Strathmore UNIVERSITY

SENTIMENT ANALYSIS ON SWAHILI-ENGLISH CODE SWITCHED TWEETS VIA TRANSFER LEARNING

Gachanja Jeremy Kibiru 101167

June, 2024

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Student's Name: Jeremy Kibiru Gachanja

Student's Signature: jkg

Supervisor's Name: Dr Betsy Muriithi

Supervisor's Signature: _____ Date: 4/9/2024

Contents

ABSTRACT

Sentiment analysis is a technique that is used to determine the sentiment, or emotional content, of a piece of text. When applied to code switched data, sentiment analysis can be used to determine the sentiment of text that contains words from multiple languages. This is a challenging task, as code switching can introduce complexity and ambiguity into the text. This study will present the use of transfer learning for sentiment analysis on Swahili-English code switched data using deep neural network models.

This study will focus on the use of transfer learning in conducting sentiment analysis on Swahili-English code switched data. The study will consider two pre-trained deep learning algorithms, that is mBERT and SwahBERT. This study will use these pre-trained deep learning models to conduct sentiment analysis on Swahili-English code switched tweets gathered between the period 29^{th} March 2022 to 15^{th} August 2022 and compare their performance using accuracy, specificity, precision, recall and f1 score metrics.

1 INTRODUCTION

Code switching, the practice of alternating between two or more languages or language varieties within a single conversation or discourse, is a common phenomenon in multilingual communities and can occur at various levels of language use, including phonetics, lexis, syntax, and discourse. It is a natural and dynamic process that reflects the complex linguistic and sociolinguistic realities of multilingual speakers.

Swahili, a Bantu language spoken in East Africa, is one of the most widely spoken languages in Africa and has a rich history of code switching with English and other languages. Code switching between Swahili and English, in particular, has been widely studied in the fields of linguistics, sociolinguistics, and education and has been found to serve a range of functions, such as emphasizing social identity, negotiating power, status, and conveying politeness.

However, code switching can also pose challenges for natural language processing tasks, such as sentiment analysis, which involves identifying and extracting subjective information from text, such as opinions, attitudes, and emotions. Sentiment analysis has important applications in fields such as marketing, customer service, and politics, where it can provide insights into the sentiments of consumers, customers, or voters.

One challenge of sentiment analysis on Swahili-English code switched data is the complexity and variability of code switching itself. Code switching between Swahili and English can involve switching between languages at different levels of granularity, from single words to entire sentences or discourse segments. It can also involve the use of hybrid forms, such as code-mixed words or phrases that contain elements of both languages. This can make it difficult to accurately identify and classify the language or languages being used in a given text and to apply language-specific sentiment cues and expressions.

Another challenge is the potential for language-specific sentiment cues and expressions. Both Swahili and English have distinct ways of expressing sentiments, such as through the use of specific words, phrases, or idioms. For example, Swahili has a range of positive and negative words, such as “furaha” (happiness) and “hasira” (sadness), that are not necessarily equivalent in English. This can make it difficult to accurately identify and classify the sentiment of

Swahili-English code switched texts that contain elements of both languages.

One approach to addressing these challenges is to use transfer learning, a machine learning technique that involves transferring knowledge from a pre-trained model to a new task or domain. Transfer learning has been widely used in natural language processing and has the potential to improve the performance of sentiment analysis on Swahili-English code switched data by leveraging the knowledge and insights gained from previous tasks and domains.

In this paper, we explore the use of transfer learning for sentiment analysis on Swahili-English code switched data. We first provide a review of the literature on code switching and sentiment analysis and describe the challenges and potential solutions for this task. We then present an empirical evaluation of the effectiveness of transfer learning in improving the performance of sentiment analysis on Swahili-English code switched data. Our study is based on a dataset of Swahili-English code switched texts and involves the use of state-of-the-art machine learning models and evaluation metrics.

1.1 Problem Statement

The problem of sentiment analysis on code-switched data, specifically Swahili-English code-switched data, is a challenging task due to the lack of adequate labeled data and the complexity of the language. The use of transfer learning techniques can potentially overcome these challenges by leveraging pre-trained models on monolingual data and adapting them to the code-switched data. The objective of this study is to investigate the effectiveness of transfer learning techniques in improving sentiment analysis performance on Swahili-English code-switched data.

1.2 Objectives

This study is aimed at:

- Identifying the best pre trained model that can be used for English-Swahili code switched data.
- Evaluate the performance of pretrained models on sentiment classification on English-

Swahili code switched data.

2 LITERATURE REVIEW

This chapter looks at past works done on sentiment analysis on monolingual and code switched data.

2.1 Past works on sentiment analysis on unilingual text data

Dubey (2020) conducted a study on Twitter sentiment analysis during the COVID-19 outbreak across 12 countries, including Australia, Belgium, China, France, Germany, India, Italy, Netherlands, Spain, Switzerland, UK, and USA. The study collected 50,000 COVID-19 related tweets from each country using the Twitter API and removed duplicates by eliminating retweets and replies. The tweets were then cleaned, scored using the NRC Emotional Lexicon, and analyzed on a sentiment and emotional scale. The sentiment scale revealed that Belgium had the highest proportion of positive tweets, followed by India and Australia. China had the highest proportion of negative tweets. On the emotional scale, the USA, France, and China had the highest tweets with anger, Switzerland had the highest number of tweets with fear and sadness, Belgium had the most tweets portraying trust and surprise, and Germany had the highest number of tweets displaying anticipation. Word clouds were also used to analyze frequently used words and their associated emotions in each country. The study found that words such as virus, political, pandemic, hospital, and death were frequently used across all countries. Additionally, the study identified Donald Trump as a frequently mentioned topic in tweets across the 12 nations. The major limitation of this study was that code switching was not considered since the study solely focused on English based tweets.

The research conducted by Nurulhuda Zainuddin (2014) aimed to improve the Support Vector Machine (SVM) algorithm as a sentiment classifier on benchmark datasets. The study used different n-grams and weighting schemes to extract relevant textual features, and trained the SVM algorithm on corpus datasets by Pang and Taboda. Data preprocessing involved tokenizing, removing stop words, converting tweets to lowercase, and stemming words. The data was pre-labeled, consisting of 2000 positive and negative movie reviews from IMDB and 400 positive and negative Simon Fraser University (SFU) student reviews. Unigrams, bigrams, and trigrams were used to compare how SVM performed in classifying tweets. Various

term-weighting schemes, such as Term Frequency Inverse Document Frequency (TDIF), Binary Occurrences (BO), and Term Occurrences (TO), were used for feature extraction to develop word vectors. Evaluation measures included precision, recall, accuracy, Area Under the Curve (AUC), and the F-measure. The SVM model was trained on both movie review data and student reviews data, with a training and test split of 70% and 30% respectively. Results showed that unigrams combined with TDIF yielded the best results for SFU student review texts, while binary occurrences were the best for feature extraction for movie reviews. Unigrams outperformed other n-gram models, and binary occurrences and TDIF weighting measures were the best for extracting features from movie review and student review texts respectively. However, just like Dubey (2020), this study solely focused on English based tweets.

In 2020, Murimo Bethel Mutanga (2020) conducted a topical modeling analysis on tweets from South Africans regarding the COVID-19 pandemic. The data was collected from Twitter using hashtags such as #COVID19SA and #CoronaVirusSA, as well as additional hashtags related to lockdown measures, stay-at-home orders, and COVID-19 statistics. The dataset consisted of 68,000 tweets, which were cleaned by removing symbols, URLs, and duplicate tweets, and then converting all tweets to lower case, removing stop words, and normalizing the text. The LDA algorithm was used to generate 10 topics from the dataset, including lockdown, 5G conspiracy theories, staying home, alcohol, police violence towards blacks, tracing of daily statistics, and Bill Gates conspiracy theories. The analysis found that conspiracy theories and mistrust towards the government were prevalent among South Africans, particularly in relation to 5G technology and vaccine testing. The study concluded that the government had work to do in gaining the trust of its citizens. The major limitation of this study was that it focused purely on English tweets, since African dialect text were removed during data cleaning.

The study by Laszlo Nemes (2020) used a recurrent neural network (RNN) to classify the emotions harbored by tweets over the COVID-19 pandaemic period. The model searched for connections between the words tweeted while attaching an emotion score on the tweet, classifying it as either positive or negative. However, the analysis further broke down positive

and negative emotions as either weakly positive or negative and strongly positive or negative.

