simpletransformers [57] library, which is based on the transformers library from HuggingFace [58].

3.4 Results

We report the detailed results for each model in Table 3.5. We make a few observations from observing the weighted and macro F1 scores for each model:

- We observe that GujaratiBERT and IndicBERT achieve the best performance compared to all other models. This could be because compared to the rest of the pretrained language models,

Model	Precision	Recall	Accuracy	Weighted F1	Macro F1
Bag of Words					
Naive Bayes	0.59	**0.58**	**0.58**	**0.57**	**0.56**
Logistic Regression	0.55	0.55	0.55	0.55	0.54
SVM	0.55	0.52	0.52	0.49	0.46
Random Forests	**0.61**	**0.58**	**0.58**	0.55	0.53
MLP	0.52	0.52	0.52	0.52	0.51
TF-IDF					
Naive Bayes	**0.66**	0.52	0.52	0.43	0.38
Logistic Regression	0.58	**0.57**	**0.57**	**0.56**	**0.55**
SVM	0.57	0.56	0.56	0.54	0.53
Random Forests	0.59	0.55	0.52	0.50	0.50
MLP	0.50	0.50	0.50	0.50	0.49
Pretrained LMs					
mBERT	0.38	0.51	0.51	0.43	0.38
XLM-RoBERTa	0.41	0.52	0.52	0.43	0.39
XLM-T	0.64	0.62	0.63	0.64	0.63
GujaratiBERT	0.64	0.64	0.64	0.64	0.64
IndicBERT	**0.65**	**0.67**	**0.66**	**0.66**	**0.66**

Table 3.5: Results of the various models on the test set. **Bold** indicates best score for each set of models. **Underline** indicates best score across all models.

these two models have been trained on a significantly higher amount of Gujarati data (during pretraining for IndicBERT, and during fine-tuning for GujaratiBERT).

- mBERT and XLM-RoBERTa perform very poorly compared to other pretrained language models. This could be because they are trained on a very large set of languages, due to which Gujarati might not have sufficient representation in the corpus and the model vocabulary causing it to underperform.

- XLM-T contained only ∼10,000 samples in Gujarati out of a total ∼198M samples in its training data. However, it still achieves comparable performance to GujaratiBERT and IndicBERT. This may be because the training data for XLM-T comes exclusively from the same domain as our

dataset (Twitter), which suggests pretraining or fine-tuning models on similar domain data in multiple languages can help improve model performance in low-resource languages.

- Despite not achieving the same performance as XLM-T, GujaratiBERT, or IndicBERT, the Naive Bayes model using TF-IDF features achieves the highest precision out of all the models trained. Other statistical models (such as Random Forests and Naive Bayes on both feature sets) also achieve reasonably high average precision (¿= 0.59) while taking significantly less computational resources and time.

3.5 Ethical Consideration

Sentiments in a dataset sourced from social media platforms can be susceptible to inherent bias due to public opinion being biased in favour of or against certain subjects, depending on external factors like demographics. During the collection and annotation process for our dataset, we switched our collection strategy from querying tweets for particular topics (events) during the initial stages to querying them using a sentiment lexicon because we observed that the topics we queried were frequently heavily biased towards either positive or negative sentiments. The privacy of platform users is another concern that is raised when collecting data from social media. To ensure that no identifying details about any Twitter user were presented to our annotators, we removed any identifying characteristics such as user mentions and URLs from the tweets, as well as the original Tweet IDs and used internally generated IDs for the annotation process. We also only release the Tweet IDs and corresponding labels in our dataset in compliance with Twitter's data-sharing policy.

Chapter 4

Ensemble Learning for Sentiment Analysis in African Languages

While Chapter 3 dealt with creating a resource for a language with almost no resources for sentiment analysis, in this chapter we discuss experiments we carried out using the AfriSenti dataset [29] for the AfriSenti Shared Task[59] at SemEval-2023.

The AfriSenti Shared Task [59] was aimed at promoting sentiment classification research in a diverse group of African Languages. Though sentiment classification is an extremely popular task in NLP, there has been relatively little work done for it in African Languages. Apart from the NaijaSenti dataset [31] (which was a part of the task itself), some of the languages where such work has been done include Amharic [30], Tunisian Arabizi [60], and Swahili [61]. The task itself was divided into three sub-tasks - monolingual classification in 12 languages separately, multilingual classification, and zero-shot classification for two languages.

We participated in the monolingual and multilingual tracks of the task. Our system consists of an ensemble model that leverages emoticon frequencies and the predictions of contextualised, transformer-based models such as AfriBERTa [62] and AfroXLMR [63], as well as simpler models such as Logistic Regression, SVM and Random Forest classifiers. We also release the code on GitHub.[1]

Our model achieves a significant range of ranks, with our best result being ranked 7th each in the Igbo, Twi, and Yoruba tracks and between 12 and 15 on the multilingual, Hausa and Xitsonga tracks. The full rankings are reported in Table 4.2.

