ANALYSIS OF EAST AFRICAN SWAHILI NEWS

Leveraging NLP and Deep Learning for Automated Classification of East African Swahili News

MORINGA DS -FT 11

GROUP VI PROJECT

PRESENTED BY :

1. Rodgers Ndemo
2. Catherine Kipkirui
3. Amani Mkaya
4. Adnan Mohamud
5. Bernice Kutwa

TABLE OF CONTENTS

ANALYSIS OF EAST AFRICAN SWAHILI NEWS

1. INTRODUCTION

1.2.0. Project overview

News media plays a crucial role in shaping public opinion, informing societies, and influencing political and social discourse. In East Africa, Swahili is one of the most widely spoken languages, serving as a unifying medium for news dissemination across multiple countries, including Tanzania, Kenya, Uganda, Rwanda, Burundi, and the Democratic Republic of Congo.

With the rise of digital journalism, there has been a rapid increase in Swahili news content, necessitating the need for automated classification of Swahili news articles. This project aims to

leverage Natural Language Processing (NLP) and Deep Learning to develop a model that can accurately categorize Swahili news content into predefined categories

1.2.1 Business overview

The growing volume of Swahili-language news presents a challenge for media organizations in managing, organizing, and delivering relevant content efficiently. Manual classification is labor-intensive, time-consuming, and prone to inconsistencies, leading to inefficiencies in news dissemination. To address this, automated categorization using Natural Language Processing (NLP) and Deep Learning offers a scalable and reliable solution. By leveraging these technologies, news platforms, media houses, and content aggregators can streamline content management, enhance searchability, and improve audience engagement.

1.2.2 Problem statement

The goal of this project is to build a Swahili news classification model that accurately categorizes news articles into six predefined categories:

- uchumi (economy)
- kitaifa (national news)
- michezo (sports)
- kimataifa (international news)
- burudani (entertainment)
- afya (health)

To achieve this, we preprocessed the text data to remove noise, tokenize, and normalize the text, followed by building a classification model. To ensure a clear, reproducible, and scalable approach, we implemented the preprocessing steps within an Scikit-learn Pipeline.

1.2.3 Objectives

- **Automating News Classification.**

How can we efficiently categorize Swahili news articles using machine learning and deep learning?

- **Understanding Media Trends**.

What are the most common news topics in East African media?

Are there any biases in media coverage based on classification trends?

- **Enhancing Content Accessibility.**

How can automated classification improve information retrieval for journalists, policymakers, and the general public?

1.2.4 Project Plan

- We have used Jira List - GROUP 6 PHASE4 - Jira to assign tasks to members and follow up on progress and the general project management
- We have used github collaboratory as our version control system for the project ● We have used CRISM DM to compile the data report

2. DATA UNDERSTANDING

2.1 Overview

This dataset consists of Swahili news articles categorized into different topics. The goal is to classify news articles into predefined categories using Natural Language Processing (NLP) and Deep Learning techniques.

2.1.0 Data Structure

The dataset contains the following columns: **id**: A unique identifier for each news article.

content: The text of the news article written in Swahili. **category**: The label representing the

category of the news article (e.g., uchumi, kitaifa, michezo).

2.2.1 Categories in the Data

The dataset has multiple categories representing different types of news. Some of the common categories include:

Uchumi (Economy): Articles related to business, finance, and economic activities.

Kitaifa (National News): General news related to Tanzania.

Michezo (Sports): News about sports teams, events, and athletes.

2.2.2 Data Size and Distribution

To understand the dataset better, key aspects to analyze include:

The **total number of articles**.

The **distribution of articles** across different categories (class imbalance analysis).

The **average length of articles** in terms of word count.

2.2.3 Data Quality Issues

Potential issues to check before preprocessing:

Missing values: Are there any missing or empty fields?

Duplicates: Are there repeated news articles?

Class imbalance: Are some categories significantly overrepresented compared to others?

Noise in text: Presence of irrelevant characters, symbols, or stopwords that may need cleaning.

