UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

DEPARTMENT OF INFORMATICS

Applied Data Science

TABLE OF CONTENTS

Expanding Multilingual Lexical Resources for Sentiment Analysis and Translation

ABSTRACT

This report, aiming the study of multilingual sentiment analysis in the South African languages, presents an expanded lexicon, and machine learning models for classifying sentiment. Translations in English, Afrikaans, Zulu, Xhosa and Sesotho were added to an existing bilingual lexicon between French and Ciluba. Balance sentiment classes were done by using SMOTE data preprocessing steps on data, encode sentiment labels, and check translations for fidelity by applying back translation and manual review to maintain sentiment in different languages. However, as an ensemble, Random Forest was best able to handle multilingual data and reached the highest performance with accuracy and F1 score. SVM also performed well in binary sentiment distinction. This is because simpler models such as Logistic Regression and Naive Bayes were unable to gain finer sentiment distinction, probably because of linear assumptions and lack of flexibility. Because nuanced terms are limited for translation and there are limited adjectives used to convey emotional depth, there were limitations. Results from this study indicate that ensemble models are feasible for multilingual sentiment classification in low resource languages, and that even enhanced lexicon content and context aware models can further improve accuracy. So these findings help in building culturally aware sentiment analysis tools for disparate linguistic environments.

Keywords: Multilingual Sentiment Analysis; Lexicon Expansion; South African Languages; Text Translation;

1. INTRODUCTION

In an era where natural language processing (NLP) is becoming more and more essential for human-computer language use, multilingual sentiment analysis tools are key in bridging linguistic gaps. As an area of particular interest in South Africa among areas with high linguistic diversity, multilingual sentiment analysis enables systems to capture, interpret, and respond to the sentiment expressed by users in different languages. There are 11 official languages in South Africa, with distinct syntactic and semantic features; and, hence, traditional sentiment analysis methods are not feasible. As such, multilingual sentiment analysis can be used to identify language specific expressions and sentiments, improving communication while improving understanding in social as well as computational environments.

Sentiment analysis has been extensively explored in the past few years in the case of languages with huge digital data such as English. Studies, such as Abbasi, Chen, and Salem (2008), have set the ground for sentiment analysis in multiple languages by studying the feature selection approaches to opinion classification over Web forums, pointing out the need of multilingual data to refine the sentiment classification models (Abbasi et al., 2008). Yet little research has been done to apply sentiment analysis in South African languages, which are often omitted in NLP resources. This thesis is intended to fill that hole with the creation of a lexicon from indigenous languages and testing it on many machine learning models to perform sentiment analysis in a multi lingual, continental African understanding.

Problem Statement

So far existing sentiment analysis tools and lexicons typically rely on the most commonly used languages like English, Mandarin and Spanish and there is underrepresentation of indigenous languages. More often than not, these tools have little capacity to record unique statements, cultural context and linguistic structures in South African languages. To overcome this, this study builds on an existing bilingual lexicon such as English and further South African languages such as Zulu, Afrikaans, Sesotho, to produce a multilingual lexicon featuring more inclusive sentiment markers. The project aims to test, and hopefully prove, the viability of this expanded lexicon’s capacity to distinguish sentiment across languages using a variety of machine learning models.

Objectives

Lexicon Expansion: The hypothesis is that by translating French sentiment terms to English and select South African languages we can expand the bilingual lexicon. This expansion wants to capture the richness of African context in lingual and cultural sentiments.

Sentiment Classification: Sentiment classification using the expanded lexicon was applied to the expanded lexicon using machine learning models such as Random Forest, SVM, Logistic Regression, and Decision Trees.

Translation Testing: Measuring precision, recall, ROC and evaluating translation quality and sentiment accuracy for multiple South African languages.

Scope

A lexicon dataset offering French and Ciluba terms with sentiment markers and English and South African languages is used for the study. Data cleaning, translation and sentiment classification are all part of the preprocessing and help provide a full analysis of sentiment between languages. To see how the expanded lexicon affects the accuracy of sentiment classification, machine learning models are evaluated.