We report observations on the performances of our individual models that we used across languages and the performance of languages across models. Additionally, we report the results for some of the experiments we carried out that did not make it to the final model.

[1] https://github.com/MG1800/afrisenti-ensemble

Language	Train				Dev				Test			
	Total	**positive**	**negative**	**neutral**	Total	**positive**	**negative**	**neutral**	Total	**positive**	**negative**	**neutral**
am	5985	22.26%	25.87%	51.87%	1498	22.24%	25.92%	51.84%	2000	21.91%	66.88%	11.21%
dz	1652	25.26%	54.03%	20.71%	415	25.36%	53.86%	20.77%	959	34.34%	49.58%	16.08%
ha	14173	33.07%	32.27%	34.66%	2678	33.13%	33.40%	33.47%	5304	33.09%	33.17%	33.74%
ig	10193	30.26%	25.51%	44.23%	1842	30.42%	25.53%	44.05%	3683	30.36%	25.61%	44.02%
kr	3303	27.23%	34.71%	38.07%	828	27.21%	34.70%	38.09%	1027	27.10%	34.60%	38.30%
ma	5584	31.49%	29.80%	38.71%	1216	31.69%	29.63%	38.68%	2962	38.64%	28.71%	32.66%
pcm	5122	35.31%	63.29%	1.41%	1282	34.89%	63.47%	1.64%	4155	33.63%	55.99%	10.38%
pt	3064	22.23%	25.53%	52.24%	768	22.29%	25.55%	52.15%	3663	17.15%	17.89%	64.96%
sw	1811	30.22%	10.55%	59.23%	454	30.24%	10.60%	59.16%	749	29.95%	10.70%	59.36%
ts	805	47.76%	35.32%	16.92%	204	47.29%	35.47%	17.24%	255	47.64%	35.43%	16.93%
twi	3482	47.23%	37.78%	15.00%	389	47.16%	37.89%	14.95%	950	47.42%	37.20%	15.38%
yo	8523	41.56%	21.97%	36.47%	2091	42.30%	21.20%	36.51%	4516	42.48%	21.73%	35.79%
multilingual	63697	32.63%	31.57%	35.79%	13665	32.32%	31.80%	35.88%	30223	32.44%	33.78%	33.79%

Table 4.1: Split-wise Label Distribution of the Datasets

4.1 Background

4.1.1 Task Description

The objective of the task [59] is to identify the polarity of a tweet (negative, positive, or neutral) in a set of 14 African languages. It is divided into three sub-tasks.

Sub-Task A is for monolingual classification systems, where a separate classifier is trained for each language and has a separate track for each of the 12 languages in this sub-task. These languages are Hausa, Yoruba, Igbo, Nigerian Pidgin, Amharic, Algerian Arabic, Moroccan Arabic(Darija), Swahili, Kinyarwanda, Twi, Mozambican Portuguese, and Xitsonga(Mozambique Dialect). Sub Task B consists of training a single multilingual model to classify tweets in all of the languages in Sub Task A. Sub Task C is for zero-shot classification in 2 languages (Tigrinya and Oromo), for which training data is not available, and only a development set is provided.

We are participating in all the tracks for Sub Task A and in Sub Task B.

4.1.2 Dataset

The AfriSenti dataset [29], provided by the organisers, consists of a manually annotated corpus of tweets in each of the 12 languages. The multilingual dataset is created by combining all of the individual datasets. Each sample in the dataset consists of an ID, the text of the tweet, and a label. The label can be ”positive”, ”neutral”, or ”negative”. The organisers have already divided the datasets into train, dev and test splits. We notice that there is a considerable amount of variation in the label distribution between

the languages. While some languages like Hausa (ha) and Darija (ma) have an almost perfectly equal representation of each of the three classes, others have a significant class imbalance in the dataset, with Nigerian Pidgin (pcm) being the biggest standout with less than 2% of the samples labelled neutral. Three other languages (Xitsonga (ts), Twi (twi), and Swahili (sw)) have one of the classes at less than 20% representation. The detailed split-wise label distribution for the datasets is given in Table 4.1.

Nine of the 12 languages in the dataset are in Latin script. Of the remaining three, Amharic is written in Ge'ez script, Algerian Arabic in Arabic script, and Darija in both Latin and Arabic scripts. The tweets are also code-mixed with English.

Apart from the details provided in [29], [31] also described the collection and annotation process for Hauso, Igbo, Naija (Nigerian Pidgin) and Yoruba. The Amharic dataset is described by [30].