2.2 Data Collection

The dataset we used for this analysis was from https://www.kaggle.com/datasets/thedevastator/east-african-news-classification a platform for data science competitions ,where data scientists and machine learning engineers can compete to create the best models for solving specific problems .

2.3 Data Summary

From checking the train dataset we observed that:

1. The data maintained uniformity from top to bottom.
2. The columns were: (id, content, category).
3. The train data had 23268 entries.

From the test dataset:

1. The data also maintained uniformity.
2. The columns were (text and label).

   This set of data did not have an id column.However the text and content columns are similar and also label and category.

3. The test data had 7338 entries.

2.4 verifying Data Validity

To ensure the dataset is suitable for analysis and modeling, we conducted the following data validity checks:

1. **Handling Missing or Null Values**

   ○ We examined the dataset for missing values in critical columns such as category labels and text content. No missing values were found .

2. **Category Distribution Analysis**

   ○ The class distribution visualization showed an imbalance, with "kitaifa" and "michezo" having the highest number of samples while "afya" had the least.

3. **Text Length Consistency**

   ○ The average word count and text length vary across categories. The "afya" category has the longest average text length, while "burudani" has the shortest.

   ○ We confirmed that there are no extreme outliers or incorrectly formatted texts.

4. **Duplicate Entry Check**

    - A check for duplicate articles was performed, and no redundant entries were found that would cause a model bias.

5. **Label Accuracy Validation**

    - A preliminary review of some sample articles confirmed that the category labels generally match the content. However, further manual verification or a data annotation process could enhance accuracy.