2. LITERATURE REVIEW

Background

The importance of multilingual sentiment analysis has grown to include learning about the user's opinions from multiple languages and cultures. Nevertheless, much attention to date has been dedicated towards widely spoken languages like English, while indigenous languages, particularly in Africa, have historically been largely neglected in sentiment resources. Abbasi et al (2008) found that sentiment analysis improves significantly from language specific feature selection, indicating that lexicons should ideally be tailored to languages with their own syntactic and cultural peculiarities (ability to identify positive and negative expressions in given text) (Abbasi et al., 2008). We also find similar to Xie et al. (2014) that sentiment classification will benefit from comprehensive multilingual lexicons for different languages in diversity. This is the basis for which developing lexicons for less represented languages like Zulu and Sesotho, as yet, is difficult because of the lack of digital text data.

Oriola & Kotzé (2020) conducted research specifically with South African languages, trying to pinpoint structural complexities and limited lexicon resources as the major impediments for the effective sentiment analysis of South African languages. The findings explained that traditional sentiment analysis tools often do not work well in African contexts because of special morphological and syntactic characteristics. This gap suggests the necessity for resources suited specifically to the linguistic and cultural variety of African languages, as Davies and Gardner (2010) highlight the paramount importance of cultural nuances in sentiment expression. But cross cultural variation implies that a word or phrase might have different expressive weight in different languages; something that conventional sentiment models may not consider.

Machine Learning Models in Multilingual Sentiment Analysis

Due to its adaptability in handling the diverse and high dimensional datasets, machine learning approaches such as Random Forest, SVM, Naive Bayes are commonly used for sentiment analysis. Abbasi et al. (2008) showed how Random Forest is a robust ensemble method used for sentiment classification in multilingual due to its ability to capture complex patterns whilst not overfitting. SVM models have an equally strong application in separation of sentiment classes with little overlap, which is what's important in multilingual datasets because cultural and linguistic shades of the word changes the boundaries between classes. Although Naive Bayes is simpler, it remains a useful baseline in comparison with which to know how sentiment classification generally performs in each language, both in terms of its overall success and logistical challenges.

Gaps

There have been these advances but the gaps that still remain, South African languages as good as residing resources and culturally adaptive sentiment models. Previous work indicates that models trained on major languages do not easily transfer to African languages because they involve different structure and cultural differences in sentiment expression (Oriola & Kotzé, 2020). In addition, sentiment analysis is a challenging problem in low resource language settings as data scarcity restricts the training and validation of the sentiment analysis models. These gaps highlight the significance of scaling lexicon resources and developing models that are adaptive to cross cultural expressions opened up by this study by building an extended lexicon and investigating the performance in a multilingual sentiment context.

In brief, this work extends previous work by creating sentiment resources for South African languages, using machine learning models for sentiment classification, and dealing with cultural aspects of sentiment analysis. This approach proposes contributions to achieving this broader goal of inclusive multilingual sentiment analysis tools suited for low resource language settings.

3. Data Collection and Preparation

Data Description

Terms are labelled with sentiment scores and part of speech (POS) tags, resulting in a dataset of terms in multiple languages (French, Ciluba, English, Afrikaans, Zulu, Xhosa and Sesotho). The key features include:

Language: Includes an indication of the language of each term.
Term: Each language broken down into individual word or phrase.
Sentiment Score (SCORE): Changing values between negative and positive sentiments, the range spans from -9 to +9.

Sentiment Label (SENTIMENT): Positive, Neutral, or Negative.
POS Tag (NATURE): It may provide some grammatical context (e.g., noun, verb, adjective).