4.1.3 Sentiment Classification

Large, multilingual Pretrained Language Models (PLMs) such as multilingual BERT [11] and RoBERTa [13] and their derivatives, which have been trained on massive corpora covering 100+ languages have been shown to perform well on downstream tasks for resource-poor languages [55], including on sentiment classification in languages such as Arabic [64] and Swahili [61].

For African languages, AfriBERTa [62] (trained on 11 languages) and AfroXLMR [63] have shown state-of-the-art performance on various downstream tasks. This includes sentiment classification on the NaijaSenti dataset for Hausa, Igbo, Yoruba, and Nigerian Pidgin [31].

Ensemble learning [18] is another popular method used to make classification models more robust [65]. An ensemble leverages the fact that different classifiers leverage different characteristics of the input during prediction. They combine the predictions of a set of diverse models to produce a more robust output. [66] show specifically that combining feature-based classifiers with deep, neural network-based classifiers can increase model performance across a variety of datasets.

We use a popular ensembling method, XGBoost [19], to combine the predictions from transformer-based classifiers as well as lightweight, feature-based classification algorithms such as SVM [51], Random Forest [52] and Logistic Regression [50]. We also incorporate emoticon frequencies into the feature vector for our final ensemble learning model.

4.2 System Overview

We train an ensemble model for each language that takes in a feature vector consisting of the output of six classification models and emoticon count features for each tweet.

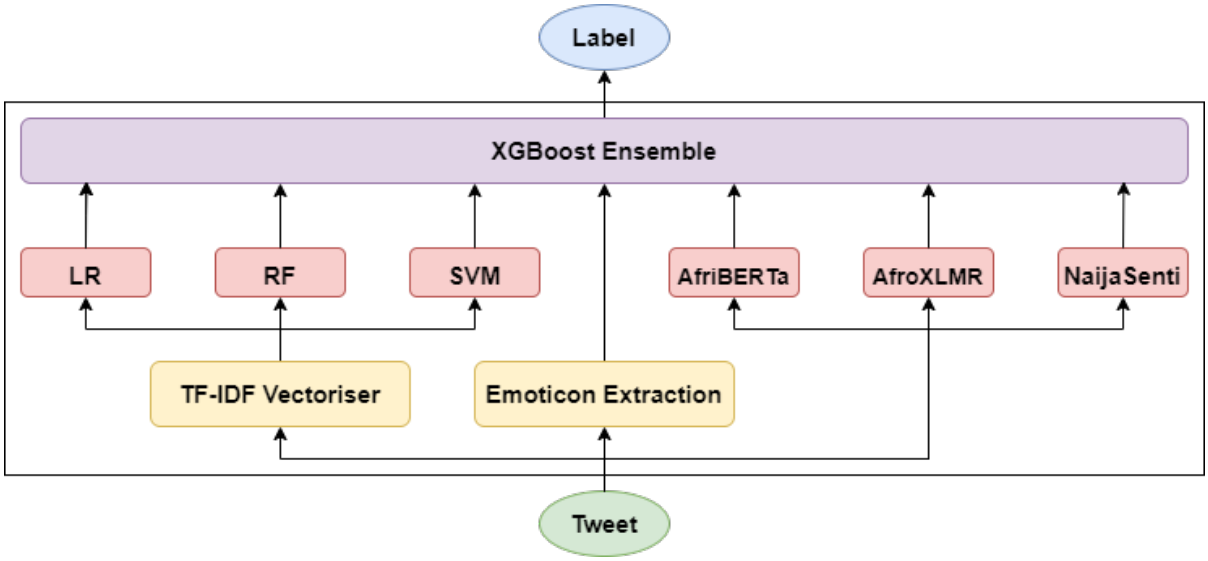

Figure 4.1: System Overview

4.2.1 Statistical models

For each language, we train three statistical models, and we obtain the probability of each class to use as features for our ensemble. We use term frequency-inverse document frequency (tf-idf) as input features to our classifiers. The classifiers we train are -

- **Logistic Regression (LR)** - we train a multi-class logistic regression [50] classifier using a one vs rest approach. We limit the model to 1000 iterations or convergence.
- **Support Vector Machine (SVM)** - An SVM [51] is a binary classifier that tries to find a hyperplane that most accurately divides the training data into two classes. We train an SVM for multi-class classification using a one-vs-rest approach and a linear kernel.
- **Random Forests (RF)** - A Random Forest [52] is a type of ensemble classifier that uses a large number of decision trees that are each trained using a different subset of the input features and a subset of the training dataset (with replacement), which are then combined using a bagging approach. We train our RF classifier using 100 such decision trees.