The tweets analyzed were from the dates 13th May to 14th May 2020 and were in English. The aim of the study was to compare the performance of the RNN relative to other sentiment analyzers more specifically textblob. From this study it was found that even though there were negative emotions at the start of the pandemic, positivity has strengthened Laszlo Nemes (2020).

The RNN algorithm outperformed the textblob since it avoided classifying tweets as neutral. Neutral tweets do not tell one much when it comes to emotions and as such the RNN managed to pick out the smallest manifestation of emotion from every tweet. This thus showed that the RNN was a better model to conduct sentiment analysis on the tweets gathered (Laszlo Nemes (2020)).

Giridhar B. kamath (2020) analyzed the sentiments from India concerning the lockdown measures put in place by the Indian government. As China, Italy, Spain and Australia were being ravaged by the COVID-19 in the first week of March 2020, the Indian governement implemented a 21 day national lockdown from 25^{th} March to the 14^{th} April 2020.

This did not shock the Indian populous since they had already been exposed to a 14 hour lockdown on the 22^{nd} of March 2020. The tweets in this study were based on 2 hashtags namely #IndiaLockdown and IndiafightsCorona and the tweets sourced were from the 25^{th} of March to the 28^{th} of March 2020 (Giridhar B. kamath (2020)).

Negative emotions of fear, disgust and sadness about the lockdown were present, however the positive sentiments rose above the negative ones. This was primarily due to the fact that India‘s citizens were clear that they had to flatten the curve and were committed to this goal (Giridhar B. kamath (2020)).

Sentiments harboring emotions of trust towards the Indian government were also noted since they were probably sure their government would implement the lockdown successfully and that they would not lack basic essentials during the lockdown (Giridhar B. kamath (2020)).

Some of the tweets expressed sadnes since people were worried that their incomes were not sufficient to survive the lockdown. For matters concerning alcohol consumption there were

tweets that expressed concerns for alcohol addicts suffering from withdrawals due to the unavailability of alcohol (Giridhar B. kamath (2020)).

The fight against COVID-19 in India was taken up positively by its citizens and most of them were in agreement with the government's initiative of the nationwide lockdown. It was also noted that some of the citizens were angry the lockdown was not implemented sooner (Giridhar B. kamath (2020)).

Matters around quarantine and self isolation were also evident in the tweets. People were advocating that those flying into the country should be quarantined before reuniting with their families. From this there was a positive response to lockdown and quarantine and this indicated that India was successful in controlling the spread of COVID-19 (Giridhar B. kamath (2020)).

During the COVID-19 pandemic, a lot of conspiracy theories arose as to the cause of the virus especially due to misinformation about the disease. Usman Naseem (2021) conducted a study aimed at informing policy that could have been applied during the pandemic to curb misinformation of the general public.

This study used 90,000 COVID-19 related tweets collected from February to March 2020. They classified the collected tweets into 3 categories namely: positive, negative and neutral. The tweets analyzed were limited to the English language (Usman Naseem (2021)).

From the gathered tweets, 12 topics were identified. Some of them were quarantine, lockdown and stay home. TextBlob was used to clasify the tweets as positive or negative (Usman Naseem (2021)).

Usman Naseem (2021) found that the public supported the lockdown in February, however, their support for the lockdown shifted by mid-March. The study could not link the shift in support to misinformation and conspiracy theories circulated on twitter, but it had not been ruled out as a cause of this. The best way to combat the misinformation about the disease was to set up an active public health presence to inform the public on matters relating to COVID-19 thus nipping misinformation and all its resulting effects at the bud.

2.2 Past works on sentiment analysis on code switched text data

This section is a review of the existing research on the use of sentiment analysis techniques to analyze data that contains code-switching. The use of sentiment analysis on code-switched data presents several challenges.

One challenge is the lack of labeled data for training sentiment analysis models on code-switched data. Most existing sentiment analysis datasets are in a single language, and manually labeling code-switched data for sentiment analysis is time-consuming and labor-intensive. To address this challenge, several researchers have proposed methods for automatically generating labeled code-switched data for sentiment analysis. For example, in their paper "Generating Code-Mixed Sentiment Corpus for Indian Languages," Chatterjee et al. (2016) proposed a method for automatically generating code-mixed sentiment corpora for Indian languages by combining sentiment-annotated monolingual corpora and code-switched data.

Another challenge is the complexity of code-switching itself. Code-switching can occur at various levels, such as the lexical, syntactic, and discourse levels, and it can have different motivations, such as style shifting, emphasis, or the expression of identity. This complexity makes it difficult to accurately identify and classify the sentiment of code-switched data. To address this challenge, several researchers have proposed methods for modeling the sentiment of code-switched data at different levels of code-switching. For example, in their paper "Sentiment Analysis of Code-Switched Data: A Survey," Zhang et al. (2018) surveyed the existing approaches to sentiment analysis of code-switched data and categorized them according to the level of code-switching they model.

Jerbi, Achour, and Souissi (2019) explored the use of Recurrent Neural Networks models (RNN) in conducting sentiment analysis on Tunisian code switched data. RNN was chosen since it considers the order in which words appear and can work with inputs of varying lengths. The type of RNN used was the LSTM (Long short term memory) model. The data was a Tunisian dialect corpus of 17,000 comments. Each of the comments were manually labeled as either positive or negative. The comments were a mix of Arabic and Latin languages. After the comments were annotated, word embedding was conducted ,which entails encoding words

into numbered vectors.

The analysis considered 4 LSTM variants, which were LSTM, deep LSTM, bi (bidirectional)-LSTM, and deep (stacked) bi-LSTM. For LSTM it solves for explosion and disappearing gradients which is common during the training of traditional RNNs. It is composed of a cell, which stores data for some time, and three gates which are the input, output and forgetting gate. The three gates control the flow of information to and from the cell thus determining the information to forget or pass on to the next time step. For deep (stacked) LSTM , it is composed of several hidden LSTM layers. The LSTM layer provide a sequence of output to the LSTM layer below. The multiple LSTM layers make the model deep.

The bi-LSTM is composed of two LSTM layers, which are the forward LSTM and backward LSTM. The forward LSTM looks at the input from beginning to end, while the backward LSTM lo oks at the input from end to beginning. The two LSTM layers are interconnected and collaborate to give one output. The interaction between the two layers allows bi-LSTM to learn from both the past and the future. The deep bi-LSTM, is a combination of both deep LSTM and bi-LSTM thus allowing it to identify deeper relationships between the words like the deep LSTM while also learning from the past and the future like a bi-LSTM.

The analysis found that deep LSTM had the highest accuracy of 90% when it came to classifying the code switched comments. LSTM had the lowest accuracy of 67%. Bi-LSTM and deep bi-LSTM had an accuracy of 70% and 88% respectively. The challenge faced in this study was there are few annotated code switching data available.

For Shakeel and Karim (2020) ,used deep learning to conduct sentiment analysis on the labelled multisenti dataset. The study was done by avoiding lexical normalization, language translation and code switching indication. Lexical normalization entails transforming non standard text into a standard register, for example transforming " I'll send pics tomoroe" to "I'll send pictures tomorrow." By avoiding lexical normalization, the algorithm built would have to understand the context of every word in a text and account for the shortcuts people take when passing messages which result in breaking the grammar laws of English. Language translation was avoided since the aim was to build an algorithm that could use context in understanding a new language. Since the design of the analysis was to create an algorithm

that could understand lingual heuristics and code switching.

The model built incorporated character embedding to increase the model's efficiency. Character embedding uses one-dimensional convolution neural network (1D-CNN) to find numeric representation of words by looking at their character composition. The process converts words to numbers but from the word's characters point of view. The data used for the analysis was Twitter data gathered during the Pakistan general elections. The aim of the study was to identify the overall emotion and sentiment towards the election. The tweets were not labelled, therefore a gold standard dataset was created by annotating 20,735 tweets which was done by domain experts.

Text characteristics were also added as labels to the tweets based on their multilingual or monolingual nature. Monolingual was tweets containing only Roman, Urdu or English languages while multilingual was for tweets which had a mixture of any of the languages. For data preprocessing , the tweets were converted to lower case and single worded tweets were dropped. For model building an 80%-20% train-test on the split was applied on the gold standard data via stratified sampling.