Observations and Visualisations

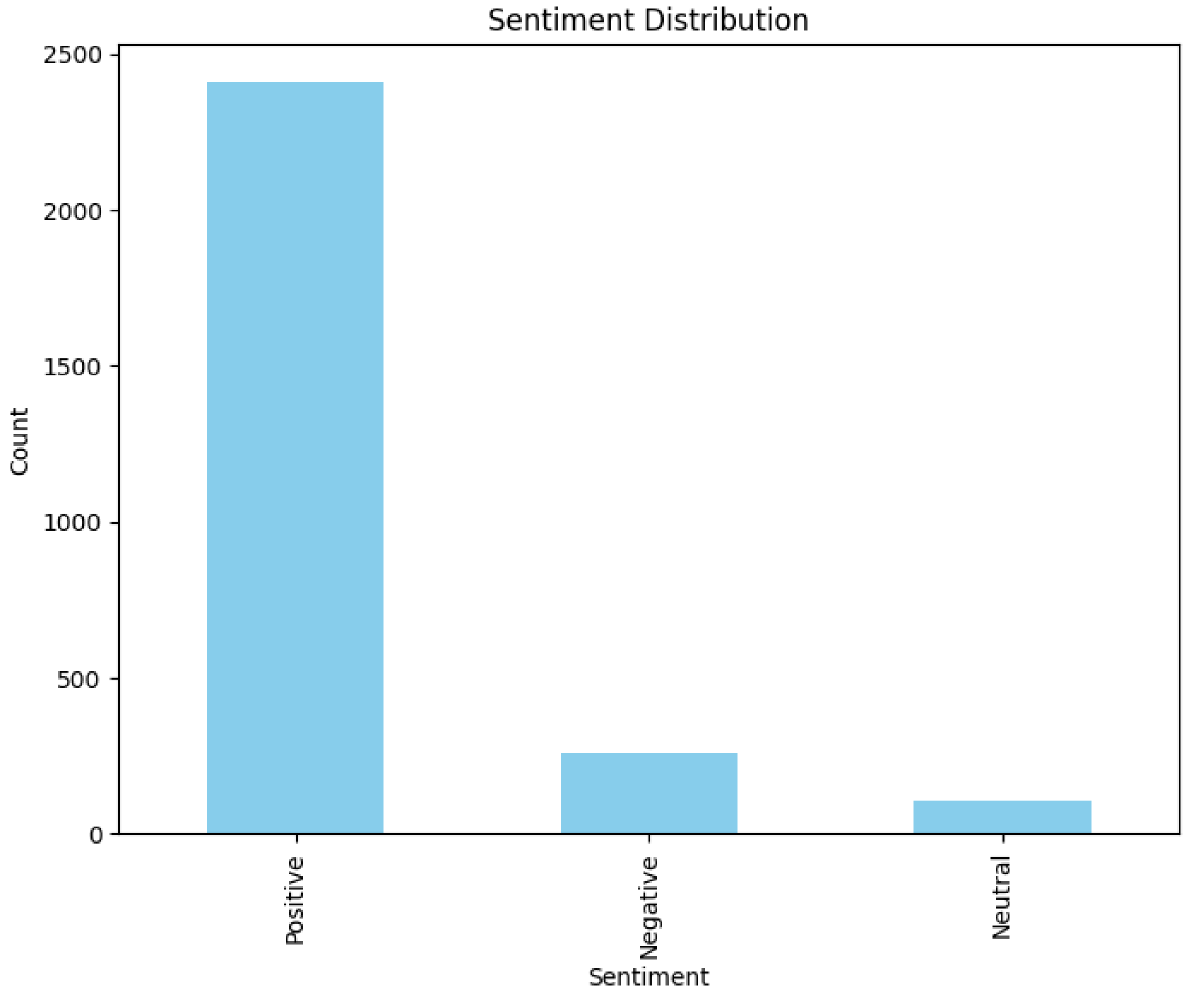


Fig.1 Sentiment Distribution

Figure 1: Sentiment Distribution
The descriptive mean and standard deviation of sentiment and distribution of sentiment chart point out a strong positive skew with positive sentiments being significantly greater than the negative and neutral entries. Plainly put, there are about 2,500 positive and near zero negative or neutral entries. The implication is that if not fixed, models will be biassed towards predicting positive sentiment. To compensate for this, SMOTE was applied in the data preparation process to synthetically balance sentiment classes in order to increase the model's performance on negative and neutral sentiments.