4.2.2 Transformer-based models

We train three different transformer models for each language and obtain the output score for each class for all the samples in the data. These scores are then used to create the feature vector for the ensemble. We use the following models -

- **AfriBERTa** - AfriBERTa [62] is an XLM-RoBERTa [13] based language model that was trained on 1 GB of text data in 11 African languages, including 6 of the 12 languages in the task. It was

shown to outperform mBERT and XLM-RoBERTa on various tasks, including text classification. We use the AfriBERTa-base variant that consists of 111M parameters.

- **AfroXLMR** - AfroXLMR[63] is another language model based on XLM-RoBERTa that was trained using 17 African languages - of which 7 are a part of the task - and three high resource languages (English, French, Arabic). The model was shown to be competitive with existing models and improve zero-shot classification for unseen languages in some tasks. We use the AfroXLMR-base variant for our system.

- **AfriBERTa-NaijaSenti** - This model is a multilingual classification model based on AfriBERTa-large that achieved the best scores on the original NaijaSenti dataset [31], which is a part of the corpus for this task. We further fine-tune this model for each language using the respective training datasets.

4.2.3 Ensemble Classifier

Each model may learn different characteristics of the data towards the training task depending on the model architecture, training objective, and any pre-trained weights used. This can lead to different model capabilities, which can be leveraged by applying an ensemble over the individual model predictions. For this, we train an XGBoost [19] classifier on the train predictions of the statistical and transformer-based prediction probabilities for each sentiment class.

Emoticons are an integral part of online text-based communication and can significantly impact the tone and overall sentiment of the text. Thus, for each language, we identify the frequently occurring emoticons (present in at least 10% of the training samples). The respective frequencies of these emoticons are generated for all data samples and used as features alongside the individual model predictions in the ensemble model. Note that in case there is no emoticon that is present in at least 10% of the training samples, emoticon features are not added to the feature, and only the classification scores are used. Figure 4.1 illustrates the top-level view of our final system.

The final model is an XGBoost classifier that trains on a feature vector consisting of the individual class probabilities of the underlying classifiers, along with the frequencies of emoticons that appeared in at least 10% of the training samples. Specifically, it consisted of:

- class probabilities from tfidf-based models: the three statistical models output the probability of the input sample belonging to each class (i.e. 3 probabilities each). We take these as features in our feature vector (combined 9 features for 3 classifiers and 3 classes)

- the fine-tuned LMs output an a score for each class in the range of (-1, 1). We softmax these scores to obtain the probability of each class to keep these features comparable to the outputs of the tf-idf based models. We again have 9 features for this class of models, with 3 features from each of the 3 models.

Language	Rank	Score
Amharic (am)	25	39.09%
Algerian Arabic (dz)	25	57.55%
Hausa (ha)	15	79.65%
Igbo (ig)	7	80.87%
Kinyarwanda (kr)	22	62.69%
Darija (ma)	27	50.68%
Nigerian Pidgin (pcm)	26	64.44%
Mozambican Portuguese (ma)	22	65.02%
Swahili (sw)	19	58.91%
Xitsonga (ts)	14	52.82%
Twi (twi)	7	66.47%
Yoruba (yo)	7	78.44%
Multilingual	12	68.84%

Table 4.2: Ranks and weighted-F1 scores for our system submission

- The number of emoticon features varies by the language, ranging between 0 to 4 depending on how many emoticons appear in at least 10% of the data. Note that in case of certain languages where there were no or very few emoticons present in the dataset, the final feature vector does not include these features.

Further details about the experimental setup are clarified in 4.3.

4.3 Experimental Details

For each language, we only use the train set provided by the organisers during the training of our models. The development set was used as an unseen set to compare the final performance of the various models that we trained during the competition.

All three of our transformer models are trained for five epochs on a Kaggle kernel with an Nvidia P100 GPU with 16 GB VRAM and 13 GB of RAM. We make use of the simpletransformers library [57], which is based on the HuggingFace Transfomers library[58].

For our statistical models, we use the implementations provided by scikit-learn [54] for both tf-idf feature extraction, training our classifiers, and evaluating them. For logistic regression, maximum iterations were set to 1000 epochs, and for the random forest classifier, we set the number of decision

trees to 100. Our SVM models were trained with a linear kernel. To ensure uniformity, we took the probability of each class as the feature to our final ensemble model.

The ensemble XGBoost model was trained with a learning rate of 10^{-6} for a pool of 100 estimators, i.e. trees, with a max tree depth of 32 and 10% of the columns sampled for each tree.