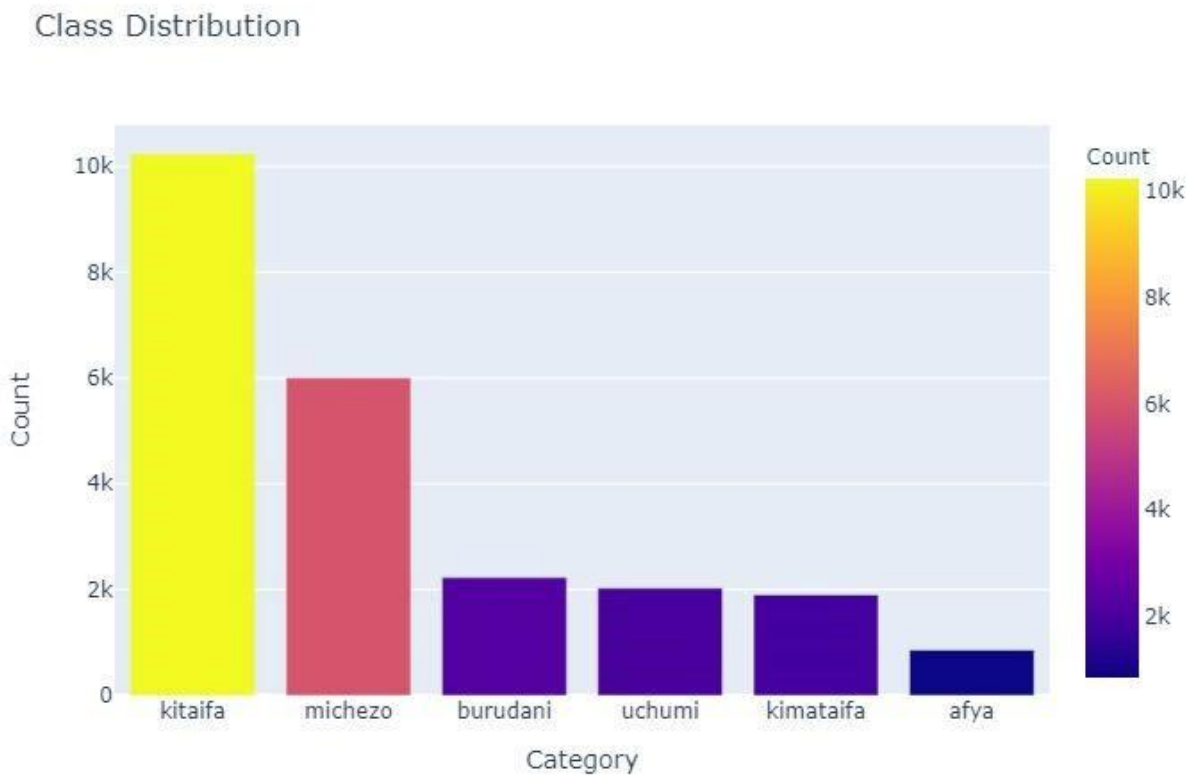
6. **Text Encoding & Formatting**

    - The dataset was reviewed for encoding issues, such as special characters or formatting inconsistencies, and necessary corrections were made to ensure smooth text processing.

By addressing these data validity concerns, we ensured that the dataset is clean, reliable, and ready for further analysis and modeling.

2.5 .Exploratory Data Analysis

2.4.1 UNIVARIATE ANALYSIS



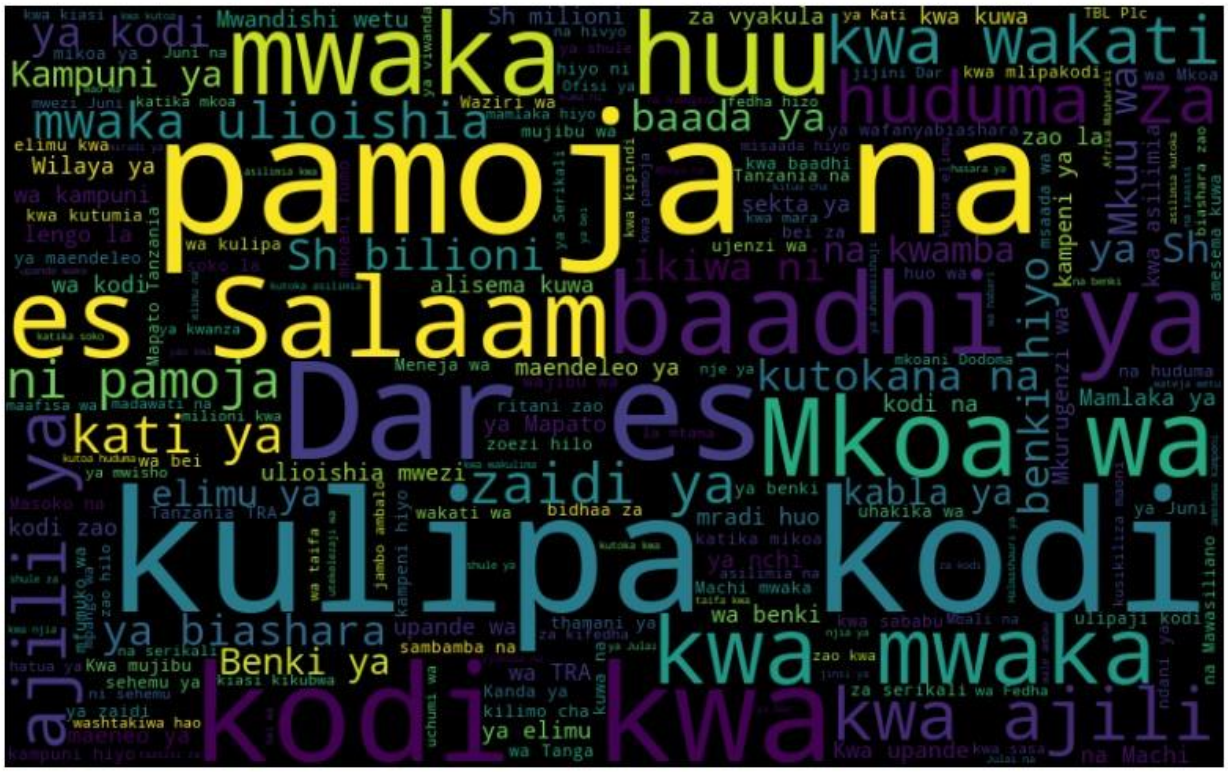
Insight:

From the above kitaifa was the most frequent category with approximately 10,000 samples. The occurrence in order of frequency is as follows:

1. Kitaifa - 10242 entries
2. Michezo - 6004 entries
3. Burudani - 2229 entries
4. Uchumi - 2028 entries
5. Kimataifa - 1906 entries
6. Afya - 859 entries

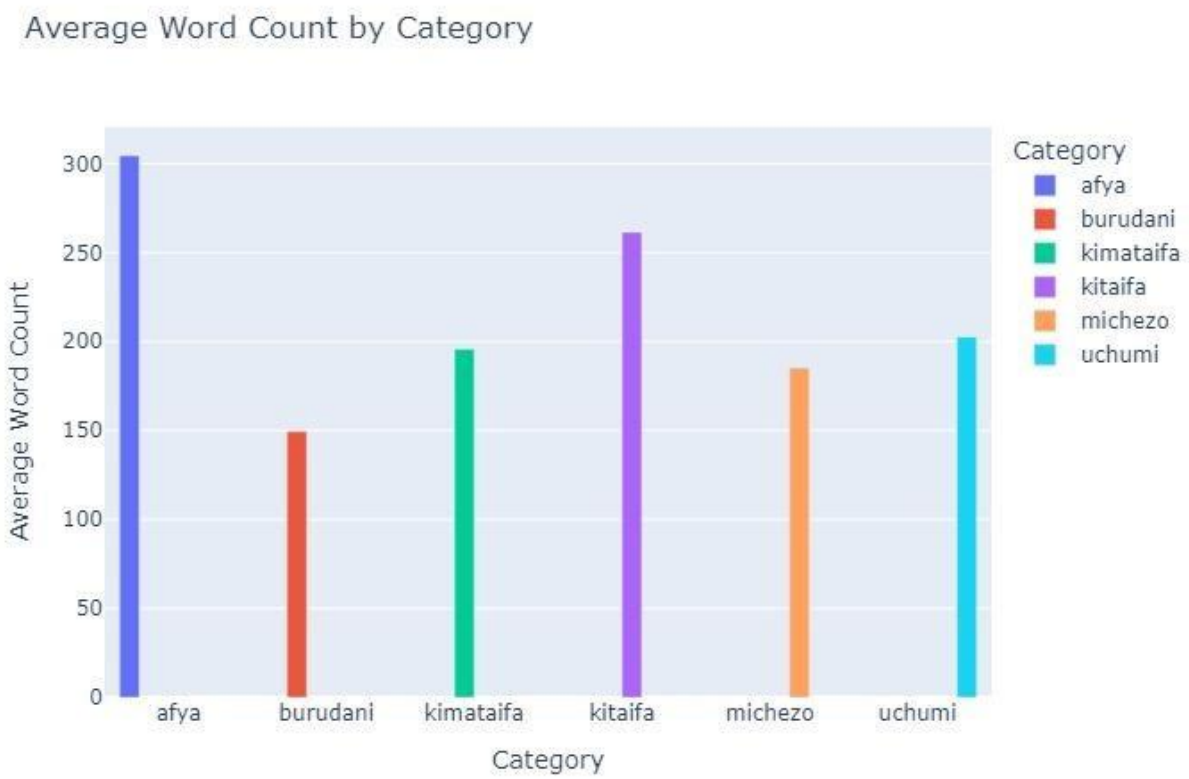
This explains that the top most frequent category of news was Kitaifa(Nationally) to mean that general news were being watched followed by Michezo (Sports),Burudani(entertainment) followed by the rest

We had a **wordcloud** for **each category** and we shall showed a sample below:



From the above wordcloud, we can observe that the frequent words in this category include: kulipa, kodi, benki, kampuni etc