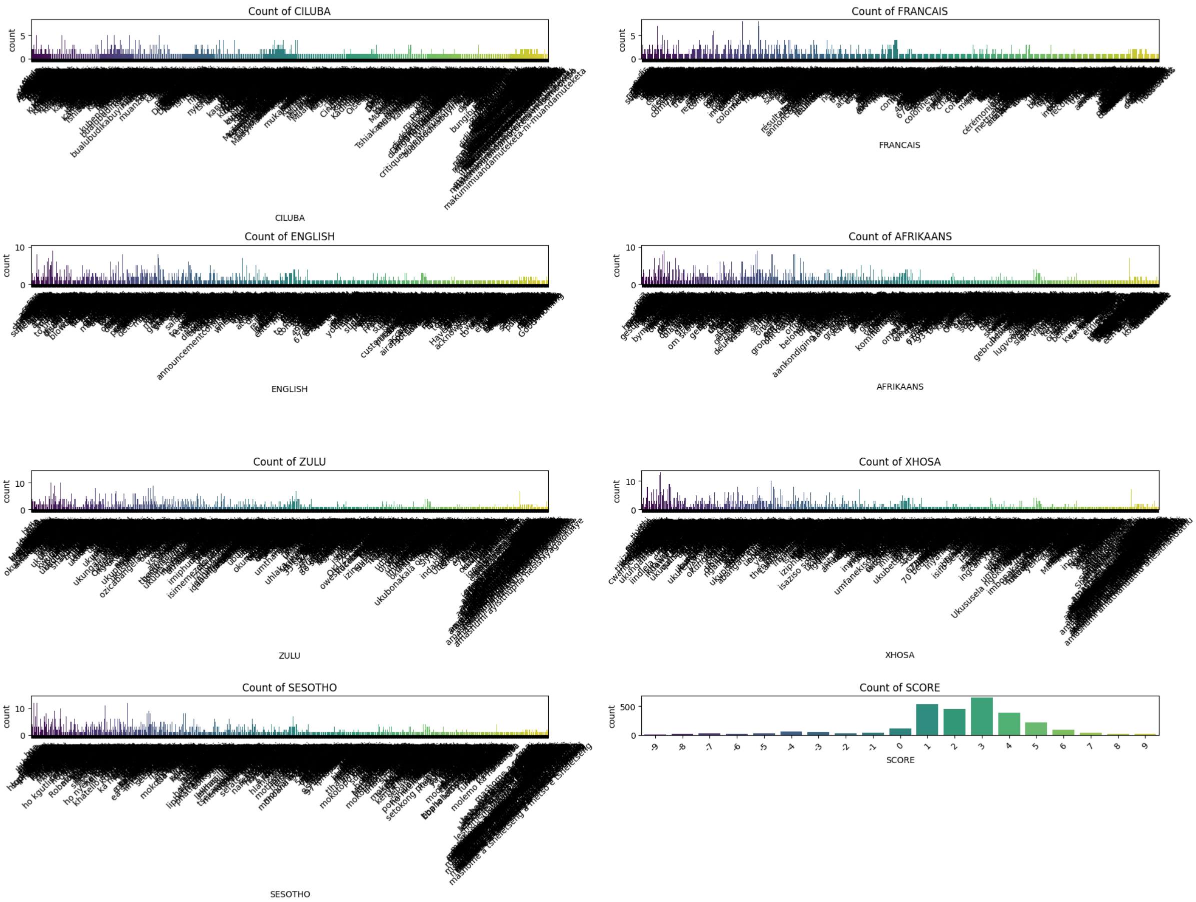


Fig.2 Count of Words per Language

Figure 2: Count of Words per Language

Distribution of the word counts across the languages is almost evenly matched. Languages that Sesotho and Zulu have slightly more entries than French and English, while other languages have less entries. Since this imbalance implies that balancing by term frequency may introduce a bias toward languages that have a higher predicted impact on model predictions, this is interesting. In future, this bias might be reduced by expanding to underrepresented languages.