Language	AfroXLMR	AfriBERTa	NaijaSenti	SVM	LR	RF	Ensemble
am	54.07%	51.57%	58.39%	33.12%	29.01%	27.85%	39.09%
dz	66.08%	47.03%	42.48%	54.34%	53.48%	54.95%	57.55%
ha	78.01%	78.46%	80.53%	73.14%	72.74%	69.89%	79.65%
ig	78.13%	79.04%	80.59%	77.69%	77.22%	74.71%	80.87%
kr	66.23%	63.25%	62.20%	57.87%	56.15%	54.31%	62.69%
ma	48.23%	41.76%	41.13%	56.59%	53.17%	46.43%	50.68%
pcm	66.70%	62.40%	68.66%	60.03%	60.79%	57.05%	64.44%
pt	68.66%	58.56%	59.48%	62.14%	61.92%	59.90%	65.02%
sw	62.00%	62.35%	60.30%	55.30%	53.67%	50.74%	58.91%
ts	39.09%	54.44%	52.87%	49.73%	49.07%	47.36%	52.82%
twi	63.00%	65.42%	64.91%	62.17%	61.90%	63.32%	66.47%
yo	69.38%	73.48%	79.29%	72.55%	72.07%	65.10%	78.44%
multilingual	68.48%	64.32%	66.74%	64.45%	64.04%	58.80%	68.84%

Table 4.3: Weighted F1 scores for each language and model trained for the task on the test set. The scores for the individual models were calculated after the release of the test set by us, while the scores for the ensemble (also on the same test set) were taken directly from the competition website.

4.4 Results and Analysis

The official rankings for the task are based on the weighted f1 scores for our systems on the test sets. Our system achieves a mixed range of results across the different tracks that we participated in. It ranks 7th on the Igbo, Twi, and Yoruba tracks, which are our best results. Apart from that, it also ranks between 12 and 15 on the multilingual, Xitsonga and Hausa tracks. The complete rankings and scores are detailed in Table 4.2. Apart from the final system, we also report weighted f1 scores for each of the individual models that we had trained, listed in Table 4.3.

We observe from the class distribution of the datasets from Table 4.1 and our ranks that a greater imbalance in the class distribution of the training set appears to negatively affect our system compared to other systems.

Comparing the performance of our models across languages, we found a significant correlation between the performance of AfriBERTa and NaijaSenti, with a Spearman's correlation of 0.96 (p-value ¡ 0.0001). This is along expected lines since NaijaSenti is based on AfriBERTa. Along similar lines, AfroXLMR was seen to perform significantly differently from these two models, with a Spearman's correlation of 0.77 (p-value ¡ 0.01) with AfriBERTa and 0.82 (p-value ¡ 0.001) with NaijaSenti. These were even lower than its correlation with the statistical models, with correlation values ranging from 0.89 to 0.92 (p-values ¡ 0.0001).

Comparing the performance of different languages across the models, we notice that there is an observable similarity between the results in Hausa (ha), Igbo (ig), and Yoruba (yo). The Spearman's correlations for pairs of these languages ranges between 0.89 to 0.96, with p-values ¡ 0.01. This could potentially be because these three languages have the largest amount of training data and are also part of the training corpus for all three of the transformer models.

Additionally, the pairs Kinyarwanda (kr)-Swahili (sw) and Nigerian Pidgin (pcm)-Amharic both have a correlation of 0.93 (p-value ¡ 0.01). This could be possibly due to typological similarities between the languages, which need to be investigated further.

We also conducted an ablation study by changing the configuration of the ensemble classifier to exclude certain features after the competition once the test set labels were released. We varied the configuration to either include (+) or exclude − the emoticon features (`EMO`), and the outputs of the transformer models (`TR`) or the statistical models (`ST`). `+EMO+TR+ST` is the configuration we submitted for the competition.

We make a few observations on the basis of this set of results:

- Overall, ensembling only the transformer-based models seems to outperform all other configurations for most languages.
- Only in the case of Darija (ma), the ensemble of the statistical models outperforms those containing transformer models. This may be because of the fact that Darija is present in a mix of both Latin and Arabic scripts, while all the others are present in single script. However, further analysis is required to confirm this hypothesis.
- Since emoticon features are only generated if there is at least 1 emoticon present in at least 10% of the training data, some of the languages (am, pcm, sw) where there are not enough emoticons available perform identically whether we include that feature or not.
- Ensembles without emoticon features frequently outperform ensembles with them. While we still do believe emoticons can contribute important information to the models, we believe the lower scores may be due to the distribution of emoticons within these datasets, and further investigation into emoticon distribution and its effects on model predictions is required.

- Ensembles without emoticon features frequently outperform ensembles with them. Although we believe emoticons do contribute important information, the lower scores may be due to a sparsity of emoticons in the dataset, resulting in the recall being low. For example, the presence of a ”positive” emoticon may be highly indicative of the overall sentiment in the tweet being positive (high precision), but its absence is not enough to ascertain that the tweet is not positive (low recall).

The weighted F1 scores for these models are reported in Table 4.4.