2.4.2 BIVARIATE ANALYSIS



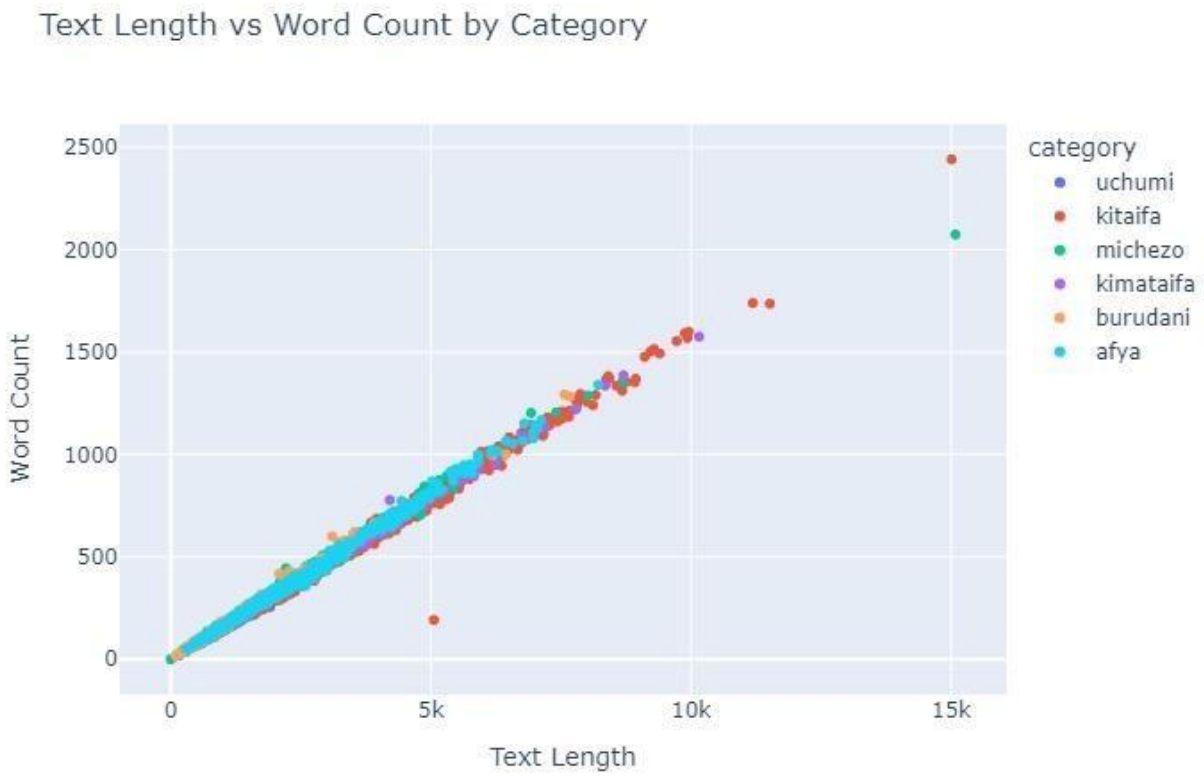
The above diagram showed the average word count by category, which revealed interesting differences between categories:

Insights:

Higher Average Word Count for "afya" and "kitaifa".The afya category has the highest average word count (~300 words), suggesting that articles in this category are typically longer and more detailed. The average word count is as follows in descending order:

1. Afya ~ 305
2. Kitaifa ~ 262
3. Uchumi ~ 203
4. Kimataifa ~ 196
5. Michezo ~ 186
6. Burudani ~ 150

2.4.3 MULTIVARIATE ANALYSIS



The above diagram of a scatter plot showed a strong positive correlation between text length and word count across different categories. Here's what it suggests about the data:

Key Insights:

1. **Linear Relationship**

The almost perfect diagonal alignment indicates that text length and word count increase proportionally — longer texts have more words, which is expected since text length typically reflects character count.

2. **Category Consistency**

The consistent clustering of categories suggests that the relationship between text length and word count holds across different categories — no major outliers by category.

3. **Outliers**

There are a few points (especially at high text lengths) that are slightly off the line — these could be unusual texts with higher or lower word density (e.g., many special characters or formatting differences).

4. **Data Quality**

The clean clustering implies that the data is relatively clean and well-structured, without major inconsistencies across categories.