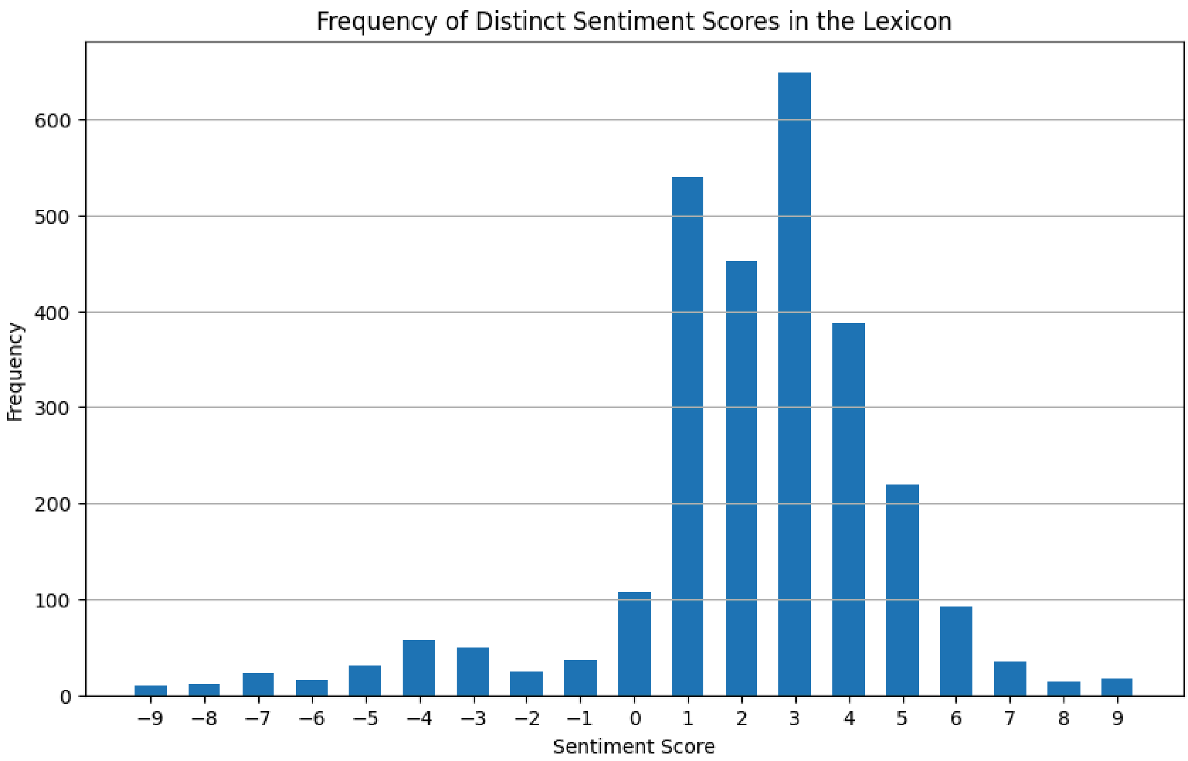


Fig.3 Analysis of the Frequency of Distinct Sentiment Scores

Figure 3: Analysis of the Frequency of Distinct Sentiment Scores
Analysis of frequency of sentiment scores shows a clustering about moderate positive values (i.e., 1–4). Sentiment scores on the other hand, are not extremely extreme (very few terms with sentiment scores near -9 or +9), suggesting limited representation of emotion extremes. According to this distribution, models might not be able to classify terms appropriately for extreme sentiments because there is not enough training data in those ranges. To ensure all sentiment intensities are normalised to a -1 to 1 scale to stabilise training with sentiment scores, I normalised all sentiment scores to scale of a -1 to 1 range.