Language	**+EMO**			**-EMO**		
	+TR	+ST	+TR+ST*	+TR	+ST	+TR+ST
am	**46.51%**	38.26%	39.09%	**46.51%**	38.26%	39.09%
dz	39.92%	53.84%	**57.51%**	54.48%	54.93%	**58.37%**
ha	**81.60%**	72.17%	79.65%	**81.87%**	72.75%	79.32%
ig	79.94%	76.36%	**80.87%**	**81.34%**	77.91%	80.74%
kr	**65.84%**	56.46%	62.69%	**65.81%**	56.60%	61.64%
ma	43.91%	**53.25%**	50.68%	44.05%	**53.61%**	50.99%
pcm	**66.02%**	62.19%	64.44%	**66.02%**	62.19%	64.44%
pt	**65.19%**	60.63%	65.02%	**65.82%**	63.11%	65.28%
sw	**62.29%**	55.20%	58.91%	**62.29%**	55.20%	58.91%
ts	**53.10%**	51.03%	52.82%	**53.51%**	52.24%	52.24%
twi	63.99%	62.86%	**66.47%**	67.01%	64.15%	**66.78%**
yo	78.38%	72.90%	**78.44%**	78.38%	72.90%	**78.44%**
multilingual	**68.69%**	64.31%	68.84%	**69.21%**	63.96%	68.84%

Table 4.4: Ablation study of different configurations of the ensemble model. Scores reported are Weighted F1 on the test set. * - Configuration that was submitted for the competition. **Underlined** indicates the best-performing model for the language across classes. **Bold** indicates the best-performing model for a language for that class of models (based on +EMO/-EMO)

4.5 Additional experiments

During the development phase of the competition, we tried out several experiments that did not make it to the final submission. This section discusses the motivation behind some of them.

Most of the tweets included in the dataset were code mixed with English. Therefore, we experimented with **replacing all emoticons with the corresponding English text** (such as replacing a smiley face with the token "smiling face"). We expected that this would help our models learn better by removing the emoticons from the vocabulary and increasing the frequency of their corresponding sentiment words. We tried doing this with AfriBERTa on Hausa and Igbo since they have the largest datasets available, and Igbo has a greater class imbalance in the training set than Hausa. However, the weighted f1 score fell from 79.88 to 77.64 for Hausa and from 80.33 to 79.12 for Igbo, so we decided not to continue with it.

We noticed that some of the models we trained were inconsistent with the neutral class, especially when there was a noticeable imbalance in the training set. To tackle this, we trained an AfriBERTa model specifically to distinguish between polar and non-polar tweets by **replacing the "positive" and "negative" labels in the dataset with a "non-neutral" label**. We expected this model to noticeably improve the classification on the neutral class and use that class score as a feature for the ensemble. However, this model showed no improvement in the f1 score for the neutral class in Igbo(0.79 f1 score for the neutral class in both cases) and scored lower in Hausa (0.74 f1 score compared to 0.78 earlier).

Finally, we also experimented with **combining the datasets for Algerian Arabic and Darija** since they are both variants of Arabic. We transliterated the datasets from Arabic script to Latin script (since Algerian Arabic was in Arabic script while Darija had a mix of both Latin and Arabic) using the Buckwalter system [67]. However, we did not notice any performance improvements in the predictions for either Algerian Arabic or Darija and decided not to pursue this experiment further.

4.6 Ethical Considerations

A sentiment classification model has significant potential use for online community management tasks such as forum moderation on social media platforms. If used without exhaustive evaluation and testing under different scenarios, it can cause significant damage, such as propagating any biases within the model. Even if the model is unbiased and robust, it can be used as a tool of suppression to identify and target individuals with specific viewpoints (such as their opinion of a particular organisation). Hence, developing a robust test for checking inherent biases is extremely important, as is exhaustive moderation and control over where and for what purposes such models are being deployed.

Chapter 5

Conclusion and Future Work

This thesis explores the field of sentiment classification, with a focus on languages suffering from a scarcity of resources. We explore tackling this challenge in two ways - by creating a new resource in a language which did not have any, and by exploring modelling techniques that could leverage available resources better.

In Chapter 3, we presented the Gujarati Sentiment Analysis Corpus (GSAC), which contains over 6500 manually annotated tweets. To the best of our knowledge, it is the first significant publicly available corpus for this task in Gujarati. We also present our annotation schema and conduct extensive experimentation to establish baselines for this new dataset. We find that pre-trained language models that included Gujarati as a part of pre-training or fine-tuning achieve better performance on this dataset compared to other models, with IndicBERT achieving the best weighted and macro F1 scores.