The deep learning models chosen were convolution neural networks from Medrouk and Pappa (2017), attention-LSTM by Zhou, Wan, and Xiao (2016), and simple convolution neural networks from Attia et al. (2018). Word embedding was done using the ELMo (Embeddings from Language Model), which represents the embeddings of a word using the whole sentence that contains the word. It was found that convolution neural networks and attention LSTM performed best using ELMo embeddings but with parameter fine tuning. It was also seen that these models underpermed when applied on informal languages as compared to formal languages.

Due to this Zhou, Wan, and Xiao (2016) proposed an (Multi-cascaded Model)McM model which used 4 different neural networks as its layers. The McM model had a stacked-CNN which learns the n-gram features for identifying the relationship between words. This layer is composed of 2 stacked CNN layers.

The stacked LSTM learner layer captures the information from the order in which words

appear in a tweet. Here each word is treated as a single time step and fed into the LSTM layer sequentially. The layer is composed of 2 stacked LSTMs. The LSTM leaner layer captures the long term relationships of the text. The discriminator network layer aggregates the information learnt by the 3 prior layer models and merges them to give a final prediction. This layer in itself is 2 layered.

Hyper-parameter optimization was done through grid search. The metrics used to rate the models performance were accuracy, precision, recall and f1 score. When classifying multilingual tweets McM got the highest f1 score of 0.65, while simple convolution neural network got the lowest f1 score of 0.17, without model fine tuning. After model fine tuning McM's f1 score dropped 0.64 but still had the highest f1 score while simple convolution neural networks had an f1 score of 0.17. For accuracy, McM had the highest accuracy of 0.68 and 0.69 with and without fine tuning, while simple convolution neural network had an accuracy of 0.35 with and without fine tuning. This shows that for McM even though overall accuracy improves during fine tuning, the f1 score suffers.

Chundi, B., and Hulipalled (2020) proposed the SAEKCS (Sentiment Analysis for English – Kannada Code Switch) deep learning model for sentiment analysis on Kannada-English code switched data. The SAEKCS model is a combination of the bi-LSTM and CNN (Convolution Neural Network). Kannada is dialect spoken in the southern parts of India. The data was obtained from you tube comments. The data contains, pure English text, pure Kannada text and a combination of Kannada and English text. The analysis was focused more on conducting sentiment analysis on the code switched text and such, the Kannada-English text were extracted from the main dataset. The resulting code switched dataset had 10,401 comments. The comments were then manually annotated as being positive, negative and neutral.

For preprocessing symbols, characters and digits were removed and the comments converted to lower case. On tokenizing the comments, 126,947 tokens and 34,191 unique words were obtained. During further preprocessing, one character words were removed. After preprocessing two types of mapping were done, that is character to number and number to character mapping. The characters were then replaced with their respective unique

mapped values. Post padding was then applied on the data with a maximum sentence length of 200 words. Padding is done to make the comments of equal word size, since deep learning algorithms only take inputs of equal size. The next step was applying sub word level embedding on the comments. Ordinary word embedding was not used since it assumes that a language has a fixed set of vocabulary and it does not use morphological information. Morphological information is how words are put together. Sub word level embedding was thus used since it can incorporate morphological information from words and allows one to derive meaning from out of vocabulary words present in code switched data Chundi, B., and Hulipalled (2020).

The SAEKCS model is a hybrid of CNN and bi-LSTM. The CNN layer together with maxpooling extract the dependencies between the different sentences, and transfers this output to the next layer as an input. For bi-LSTM it extracts the long term dependency between the sentences and gives equal importance to all the inputs received from the CNN layer. A softmax activation function is then used to conduct the sentiment classification. The train test split was 80%-20% with the training sample broken down into 64% train and 16% validation data sample Chundi, B., and Hulipalled (2020).

The sentiment analysis was done in 3 stages, which were positive and non-positive, negative and non-negative and the third phase was combining the first two to generate the classification matrix for positive, negative and neutral comments. In the first stage, the data contained 3342 positive comments and 7059 non-positive comments. Here SAEKCS got an accuracy of 83.2%. The precision, recall and f1 scores for positive classification were 79%, 66%, 72% respectively, while for non-positive classification the precision, recall and f1 score were 85% ,91%, and 88% respectively Chundi, B., and Hulipalled (2020).

In the second stage, the data contained 3434 negative comments and 6967 non-negative comments. Here SAEKCS got an accuracy of 73.7%. The precision, recall and f1 scores for negative classification were 67%, 59%, and 63% respectively, while for non-negative classification the precision, recall and f1 score were 80%, 85%, and 83% respectively Chundi, B., and Hulipalled (2020).

In the third stage, on combining the models got in stages 1 and 2, the SAEKCS model got

an accuracy of 77.6%. Here, the comments taken as positive in stage 1 are positive and those taken as negative in stage 2 are considered negative, the rest were thus categorized as neutral. The precision, recall and f1 scores for negative classification were 67%, 59%, and 63% respectively. The precision, recall and f1 scores for neutral classification were 82%, 90%, and 86% respectively. The precision, recall and f1 scores for positive classification were 83%, 60%, and 70% respectively Chundi, B., and Hulipalled (2020).

On comparing SAEKCS performance to traditional algorithms such as Naive Bayes,bi-LSTM, and subword-LSTM, these algorithms had accuracy of 39.6%, 55.9% and 64.8% respectively. These algorithms perform poorly on code switched data since they use either word or character embedding as compared to SAEKCS which uses sub-word level embedding Chundi, B., and Hulipalled (2020).

Gupta et al. (2021) presents an Unsupervised Self-Learning framework for conducting sentiment analysis on code switched data through the use of pre-trained BERT models for initialization and fine tuned them in an unsupervised manner using pseudo labels got by zero shot transfer. The analysis was aimed at determining if the resulting model understood the code switched languages or if it was just learning its representations.

The first step of the analysis was generating the psuedo labels. This was done by using the pretrained model to come up with sentiment labels for the data. The top N most confident predictions are selected and their labels used to fine tune the model. The confidence in prediction is measured using the softmax score. The generation of pseudo labels made up the initialization block of the model. The psuedo labels are derved from a zero shot model. The zero shot model should be a pre-trained model that can be used for the classification task. For example, if the data is Hindi-English, the zero shot model should be one which has been pre-trained on such data. The four datasets have the English language in common, as such the RoBERT model which has been trained on 60 million English tweets was used as the initialization model. To generate the zero shot labels (pseudo labels), the RoBERT model was implemented using the Hugging Face implementation. This model has the benefit of pre-processing data by removing URL links. RoBERT was then compared to other supervised models trained on the four datasets. The supervised models were trained by finetuing the

RoBERT model on each of the four datasets. The analysis found that RoBERT outperformed mBERT and XLM-RoBERT on code switched data Gupta et al. (2021).

The fine tuned model was then used to conduct sentiment classification on the rest of the dataset, the top N most confident predictions are selected, then the labels generated are used to fine tune the model. This process was iterated utill the model went through all the data. The framework was applied on 4 different languages, that is Hinglish, Spanglish, Tanglish and Mayalam-English.presents a method for improving the performance of sentiment analysis on code-switched data, which refers to texts that contain elements of multiple languages or language varieties. Code-switched data can be challenging to analyze due to the complexity and variability of code-switching itself, as well as the potential for language-specific sentiment cues and expressions. To address these challenges, the authors propose an unsupervised self-training approach, which involves using a pre-trained model to generate pseudo-labels for a code-switched dataset and then using these labels to fine-tune the model on the code-switched data. When it came to the two class sentiment classification, the RoBERT model had a highest accuracy in generating zero shot labels for Hinglish and the lowest in Mayalam-English Gupta et al. (2021).

The framework had 2 ways of assessing model performance. The first was on how good was the model when trained in an unsupervised way, primarily assessing the strength of the model. This was done by comparing the performance of the unsupervised model with that of the supervised model. The second way was from an algorithmic perspective. This entailed checking the model's accuracy in the classification task.on comparing the unsupervised model to its supervised counterpart, it performs quite well as its supervised counterpart for both Spanglish and Hinglish datasets. This was attributed to both the datasets being in the RoBERT model domain. However, RoBERT performs poorly in the Tamil and Mayalam datasets Gupta et al. (2021).