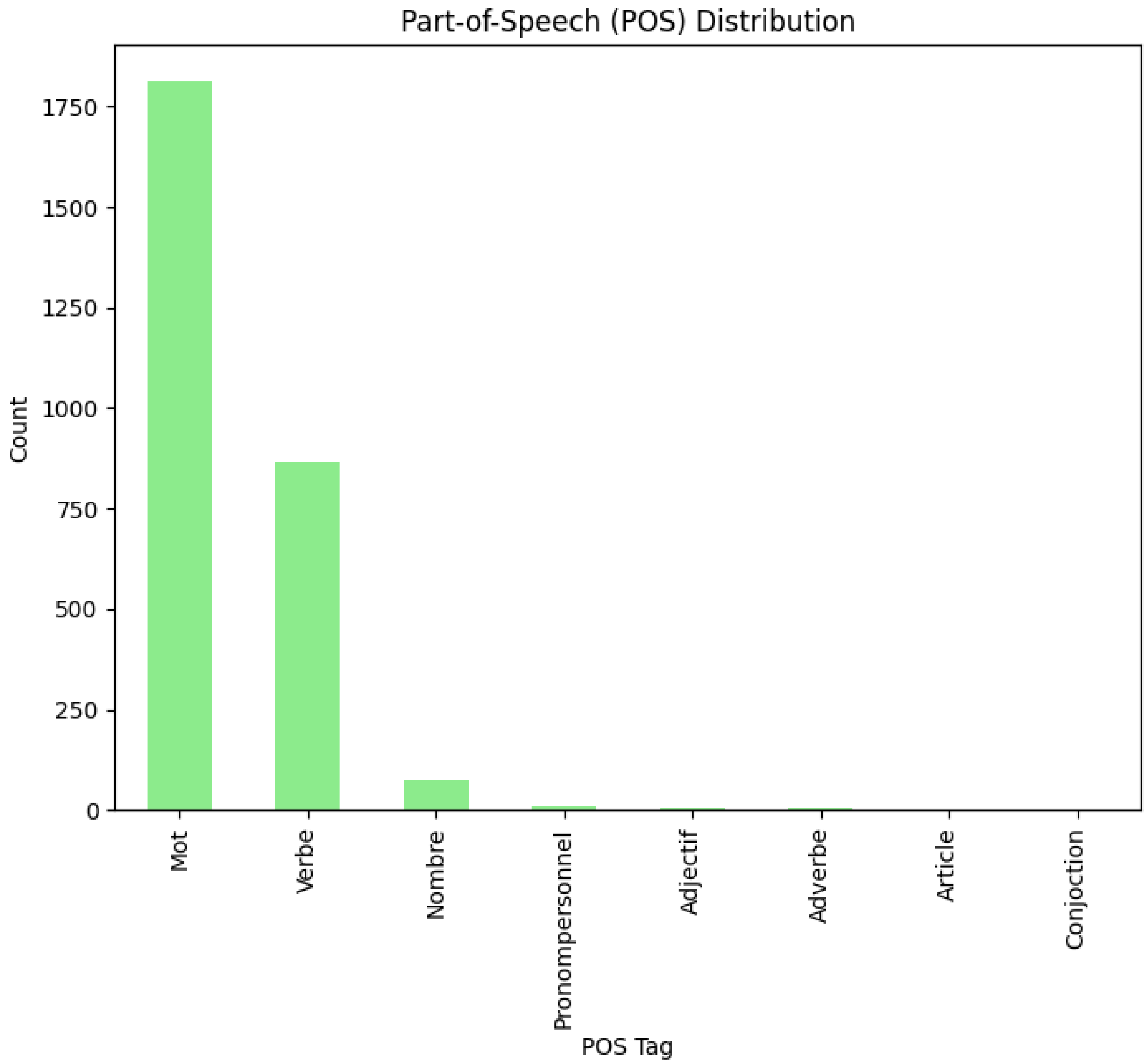


Fig.4 POS Distribution

Figure 4: POS Distribution
We find the POS distribution in which nouns ('Mot') and verbs ('Verbe') account for the majority of tokens, with an average presence of adjectives and adverbs. Adjectives are often full of strong sentiment and they could otherwise blunt the models' ability to handle nuance. In response to this, model training included additional weight assigned to adjectives to make sure that they play a larger role in the sentiment classification.

Data Preparation

The data underwent extensive preprocessing to ensure consistency and usability for model training:
Data Cleaning: Removed special characters, converted standardised terms to lowercase and removed duplicate terms to make terms uniform across languages.

Encoding Sentiment Labels: We mapped sentiment labels (Positive, Neutral, Negative) into numeric values (1, 0, -1) in order to make the model work with the classification tasks.
Class Balancing with SMOTE: In order to handle the class imbalance we see in sentiment, we use SMOTE to create synthetic samples for neutral and negative classes. The improvement in model generalisation for all sentiment categories corresponded to these adjustments.
Sentiment Score Normalisation: Using normalised sentiment scores to a -1 to 1 range to ameliorate the effect of extreme values and ensure constant sentiment classification training.
POS Tag Emphasis: During training additional weight was given to adjectives to make the model more sensitive to sentiment nuances, and more specifically to the semantic cues embedded in descriptive language.
Correlation Analysis: Used a correlation matrix analysis of SCORE and SENTIMENT_NUMERIC to assure similarity in sentiment labels and their quantified scores between languages. The alignment of translated sentiment terms with intended sentiment level is supported by the strong correlation.