Continuing this effort, we plan to explore methods to extend this dataset automatically by using this dataset as a seed dataset to label additional data (such as by bootstrapping) or by exploring other avenues of acquiring data, such as via machine translation of existing datasets in other languages such as English or Hindi. Additionally, with the rise of multilingual Large Language Models recently, we would also like to explore using these models to synthetically generate more training data or help annotate existing unlabelled data through prompt engineering, as a means of creating additional resources.

In Chapter 4, we describe our system submission for the AfriSenti shared task at Semeval-2023. We combine the predictions of three transformer-based classifiers and three statistical ones and add emoticon frequencies to construct a feature vector for an XGBoost-based ensemble model. Though the system achieves mixed results in the rankings, we analyse the performance of each of the individual models to show a significant correlation between two of our transformer models and between several language pairs. We also describe additional experiments that we conducted which were not incorporated into the submitted system but gave valuable insights. These included replacing emoticons with text, combining two of the three classes to train a classifier to specifically distinguish the third class, and combining multiple datasets of similar languages to try and increase the training data. We also performed an ablation study of various model architectures to examine the interaction between various types of models and features.

We believe the additional investigation into combining datasets from different but related languages (such as leveraging Arabic resources for Darija and Algerian Arabic) could lead to more robust models. We would also like to investigate the correlated language pairs for linguistic and typological features that could potentially explain that observation. We also believe a similar approach could be explored in Indian languages, with datasets available in various languages which are both geographically and linguistically diverse in nature.

Related Publications

1. **Monil Gokani** and Radhika Mamidi. 2023. GSAC: A Gujarati Sentiment Analysis Corpus from Twitter. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 129–137, Toronto, Canada. Association for Computational Linguistics.

2. **Monil Gokani**, K V Aditya Srivatsa, and Radhika Mamidi. 2023. Witcherses at SemEval-2023 Task 12: Ensemble Learning for African Sentiment Analysis. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 357–364, Toronto, Canada. Association for Computational Linguistics.

Other Publications

1. K V Aditya Srivatsa, **Monil Gokani**, and Manish Shrivastava. 2021. SimpleNER Sentence Simplification System for GEM 2021. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 155–160, Online. Association for Computational Linguistics.

Bibliography

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[2] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Franz A. Heinsen. An algorithm for routing vectors in sequences, 2022.

[4] Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. Entailment as few-shot learner, 2021.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[9] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[13] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019.

[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.

[15] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*, 2020.

[16] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[17] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

[18] Zhi-Hua Zhou. *Ensemble Learning*, pages 270–273. Springer US, Boston, MA, 2009.

[19] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[20] Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. Shared task on sentiment analysis in indian languages (sail) tweets - an overview. In Rajendra Prasath, Anil Kumar Vuppala, and T. Kathirvalavakumar, editors, *Mining Intelligence and Knowledge Exploration*, pages 650–655, Cham, 2015. Springer International Publishing.

[21] Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[22] Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. L3CubeMahaSent: A Marathi tweet-based sentiment analysis dataset. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 213–220, Online, April 2021. Association for Computational Linguistics.

[23] Sandeep Sricharan Mukku and Radhika Mamidi. ACTSA: Annotated corpus for Telugu sentiment analysis. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 54–58, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[24] Rama Rohit Reddy Gangula and Radhika Mamidi. Resource creation towards automated sentiment analysis in Telugu (a low resource language) and integrating multiple domain sources to enhance sentiment prediction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[25] Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[26] Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. BanglaBook: A large-scale Bangla dataset for sentiment analysis from book reviews. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1237–1247, Toronto, Canada, July 2023. Association for Computational Linguistics.

[27] Rajenthiran Jenarthanan, Yasas Senarath, and Uthayasanker Thayasivam. Actsea: Annotated corpus for tamil & sinhala emotion analysis. In *2019 Moratuwa Engineering Research Conference (MERCon)*, pages 49–53, 2019.

[28] Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. A sentiment analysis dataset for code-mixed malayalam-english. *CoRR*, abs/2006.00210, 2020.

[29] Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif M. Mohammad, Sebastian Ruder, Oumaima Hourrane, Pavel Brazdil, Felermino Dário Mário António Ali, Davis David, Salomey Osei, Bello Shehu Bello, Falalu Ibrahim, Tajuddeen Gwadabe, Samuel Rutunda, Tadesse Belay, Wendimu Baye Messelle, Hailu Beshada Balcha, Sisay Adugna Chala, Hagos Tesfahun Gebremichael, Bernard Opoku, and Steven Arthur. AfriSenti: A Twitter Sentiment Analysis Benchmark for African Languages, 2023.