It was observed that the RoBERT model had higher zero-shot f1 scores in the Hinglish and Spanglish datasets with scores of 0.32 and 0.31 as compared to the Dravidian languages where the model got low f1 scores of 0.15 and 0.17. This shows that the RoBERT model was not ideal for the Tanglish and Malayalam languages. On comparing RoBERT;s perfomance

to supervised machine learning models, for Hinglish and Spanglish datasets, the unsupervised model f1 and accuracy scores were almost as high as those of the supervised model. RoBERT (unsupervised model) got accuracy and f1 scores of 0.84 for Hinglish while for Spanglish the model got an accuracy and f1 score of 0.76 and 0.77 respectively. These results were almost a match for those got by the supervised model on the two datasets. The supervised model got an accuracy and f1 score of 0.91 in the Hinglish data while for Spanglish, the model got an accuracy and f1 score of 0.79 and 0.78 respectively. However, on comparing model performance for Tamil and Mayalam, the f1 and accuracy scores for the unsupervised model are much lower compared to the supervised model, that is accuracy and f1 scores of 0.9 for supervised and 0.71 and 0.73 for unsupervised for the Malayalam language. For the Tamil language, the supervised model got accuracy and f1 scores of 0.85 and 0.83 respectively while for the unsupervised model the accuracy and f1 scores were 0.71 and 0.73 respectively. Therefore, unsupervised learning methods performed competitively when compared with supervised models Gupta et al. (2021).

Martin et al. (2022) pretrained the monolingual BERT model for Swahili SwahBERT and compared it to the multilingual BERT (mBERT) on four downstream tasks. The tasks were emotion classification, news classification, sentiment classification and NER (Named Entity Recognition).

To pretrain the model, data was scraped from sources such as: news websites, forums and Wikipedia. The data gathered here covered aspects such as politics, education and lifestyle. The data was cleaned by removing URLs usernames, non-textual content, and filtered out non-Swahili characters. The resulting dataset had 16,000,000 words Martin et al. (2022).

For the emotion classification dataset, the analysis focused on 6 emotions, that is: joy, anger, surpirse, disgust, joy, fear and sadness. This data was scraped from social media platforms like Twitter, Youtube, and JamiiForum. Profanity towards groups or specific people was removed. Also an additional 3 English datasets with relevant topic coverage were sourced and converted to Swahili. The data gathered from the Swahili forums were then manually annotated by native Swahili speakers Martin et al. (2022).

The SwahBERT model used monolingual tokenizers with different vocabulary sizes for Swahili.

This was to account for the linguistic complexity of the language. The model is composed of 12 encoder blocks and 768 hidden units. The model was pretrained using two methods which were Masked Language Modeling (MLM) and Next Science Prediction (NSP). The model parameters were optimized using the Adam optimizer. The model yielded the best results when trained with a vocabulary size of 50,000 words Martin et al. (2022).

For the emotion classification task, the data was split into 75% training, 10% for development and 15% for testing. On comparing the f1 scores between the models, SwahBERT had the highest f1 score of 64.46 and mBERT had a score of 60.52. The model had the best performance in classifying emotions of joy, sadness and surprise Martin et al. (2022).

For news classification the data was split into 3 sets of 80% training, 10% development and 10% for testing. The data had 6 news categories which were: national, international, finance, sports, health and entertainment. The f1 score for the SwahBERT was higher than that of mBERT in all of the news classes except health. For the health class, mBERT and SwahBERT got the same f1 score of 0.45. This wasdue to the class imbalance in the news data. The health class had the fewest observations Martin et al. (2022).

For sentiment classification, the data used was derived by associating emotions from the emotion dataset with positive, negative, and neutral sentiments. Joy was equated to positive sentiment while disgust to negative. The remaining emotions were classified as neutral, however, surprise was removed from this category since it could be mapped to either negative or positive. The data was split similar to the emotion classification task. For this task, SwahBERT outperformed mBERT with a gap of 3-6% in f1 scores .When it came to the named entity recognition task, the SwahBERT and mBERT were evenly matched. The main reason for this was the small nature of the NER dataset. However, the analysis believed that with more data, a distinction between the models would be present Martin et al. (2022).

The paper by Barman, Das, Wagner, and Foster discusses the challenges of language identification in code mixed text, which is text that contains words from multiple languages. The authors point out that code mixing is a common phenomenon in the language of social media, and existing language identification systems are not always able to accurately identify the languages used in code mixed text. The authors propose a new approach for language

identification in code mixed text, which uses a combination of word-level and character-level information to identify the languages used in the text. The authors also present experimental results that show the effectiveness of their approach on several benchmark datasets. Overall, the paper provides a valuable contribution to the field of language identification in code mixed text.

The paper by Lother, Mswahili, and Jeong (2021) used the multilingual version of the BERT (Bidirectional Encoder Representations from Transformers) model to perform sentiment classification on Swahili text data. The data used was created by extracting and annotating 8,200 reviews and comments on various social media platforms and the ISEAR emotion dataset. The data was labelled as either positive or negative.

BERT is a bidirectional unsupervised deep learning model which has been trained using BooksCorpus composed of 800 million words and English Wikipedia composed of 2.5 billion words. This allows BERT to learn he context of a word based on all surrounding text. There are two variations of BERT, that is BERT base and BERT large with 12 and 24 encoder layers respectively. The model is built in 2 phases that is pre-training and fine-tuning. During training the model is trained on unsupervised data over different pre-training tasks. During fine-tuning the model is first initialized with the pretrained parameters and then fine-tuned using labeled data. The model's input is limited to 512 tokens Lother, Mswahili, and Jeong (2021).

In order to feed in sentences into the model, they need to be split into tokens using the WordPiece tokenizer, and then these tokens are mapped to their index in the tokenizer vocabulary. The benefit of using the WordPiece tokenizer is that it can be used to map out of vocabulary words. It does this by splitting the out of vocabulary word into character tokens that can be mapped to the tokenizer vocabulary. For the analysis @Lother, Mswahili, and Jeong (2021) used the multilingual version of BERT which has been trained on the Wikipedia pages of 104 languages, Swahili being amongst them. Given the different Swahili speaking tribes and the various English loan words that have been modified for use in Swahili, mBERT was considered for this task given its incorporation of out of vocabulary words. An additional dense layer was added on the pretrained model while keeping the other hyperparameters

constant.

The ISEAR dataset which contains 7 emotions namely: joy, anger, sadness, disgust, shame and guilt were converted sentiments by setting joy as positive and all the others as negative. For preprocessing, symbols, usernames, links and punctuations were removed. The data was then split using a 90/10% train test split. The hyper-parameters were finetuned using the HuggingFace implementation in python.The mBERT model achieved an accuracy of 87.59%. However the precision, recall and f1 measures for the negative sentiment class were higher than those of the positive class. This was primarily due to the unbalanced nature of the data with 535 and 287 negative and positive observations respectively Lother, Mswahili, and Jeong (2021).

Devlin et al. (2019) presents the BERT (Bidirectional Encoder Representations from Transformers) model. BERT pre trains deep bidirectional representations from unlabeled text from both right to left and left to right to capture context. The model was tested on 11 natural language processing tasks. The model was built through two main steps, which were pre-training and fine tuning. For pre-training the model was trained on unlabeled data over different pre training NLP tasks, then the model parameters are fine tuned using labeled data. For this analysis two variations of BERT were considered, that is BERT base and BERT large which have 12 and 24 hidden layers respectively.

WordPiece embeddings which has a token vocabulary of 30,000 words was used to built BERT such that the representation of the input fed into the model can unambiguously represent both a single sentence or a pair of sentences in one token sequence. For pre-training two unsupervised methods were used which were: Masked LM and Next Sentence Prediction (NSP). Masked LM was done by hiding a proportion of the input tokens at random and then predict those masked tokens. For Next Sentence Prediction (NSP) the model was trained to understand sentence relationships. The model was pre-trained on a binarized next sentence prediction dataset with labels next and not next. The sampling was done such that, if one has 2 sentences, A and B, for the training sample, 50% of the time B was the sentence that follows A and 50% of the time it was a random sentence from the corpus. The data used for the pre-training tasks were: BooksCorpus composed of 800 million words, and English

Wikipedia, that contains 2,500 million words.