3. MODELING

3.1 Preprocessing

This step involved preparing the clean data for modeling.

1. Creating pipelines for vectorizing the feature and label encoding the target.
2. Calling the train and test, Xand y for use in model training and testing

3.1.0 Pipelines

Machine learning and deep learning models do deal with texts hence the need to change the text to a format in which the models would be able to handle the text. Term Frequency-Inverse Document Frequency Vectorizer converts text data into numerical feature vectors that can be used by machine learning and deep learning models. TF-IDF is useful because:

1. Terms that are unique to a document are given higher weight.
2. Terms that are common across all documents (like "za" or "na") are down weighted.
3. It helps improve the relevance of features for text classification and clustering.1.

Label Encoding for Target Variable

Since our target variable is categorical, we need to convert the class labels into a numerical format that machine learning models can understand. We used label encoding, which assigns a unique integer to each category.

Implementation

We applied label encoding using sklearn.preprocessing .LabelEncoder.The encoded categories are as follows:

Original Category	Encoded Value
Afya	0
Burudani	1
Kimataifa	2
Kitaifa	3
Michezo	4
Uchumi	5

This encoding ensured that our machine learning model could process the target variable as numerical data.

3.1.1 Actual modelling

This part presents an evaluation of various machine learning models applied to the Swahili news classification task.

The goal was to develop a robust model that effectively categorizes news articles into their respective classes.

3.1.1.0 Dimensionality Reduction Attempts

Initially, we attempted Principal Component Analysis (PCA) for dimensionality reduction, but our data did not allow for its application. Instead, we tried Singular Value Decomposition (SVD), which did not significantly improve performance.

3.1.1.1 Hyperparameter Tuning & Model Performance

We optimized and evaluated three machine learning models:

1. **Naïve Bayes**

- Best Hyperparameters: **alpha =0.1**

- Performance:
    - Accuracy: **82.57%**
    - Macro F1 Score: **0.749**
    - Weakest Class: **"Kitaifa"** (F1-score: 0.4468)
    - Strongest Class: **"Kimataifa"** (F1-score: 0.9424)

2. Logistic Regression

- Best Hyperparameters: **C =1 , solver = 'liblinear**'
- Performance:
    - Accuracy: **79.56%**
    - Macro F1 Score: **0.7956**

3. **Linear SVM**

- Best Hyperparameters : **C = 0.1**
- Performance:
    - Accuracy: **79.95%**
    - Macro F1 Score: **0.7995**

4. **Model Selection & Ensemble Learning**

After evaluating individual models, we combined them using an ensemble approach to improve classification robustness.

Ensemble Model Performance:

- Accuracy: 84.78%
- Macro F1 Score: 0.7401
- Improved class balance compared to individual models

We faced a few challenges with this and these were our next steps:

- Overfitting: Despite hyperparameter tuning and ensemble modeling, models still showed signs of overfitting.
- Data Complexity: Given the high dimensionality and complexity of the dataset, traditional models were not sufficient.

Due to these challenges we went further into Deep Learning models, as they would better capture the intricacies of the Swahili news dataset.

4. DEEP LEARNING

We used a Long Short-Term Memory (**LSTM)** model, which is a specialized form of RNN designed to handle sequential data. LSTMs are particularly well-suited for text-based tasks because they can capture the context and long-term dependencies between words. This makes them highly effective for tasks such as sentiment analysis, text classification, and language modeling.

The model will process the input text data, learn the underlying patterns, and classify the text into predefined categories. By leveraging the power of deep learning, the model aims to achieve higher accuracy and better generalization than traditional machine learning approaches.

4.1.0 Model observations and next steps

Learning Curve Trends

1. Training Loss: Gradually decreasing, indicating that the model is learning.
2. Validation Loss: Remained almost flat, suggesting poor generalization.
3. Training Accuracy: Increased slowly but still low, showing that the model struggles to capture complex patterns.
4. Validation Accuracy: Stayed fixed around 45.64% across epochs — a sign of underfitting.