[30] Seid Muhie Yimam, Hizkiel Mitiku Alemayehu, Abinew Ayele, and Chris Biemann. Exploring Amharic sentiment analysis from social media texts: Building annotation tools and classification models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1048–1060, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[31] Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Sa'id Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Salahudeen Abdullahi, Anuoluwapo Aremu, Alípio Jorge, and Pavel Brazdil. NaijaSenti: A Nigerian Twitter sentiment corpus for multilingual sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 590–602, Marseille, France, June 2022. European Language Resources Association.

[32] Braja Gopal Patra, Dipankar Das, and Amitava Das. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task at @icon-2017. 2018.

[33] David M. Eberhard, Gary F. Simons, and Charles D. Fennig. *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, 26 edition, 2023.

[34] Vrunda C Joshi and Vipul M Vekariya. An approach to sentiment analysis on gujarati tweets. *Advances in Computational Sciences and Technology*, 10(5):1487–1493, 2017.

[35] Bhavin Mehta and Bhargav Rajyagor. Gujarati poetry classification based on emotions using deep learning. 2021.

[36] Lata Gohil and Dharmendra Patel. A sentiment analysis of gujarati text using gujarati senti word net. *International Journal of Innovative Technology and Exploring Engineering*, 2019.

[37] Parita Vishal Shah and Priya Swaminarayan. Lexicon-based sentiment analysis on movie review in the gujarati language. *Int. J. Inf. Technol. Commun. Convergence*, 2021.

[38] Parita Shah and Priya Swaminarayan. Machine learning-based sentiment analysis of gujarati reviews. *International Journal of Data Analysis Techniques and Strategies*, 2022.

[39] Parita Shah, Priya Swaminarayan, and Maitri Patel. Sentiment analysis on film review in gujarati language using machine learning. *International Journal of Electrical and Computer Engineering*, 12(1):1030, 2022.

[40] Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. L3cubemahasent: A marathi tweet-based sentiment analysis dataset. *CoRR*, abs/2103.11408, 2021.

[41] Bharathi Raja Chakravarthi, KP Soman, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Kingston Pal Thamburaj, John P McCrae, et al. Dravidianmultimodality: A dataset for multimodal sentiment analysis in tamil and malayalam. *arXiv preprint arXiv:2106.04853*, 2021.

[42] Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France, June 2022. European Language Resources Association.

[43] Mengjie Zhao and Hinrich Schütze. A multilingual BPE embedding space for universal sentiment lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3506–3517, Florence, Italy, July 2019. Association for Computational Linguistics.

[44] Sven Buechel, Susanna Rücker, and Udo Hahn. Learning and evaluating emotion lexicons for 91 languages. *CoRR*, abs/2005.05672, 2020.

[45] Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Said Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Abdullahi Salahudeen, Aremu Anuoluwapo, Alípio Jeorge, and Pavel Brazdil. Naijasenti: A nigerian twitter sentiment corpus for multilingual sentiment analysis. *CoRR*, abs/2201.08277, 2022.

[46] Yanqing Chen and Steven Skiena. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–389, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[47] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychol. Bull.*, 76(5):378–382, November 1971.

[48] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. *Work Learn Text Categ*, 752, 05 2001.

[49] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[50] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[51] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[52] Leo Breiman. *Machine Learning*, 45(1):5–32, 2001.

[53] F Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, November 1958.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[55] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics.

[56] Raviraj Joshi. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*, 2022.

[57] T. C. Rajapakse. Simple transformers. `https://github.com/ThilinaRajapakse/simpletransformers`, 2019.

[58] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[59] Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Seid Muhie Yimam, David Ifeoluwa Adelani, Ibrahim Sa'id Ahmad, Nedjma Ousidhoum, Abinew Ali Ayele, Saif M. Mohammad, Meriem Beloucif, and Sebastian Ruder. SemEval-2023 Task 12: Sentiment Analysis for African Languages (AfriSenti-SemEval). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics, 2023.

[60] Chayma Fourati, Hatem Haddad, Abir Messaoudi, Moez BenHajhmida, Aymen Ben Elhaj Mabrouk, and Malek Naski. Introducing a large Tunisian Arabizi dialectal dataset for sentiment analysis. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 226–230, Kyiv, Ukraine (Virtual), April 2021. Association for Computational Linguistics.

[61] Gati L. Martin, Medard E. Mswahili, and Young-Seob Jeong. Sentiment classification in swahili language using multilingual bert, 2021.

[62] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[63] Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[64] Ali Saleh Alammary. Bert models for arabic text classification: A systematic review. *Applied Sciences*, 12(11), 2022.

[65] Jacqueline Kazmaier and Jan H. van Vuuren. The power of ensemble learning in sentiment analysis. *Expert Systems with Applications*, 187:115819, 2022.

[66] Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236–246, 2017.

[67] Tim Buckwalter. Arabic transliteration. `http://www.qamus.org/transliteration.htm`, 2002.