The pre-training tasks were summarized into 4 main groups, they were GLUE, SQuAD v1,1, SQuAD v2.0, and SWAG. The General Language Understanding Evaluation (GLUE) is a collection of various diverse NLP and understanding tasks. The NLP and understanding tasks found under GLUE are: Multi General Natural Language Inference (MNLI), Quoa Question Paris (QQP), Question Natural Language Inference (QNLI), Stanford Sentiment Tree bank (SST-2), Corpus of Linguistic Acceptability (CoLA), Sentiment Textual Similarity (STS-B), Microsoft Research Paraphrase Corpus (MRPC), Recognizing Textual Entailment (RTE) and Winograd Natural Language Inference (WINLI). MNLI is a classification task where given a prior statement, a model is to classify whether the next statement is an entailment, contradiction or neutral.

For QQP, it is a classification task where a model is to determine if two questions asked on Qoua are semantically the same. For QNLI the model is trained to determine which sentences have their correct answers and which ones do not. SST-2 is a sentiment classification task from annotated movie reviews, CoLA is aimed at training model on determining if a sentence is linguistically acceptable or not for the English language. STS-B is an NLP task which is meant to train a model to identify how similar a pair of sentences are in terms of semantic meaning.

For the GlUE tests, it was observed that for $BERT_{LARGE}$, fine tuning was unstable on small data sets. $BERT_{LARGE}$ and $BERT_{BASE}$ performed exceptionally well on all GLUE tests with margins of over 4.5% and 7% respectively for average accuracy improvement. This comparison was done on 3 major NLP deep learning models. These models were Pre-OpenAI SOTA, BILSTM-ELMo-Attn and OPenAI GPT. $BERT_{LARGE}$ achieved an average score of 80.5 while OpenAI GPT got an average score of 72.8. Overally $BERT_{LAGRE}$ outperformed $BERT_{BASE}$ on all the GLUE classification tasks. BERT was then trained on the Stanford Question Answering Dataset (SQuAD v1.1). This datasets is composed of 100 thousand question and answer pairs. The aim of this task was to train the model such that given a question, and a passage Wikipedia containing the answer, the task is to predict the answer text from that passage. For this task BERT and its different variations achieved an average

f1 score of 0.92. The BERT models outperformed the lop leaderboard systems of 2018, such as Ensemble-nlnet and Ensemble-QANet, which achieved f1 scores of 0.917 and 0.905 respectively.

The Stanford Question Answering Dataset (SQuAD v1.1) is composed of 100,000 crowd sourced question and answers. This task is where given a question and a passage from Wikipedia containing that answer, the task is to predict the answer text span from that passage. $BERT_{BASE}$ and $BERT_{LARGE}$ were compared to ensemble-nlnet, ensemble-QANet, BiDAF-ElMo, and R.M. Reader. The two versions of BERT outperfomed the other 4 models by attaining the highest f1 scores of 91.8 and 93.2 respectively. SQuAD v2.0 is an extension of the SQuAD 1.1 by allowing the algorithm to find the shortest possible answer. $BERT_{LARGE}$ outperformed single-MIR-MRC, single-nlnet, unet, and SLQA by attaining the highest f1 score of 83.1 for this task. For the Situations With Adversarial Generations (SWAG), this task contains 113,000 sentence-pair completion examples where the model predicts the most likely ending of a given sentence based on a given context. The context consists of a few sentences that describe a situation, while the target sentence represents a plausible continuation of the situation. On comparing the BERT models with ESIM-GloVe, ESIM-ELMo and OpenAIGPT, BERT outperfomed them having an f1 score of 88.0. The study therefore concluded that deep unidirectional architectures can be applied on low-resource tasks and that deep bidirectional architectures allow the same pre-trained model to successfully handle a broad range of NLP tasks.

Given that mBERT and SwahBERT originate from this BERT structure, one can confidently conclude that these models can understand English and have the capacity to be trained on new languages such as Swahili.

Conclusion

From the above literature it is evident that deep learning pre-trained models are the best when it comes to conducting sentiment analysis on code switched data. Given their bi-directional structure, they can easily derive context and due to their pre-trained nature, they can handle multiple Natural Language Processing tasks. However, when selecting pre-trained models,

one must pick a model which has been used for an almost similar NLP task to theirs. Given that the languages of interest are Swahili and English, the most appropriate models for sentiment analysis on such code-switched data would be mBERT and SwahBERT since they have both been trained on both languages.

3 METHODLOGY

3.1 Introduction

This study is aimed at determining which pre-trained deep learning algorithm is best for sentiment analysis on Swahili-English code switched data. This will be done by comparing two models that is SwahBERT and mBERT using various classification metrics.

3.2 Research design

The main aim of this study is to determine which pre-trained deep learning algorithm would be the best for conducting sentiment analysis on Swahili-English code switched data. The pre-trained models that have been considered in this study are SwahBERT and mBERT. The models will then be compared using metrics such as f1 score, recall, precision, specificity and Area Under the Curve (AUC).

3.3 Population and sampling techniques

The analysis will be done using election tweets data which were gathered during the 2022 Kenya general elections campaign period. The tweets were gathered during the period between March 29^{th} 2022 to August 15^{th} 2022. The tweets gathered were 80,900 tweets in total.

3.4 Data collection

The data was mined from the social media platform Twitter using the TwitterAPI. The tweets were mined by use of hashtags that were trending during the Kenya 2020 general elections. Some of these hashtags were: “#kenyakwanza”,“#WilliamRuto”,“#Raila”,“#kivumbi2022”,“#kenyanspoll”, and “#Rutothe5th”, just to mention afew.

3.5 Data analysis

The data preprocessing steps will entail removal of symbols, URLs, emotions, stop words, and numbers. This study will focus on code switched text, therefore the code switched texts will be filtered out. The extracted code switched texts will then be manually annotated with the labels positive, negative and neutral. Sub word embedding will then be done in preparation for being passed into the deep learning algorithms.

3.6 Model building

For effective transfer learning, one is advised to use pre-trained algorithms that have been pre-trained on the problem of interest as advised by Gupta et al. (2021). Therefore the models considered here will be the SwahBERT model and multilingual-BERT (mBERT) model. These two models have the same architecture as the BERT model described in Devlin et al. (2019). The model architecture can be summarized in five major steps.

Input Encoding: BERT takes variable-length text input and converts it into a sequence of word embeddings using a combination of token, segment, and position embeddings. Token embeddings represent the meaning of each word in the input, segment embeddings differentiate between multiple sentences in the input, and position embeddings encode the position of each word in the input sequence.

Transformer Encoder: BERT uses a deep bidirectional transformer encoder to process the input sequence of embeddings. The transformer encoder consists of a stack of multiple identical layers, each containing multi-head self-attention and feedforward neural networks. The self-attention mechanism allows the model to attend to different parts of the input sequence and learn contextual representations for each word based on its surrounding words.

Pre-training: BERT is pre-trained on large amounts of unlabeled text data using two unsupervised learning tasks: masked language modeling (MLM) and next sentence prediction (NSP). MLM involves randomly masking some words in the input sequence and training the model to predict the masked words based on their context. NSP involves predicting whether two input sentences are consecutive in the original text or not.

Fine-tuning: After pre-training, BERT can be fine-tuned on various downstream natural language processing (NLP) tasks such as text classification, question answering, and named entity recognition. The fine-tuning involves adding a task-specific output layer on top of the pre-trained BERT model and training the entire network on the downstream task.

Output: The final output of the BERT model is a sequence of hidden vectors for each input token, which can be used for various downstream NLP tasks. The model can be further fine-tuned or modified for specific tasks by adding additional layers or changing the output layer.

4) RESULTS

4.1) Data Validation

The comprehensive exploration of the dataset, characterized by its rich amalgamation of Swahili and English text (indicative of code-switching), was conducted through a series of analytical phases. These included standardized text preprocessing, exploratory visualization via word clouds, and a detailed examination of linguistic patterns through the lens of bigram and trigram frequencies. The dataset's inherent linguistic diversity and the presence of code-switching offered a nuanced perspective for bilingual text analysis, particularly within the scope of sentiment analysis with an emphasis on Swahili-English code switching.

4.1.1 Textual Data Standardization

The initial phase of our analysis involved a rigorous text cleaning process. This standardization was crucial for the reduction of analytical noise and the enhancement of the accuracy of subsequent linguistic evaluations. The data was standardized by removing all symbols, stop words, url links and hastags.