Observations & Issues

1. Model is learning slowly.
2. No sign of overfitting (training and validation performance are closely aligned).
3. Validation accuracy stagnating at around 45% suggests the model is underfitting.

The problem could be due to:

1. Insufficient complexity in the model.
2. Learning rate might be too high or too low.
3. Training data imbalance or not enough data.

Recommendations for Improvement

The model showed signs of **underfitting**, with validation accuracy stagnating at **45.64%** and training loss decreasing slowly. There were no signs of overfitting, but the model struggled to capture complex patterns. Potential causes included **insufficient model complexity, suboptimal learning rate, or data imbalance**.

Increase Model Complexity: Increase the number of LSTM units or the number of dense layers.

4.1.1 Tuned LSTM Model

Observation

Test Accuracy (45.62%) was moderate but unsatisfactory, indicating learning difficulties.

High Test Loss (1.4286) suggested poor class separation and underfitting.

Severe Class Imbalance: The model predicted **class 3** for almost all inputs.

High Misclassification Rate: The model failed to distinguish between other classes.

We could not tune this model further as it did not promise yielding any better performance

4.1.2 BiLSTM model

A Bidirectional Long Short-Term Memory (BiLSTM) model is an extension of the traditional LSTM model. The key difference is that a BiLSTM processes information in both forward and backward directions, which allows it to capture context from both past and future states in a sequence.

We decided to use it and these are our findings:

Stagnant Validation Accuracy: Stuck at **45.64% from epoch 2 onward**, indicating limited learning progress.

Slow Training Loss Reduction: Training loss decreased, but validation loss remained flat, suggesting poor generalization.

Low Overall Accuracy: Training accuracy (~44%) and validation accuracy (45.64%) indicated the model struggles to capture deeper semantic patterns.

We still went ahead and used BERT Model

4.1.3 BERT Model

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based deep learning model designed for natural language processing (NLP) tasks. It was introduced by Google AI in 2018 and quickly became the state-of-the-art for a wide range of NLP tasks.

After Fine Tuning the model :

Justification of the Model (Fine-Tuned BERT-base-uncased for Sequence Classification)

- **High Accuracy (89.56%)**: The model correctly predicted the target class in nearly 9 out of 10 cases, indicating strong pattern recognition and reliable predictions.
- **High F1 Score (89.09%)**: A balanced measure of precision and recall, confirmed the model minimizes false positives and negatives while maintaining accuracy.
- **Small Gap Between Accuracy and F1 Score**: The close alignment suggested consistency in predictions and no significant class imbalance.
- **Strong Generalization**: High validation performance confirmed the model's ability to understand and interpret unseen data effectively, leveraging BERT's deep contextual learning.

5. CONCLUSION

1. **Average Text Length by Category:**

The "afya" and "kitaifa" categories have the longest average text lengths.

Significant variation in text length across categories suggests that longer texts may introduce complexity in model training.

2. **Text Length vs. Word Count:**

Strong linear relationship between text length and word count, which is expected.

3. **Average Word Count by Category:**

The "afya" category has the highest average word count, indicating longer and possibly more complex content.

The "burudani" category has shorter texts, which could lead to reduced context in model predictions.

4. **Best classifier**:

Fine-Tuned BERT-base-uncased for Sequence Classification achieves **high accuracy (89.56%)** and a strong **F1 score (89.09%),** indicating that **it effectively captures and interprets complex language patterns.** The high scores reflect the model's strong generalization ability, demonstrating that it can perform well on unseen data without overfitting.

6. RECOMMENDATION

1. **Optimize Content length for categories:**

The variation in average text length and word count by category suggests differences in content consumption patterns.

Focus on creating balanced content. Shorter and more concise for categories like "burudani" and more detailed for categories like "afya" and "kitaifa."

2. **Enhance User Engagement:**

The high text length and word count in certain categories (e.g., "afya") indicate that users may engage better with longer content in these areas.

Tailor content length based on user preferences to improve retention and interaction.

3.Automate Content Classification:

Automate content tagging and classification to improve searchability and content recommendations using the **Fine-Tuned BERT-base-uncased for Sequence Classification**