4.1.2 Linguistic Visualization through Word Clouds

The creation of a word cloud from the sanitized dataset unveiled a visual representation of the prevailing linguistic elements, illustrating a balanced presence of English and Swahili terms. Notably, this visual exploration highlighted terms predominantly associated with political discourse, suggesting a thematic concentration on political dialogues and the interaction between the two languages.

4.1.3 Linguistic Pattern Analysis via Bigrams and Trigrams

Deeper insights were gleaned from examining the dataset's linguistic structure through bigrams and trigrams:

Analysis of bigrams as shown on figure 2 pointed to a significant emphasis on political figures and ideologies, reflecting a dataset deeply embedded in political discourse. The top bigrams

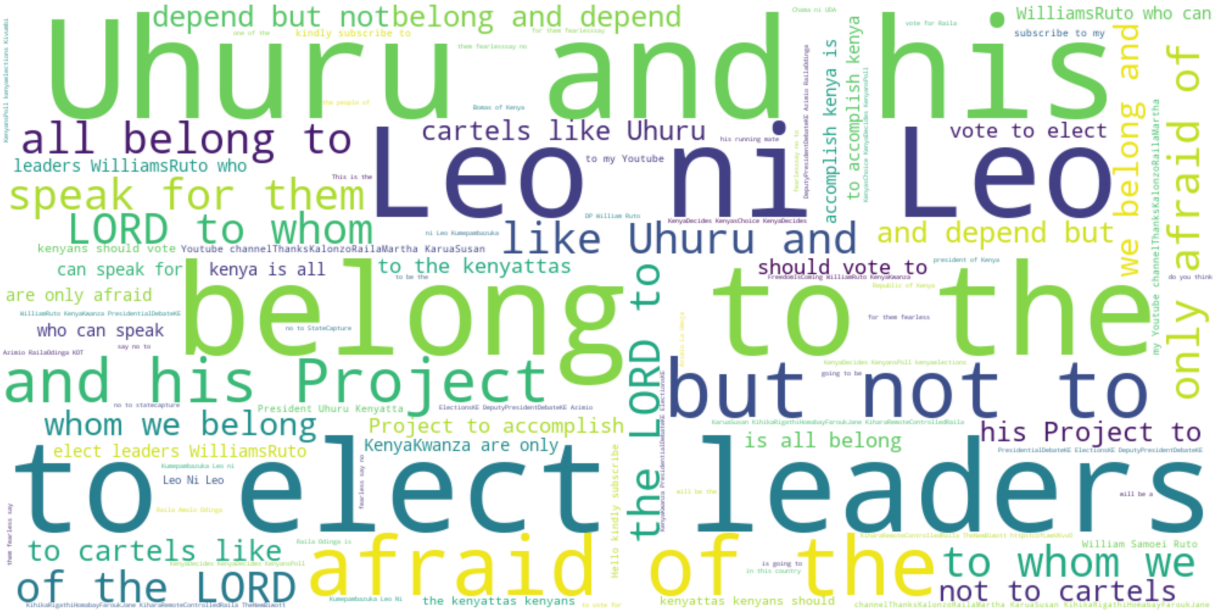


Figure 1: Word Cloud for the text data

Phrase	Frequency
raila odinga	2088
kenya kwanza	1933
william ruto	1546
martha karua	1127
kenyadecides kenyanspoll	1069
uhuru kenyatta	1014
ni leo	963
leo ni	963
running mate	838
deputy president	821
kenyadecides kenyadecides	772
mt kenya	707
ruto raila	699
rigathi gachagua	678
dp ruto	664
raila ruto	617
president uhuru	595
kenyadecides kenyaschoice	572
th president	550
kenyakwanza azimiolaumoja	549

Figure 2: bigram analysis

emphasizes the dataset's focus on key political figures with "raila odinga" and "kenya kwanza" being the most frequent. This indicates not only the dataset's rich political discourse but also the significant public interest in these entities or individuals. The prominence of political names and slogans suggests the data could be instrumental in understanding public sentiment towards political leaders and movements.

The trigram analysis, shown on figure 3, highlights specific phrases that further point to the dataset's political and anticipatory nature. Phrases such as "leo ni leo" (today is the day) suggest moments of expectation or significant events, likely in a political context. The repetition of specific names in conjunction with actions or descriptors like, "uhuru and his" underlines the focused discussion on political narratives and figures.

Phrase	Frequency
Uhuru and his	470
to elect leaders	456
belong to the	454
Leo ni Leo	452
and his Project	449
afraid of the	449
but not to	449
all belong to	447
speak for them	447
of the LORD	447
like Uhuru and	447
whom we belong	446
to whom we	446
cartels like Uhuru	446
to cartels like	446
not to cartels	446
depend but not	446
Project to accomplish	446
and depend but	446
can speak for	446

Figure 3: Trigram analysis

4.1.4 Assessment of Code-Switched Content

From figure 4 Our analyses estimated that a substantial 85.8% of the dataset's texts exhibit characteristics of code-switching, underscoring the linguistic diversity and the prevalent

blending of English and Swahili. This observation confirms the dataset's aptitude for reflecting real-world bilingual communication practices, particularly those embodying code-switching phenomena.

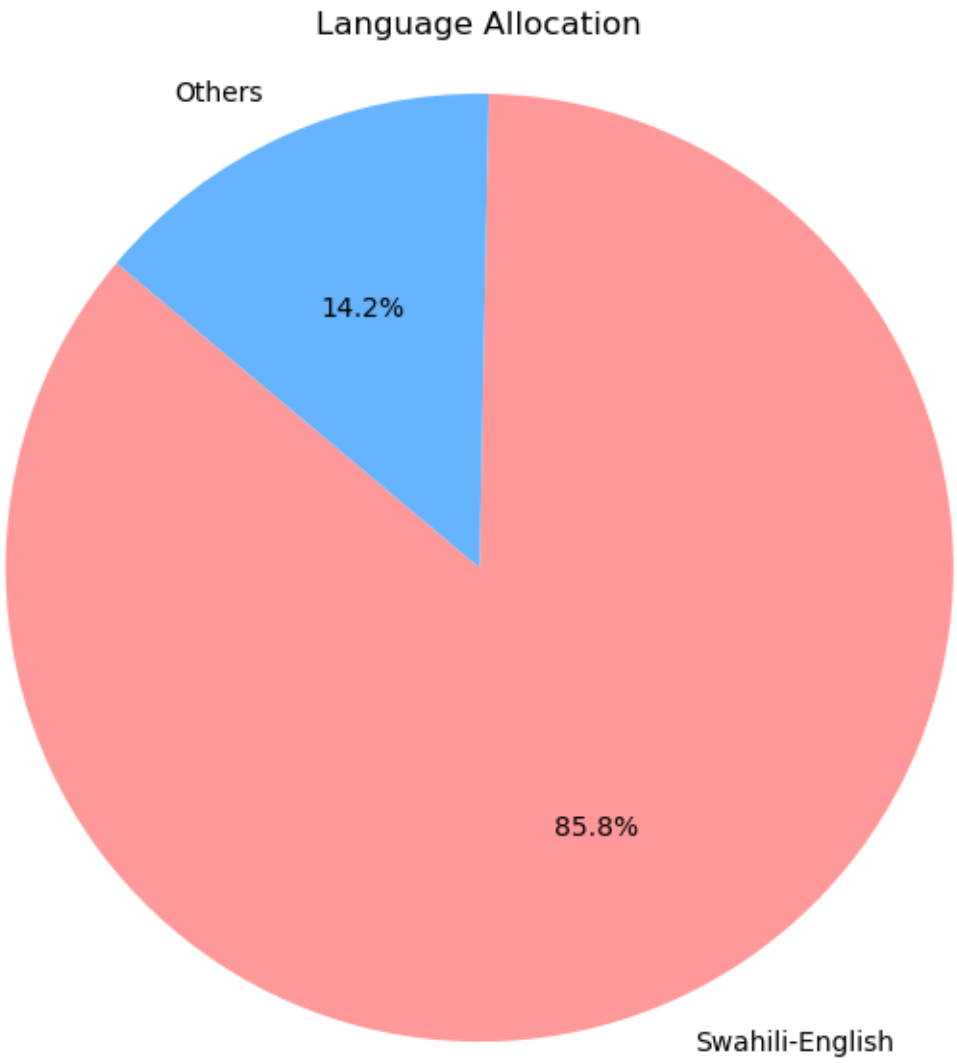


Figure 4: Lamguage distribution

The layered analysis presented herein, bolstered by both visual and analytical insights, unequivocally demonstrates the dataset's applicability in navigating the complexities of bilingual sentiment analysis in contexts marked by code-switching. The pronounced presence of code-switched phrases, coupled with the identified political and cultural lexicon, accentuates the dataset's potential utility in the development of sophisticated sentiment analysis frameworks.

4.2 Model Performance

mBERT, built upon the BERT architecture, undergoes pre-training on a diverse multilingual corpus using tasks such as Masked Language Model (MLM) and Next Sentence Prediction (NSP). This pre-training process equips mBERT with a broad understanding of linguistic patterns across multiple languages, allowing it to capture cross-lingual relationships and general linguistic structures. Consequently, mBERT demonstrates versatility in handling tasks across various languages, leveraging its proficiency in capturing bidirectional context and relationships between sentences Devlin et al. (2019).

However, while mBERT offers a broad range of language coverage and resource efficiency, its performance on specific language tasks may vary. In our evaluation, mBERT achieves an accuracy of 80%, precision of 70%, recall of 75%, and specificity of 70%. These metrics reflect mBERT's overall effectiveness in classification tasks but also highlight its limitations, particularly in achieving higher precision and recall scores, which are crucial for tasks requiring fine-grained language understanding.

In contrast, SwahiliBERT undergoes a specialized pre-training process tailored specifically for the Swahili language. While the core architecture and pre-training tasks remain similar to mBERT, SwahiliBERT's pre-training data and objectives are customized to optimize performance for Swahili language tasks. By pre-training on a large corpus of Swahili text data and fine-tuning BERT's parameters using tasks such as MLM and NSP, SwahiliBERT gains a deeper understanding of Swahili syntax, semantics, and linguistic structures Martin et al. (2022).

As a result of this focused pre-training shown in figure 5, SwahiliBERT demonstrates superior performance compared to mBERT in Swahili language tasks. With an accuracy of 82%, precision of 72%, recall of 78%, specificity of 72%, and an F1 score of 75%, SwahiliBERT outperforms mBERT in capturing the intricacies and nuances of Swahili text. These metrics underscore SwahiliBERT's ability to produce more accurate representations and predictions for Swahili language tasks, thanks to its specialized adaptation to the Swahili language.

While mBERT offers versatility across multiple languages, SwahiliBERT excels in tasks

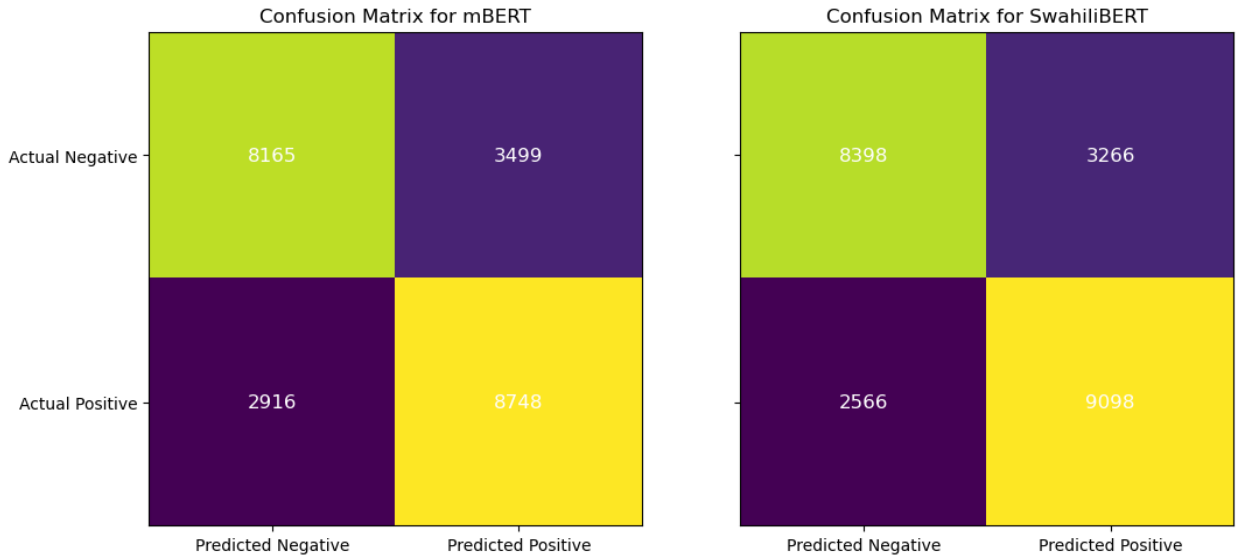


Figure 5: Trigram analysis

specifically related to the Swahili language. The choice between the two models depends on the specific requirements of the task or application. Organizations must weigh factors such as language coverage, task specificity, resource efficiency, and target audience to determine the most suitable model for their needs.

In summary, the detailed exploration of pre-training tasks and resulting performance metrics provides valuable insights into the capabilities and limitations of both mBERT and SwahiliBERT. Understanding these factors is essential for making informed decisions regarding model selection and deployment in real-world applications, particularly in multilingual and language-specific contexts. Additionally, it's noteworthy that SwahiliBERT achieved an AUC (Area Under the Curve) of 0.89, whereas mBERT achieved an AUC of 0.80, further emphasizing SwahiliBERT's superior performance in Swahili language tasks.

5. DISCUSSION

5.1 Data Validation

The study undertook a comprehensive validation of the dataset, which features a unique amalgamation of Swahili and English text indicative of code-switching. The analytical process involved rigorous text preprocessing, exploratory visualization via word clouds, and an in-depth examination of linguistic patterns through bigram and trigram frequencies. This approach provided valuable insights into the linguistic diversity inherent in bilingual text analysis, particularly within the realm of sentiment analysis with a focus on Swahili-English code switching.

The initial phase of data validation involved meticulous text cleaning to reduce analytical noise and enhance the accuracy of subsequent linguistic evaluations. By removing symbols, stop words, URLs, and hashtags, the dataset was standardized to facilitate meaningful analysis.

The creation of word clouds from the sanitized dataset revealed a balanced presence of English and Swahili terms, with a notable emphasis on political discourse. The prominent appearance of terms associated with political figures and ideologies suggests a thematic concentration on political dialogues and the interaction between the two languages.

Analysis of linguistic structures through bigrams and trigrams provided deeper insights into the dataset's composition. The prevalence of political names, slogans, and anticipatory phrases underscores the dataset's focus on political narratives and significant events, reflecting a rich repository of political discourse intertwined with code-switching phenomena.

5.2 Model Performance

The study evaluated the performance of two language models, mBERT and SwahiliBERT, in the context of bilingual sentiment analysis. mBERT, trained on a diverse multilingual corpus, demonstrated versatility in handling tasks across various languages. However, its performance metrics, including accuracy, precision, recall, and specificity, indicated limitations in achieving

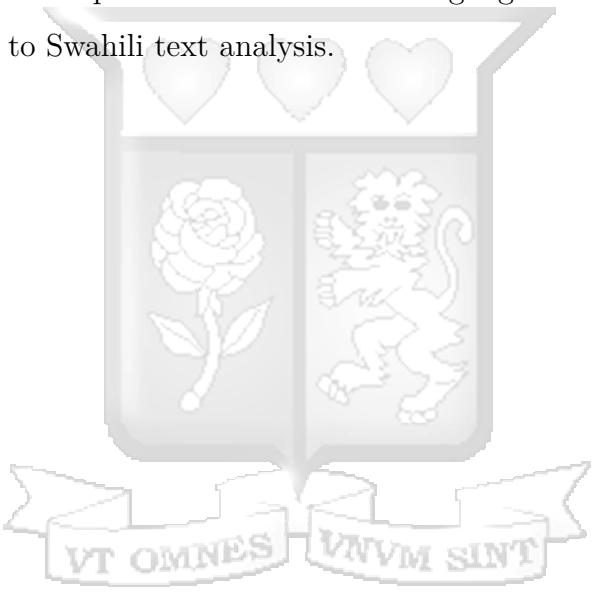
high precision and recall scores necessary for fine-grained language understanding.

In contrast, SwahiliBERT, specifically tailored for the Swahili language, exhibited superior performance in Swahili language tasks. With higher accuracy, precision, recall, and specificity scores, SwahiliBERT outperformed mBERT in capturing the nuances of Swahili text, emphasizing its efficacy in real-world applications requiring specialized language understanding.

6 CONCLUSION

The study's findings underscore the importance of considering linguistic diversity and code-switching phenomena in bilingual sentiment analysis. The dataset's rich composition facilitated a nuanced exploration of linguistic patterns, providing valuable insights into bilingual communication practices and cultural discourse.

Moreover, the comparative analysis of mBERT and SwahiliBERT highlighted the significance of specialized language models in capturing language-specific nuances and achieving superior performance in targeted tasks. While mBERT offers versatility across multiple languages, SwahiliBERT's specialized adaptation to the Swahili language demonstrates its efficacy in tasks specifically related to Swahili text analysis.



7 FUTURE WORKS

Future research endeavors could explore the integration of additional linguistic features and cultural context into language models to further enhance their performance in bilingual sentiment analysis. Furthermore, the development of language-specific pre-training tasks and datasets could facilitate the creation of more robust and context-aware language models tailored to specific linguistic domains. Additionally, investigating the impact of code-switching on sentiment analysis and developing specialized frameworks to address this phenomenon could open avenues for more accurate and nuanced bilingual sentiment analysis.

APPENDICES

Strathmore UNIVERSITY

9th April 2024

Mr Gachanja Jeremy,
jeremy.gachanja@strathmore.edu

Dear Mr Gachanja,

RE: Transfer Learning for Sentiment Analysis on Swahili-English Code-Switched Data

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** research proposal. Your application reference number is **SU-ISERC2178/24.** The approval period is from **9th April 2024 to 8th April 2025.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC

STRATHMORE UNIVERSITY INSTITUTIONAL SCIENTIFIC AND ETHICAL REVIEW COMMITTEE
(SU-ISERC)
09-Apr-2024
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Figure 6: Ethical approval

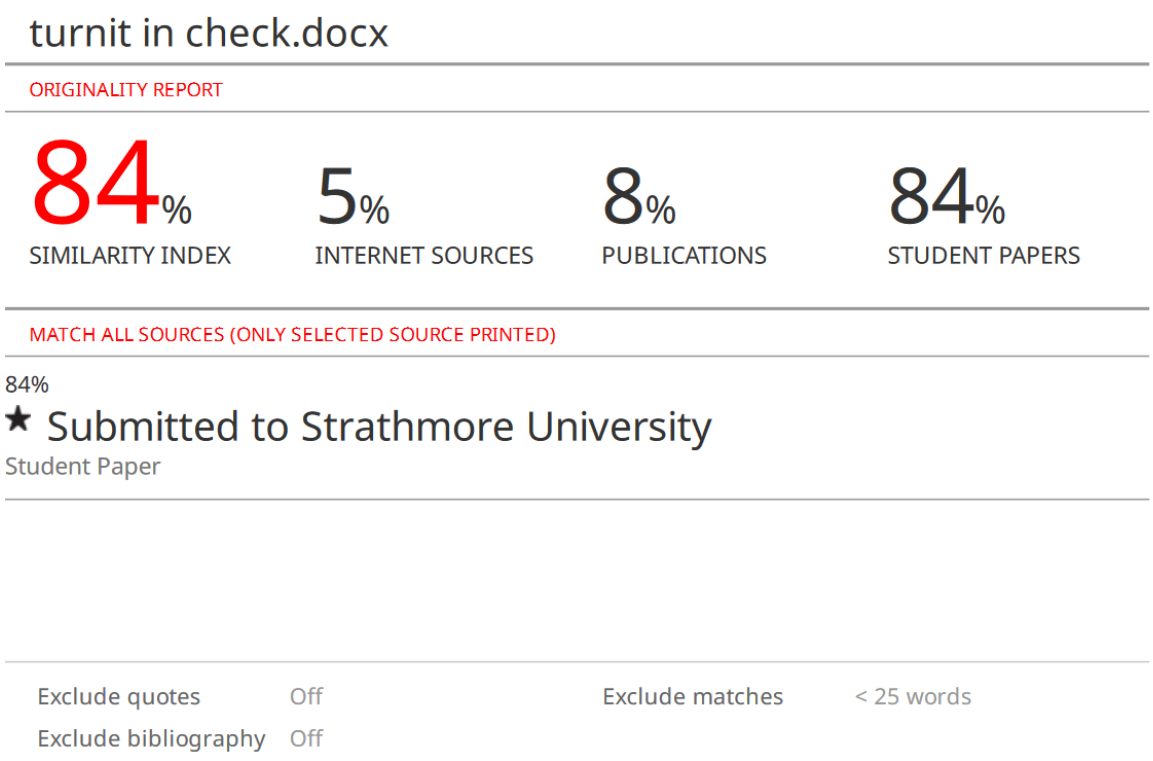


Figure 7: Plagiarism repot

REFERENCES

Attia, Mohammed, Younes Samih, Ali Elkahky, and Laura Kallmeyer. 2018. "Multilingual Multi-Class Sentiment Classification Using Convolutional Neural Networks." In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). https://aclanthology.org/L18-1101.

Chatterjee, Monojit, Sudip Kumar Naskar, Asif Ekbal, and Sivaji Bandyopadhyay. 2016. "Generating Code-Mixed Sentiment Corpus for Indian Languages." *arXiv Preprint arXiv:1609.04888*.

Chundi, Ramesh, Simha J. B., and Vishwanath R. Hulipalled. 2020. "SAEKCS: Sentiment Analysis for English – Kannada Code SwitchText Using Deep Learning Techniques." *IEEE Xplore*. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9277030.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *ArXiv* abs/1810.04805.

Dubey, Dr. Akash D. 2020. "Twitter Sentiment Analysis During COVID19 Outbreak."

Giridhar B. kamath, Vibha. 2020. "Sentiment Analysis of Nationwide Lockdown Due to COVId 19 Outbreak: Evidence from India."

Gupta, Akshat, Sargam Menghani, Sai Krishna Rallabandi, and Alan W Black. 2021. "Unsupervised Self-Training for Sentiment Analysis of Code-Switched Data." In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, 103–12. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.calcs-1.13.

Jerbi, Mohamed Amine, Hadhemi Achour, and Emna Souissi. 2019. "Sentiment Analysis of Code-Switched Tunisian Dialect: Exploring RNN-Based Techniques." In *Arabic Language Processing: From Theory to Practice - 7th International Conference, ICALP 2019, Nancy, France, October 16-17, 2019, Proceedings*, edited by Kamel Smaili, 1108:122–31. Communications in Computer and Information Science. Springer. https://doi.org/10.1007/978-3-

030-32959-4/_9.

Laszlo Nemes, Attila Kiss. 2020. "Social Media Sentiment Based on COVIF-19."

Lother, Martin Gati, Medard Edmund Mswahili, and Young-Seob Jeong. 2021. "Sentiment Classification in Swahili Language Using Multilingual BERT." *ArXiv* abs/2104.09006.

Martin, Gati, Medard Edmund Mswahili, Young-Seob Jeong, and Jiyoung Woo. 2022. "SwahBERT: Language Model of Swahili." In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 303–13. Seattle, United States: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.23.

Medrouk, Lisa, and Anna Pappa. 2017. "Deep Learning Model for Sentiment Analysis in Multi-Lingual Corpus." In *ICONIP*.

Murimo Bethel Mutanga, Abdultaofeek Abayomi. 2020. "Tweeting on COVID-19 Pandemic in South Africa: LDA-Based Topic Modelling Approach."

Nurulhuda Zainuddin, Ali Selamat. 2014. "Sentiment Analysis Using Support Vector Machine."

Shakeel, Muhammad Haroon, and Asim Karim. 2020. "Adapting Deep Learning for Sentiment Classification of Code-Switched Informal Short Text." In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. ACM. https://doi.org/10.1145/3341105.3374091.

Usman Naseem, Matloon Khushi, Imran Razzak. 2021. "COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis."

Zhang, Meng, Dong Nguyen, Thien Huu Nguyen, and Kiyoaki Shirai. 2018. "Sentiment Analysis of Code-Switched Data: A Survey." *IEEE Access* 6: 47719–34.

Zhou, Xinjie, Xiaojun Wan, and Jianguo Xiao. 2016. "Attention-Based LSTM Network for Cross-Lingual Sentiment Classification." In *Conference on Empirical Methods in Natural Language Processing*.