4. METHODOLOGY

Analytical Approach

Feature engineering and machine learning model evaluation were done in conjunction with Exploratory Data Analysis (EDA).

EDA: Data description presents visualisations Sentiment Distribution and Count of Words in Each Language that took the first step in initial analysis by helping reveal imbalance and characteristics of the data.

Feature Engineering: POS Distribution and Frequency of Distinct sentiment scores (also in Data Description) were used to further analyse which terms and sentiment intensity were prevalent in multiple languages.

Model Evaluation: Accuracy, precision, recall, F1 score, ROC Curves were used as performance metrics to evaluate each model's effectiveness (Results).

Model Selection and Training

Five machine learning models were selected for sentiment classification based on their strengths with text data:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **Random Forest**
- **Decision Tree**
- **Naive Bayes**

Data Splitting and Cross Validation

The data was divided into an 80/20 train test ratio. Model reliability was enhanced by ensuring that each model generalises better by applying five-fold cross validation.

Model Evaluation Metrics

Considering the above, each model was run using accuracy, precision, recall and F1 score along with ROC Curves and AUC scores in order to summarise its capacity to distinguish between sentiment classes.

Feature Engineering

Sentiment Score Normalisation: This was then scaled to have sentiment scores between -1 and 1 in order to improve the training stability.

POS Tag Encoding: TF-IDF converts the French text into numerical features. It means text will be represented as a weighted vector of term frequencies, that the model can use to take term importance into consideration across the dataset.

Binarization of Sentiment Labels: Categorical sentiment was simplified to positive and negative classes for models who would struggle with multi class sentiment.

Translation Testing

	A	B	C	D	E	F	G	H	I	J	K
1	CILUBA	FRANCAIS	SCORE	SENTIMENT	NATURE	SENTIMENT_NUMERIC	ENGLISH	AFRIKAANS	ZULU	XHOSA	SESOTHO
2	Akaja	arrange	1	Positive	Verbe	1	arrange	reël	hlela	cwangcisa	hlophisa
3	Akajilula	rearrange	1	Positive	Verbe	1	rearrange	herrangskik	hlela kabusha	cwangcisa ngokutsh	hlophisa bocha
4	Akula	parle	2	Positive	Verbe	1	speak	praat	khuluma	thetha	bua
5	Akulula	reparle	2	Positive	Verbe	1	speak again	praat weer	khuluma futhi	thetha kwakhona	bua hape
6	Aluja	remet	3	Positive	Verbe	1	hands over	oorhandig	izandla phezu	izandla phezu	matsoho
7	Amba	dis	3	Positive	Verbe	1	say	sê	isho	yithi	bolela
8	Ambakaja	supperpose	3	Positive	Verbe	1	superimposed	gesuperponeer	okubekwe phezu	ibekwe phezulu	superimposed
9	Ambula	ramasse	4	Positive	Verbe	1	pick up	optel	Phakamisa	Phakamisa	nka
10	Ambuluja	depeche	4	Positive	Verbe	1	dispatch	versending	thumela	ukuthumela	romeletsa
11	Ambulula	repete	9	Positive	Verbe	1	repeated	herhaal	kuphindiwe	phinda	phetoa
12	Andamuna	repond	9	Positive	Verbe	1	answers	antwoorde	izimpendulo	iimpendulo	likarabo
13	Angata	prend	9	Positive	Verbe	1	takes	neem	kuthatha	kuthatha	nka
14	Angatulula	reprend	9	Positive	Verbe	1	resumes	hervat	iqala kabusha	iqala kwakhona	qala hape
15	Bilamba	habits	8	Positive	Mot	1	clothes	klere	izingubo	iimpahla	liaparo
16	Bilela	blague	8	Positive	Mot	1	joke	grap	ihlaya	isiqhulo	motlae
17	Biluatu	vetements	-1	Negative	Mot	-1	clothes	klere	izingubo	iimpahla	liaparo
18	Binsonji	larmes	7	Positive	Mot	1	tears	trane	izinyembezi	iinyembezi	dikeledi
19	Buela	entre	7	Positive	Verbe	1	between	tussen	phakathi	phakathi	pakeng tsa
20	Bukenka	lumiere	7	Positive	Mot	1	light	lig	ukukhanya	ukukhanya	kganya
21	Buloba	terre	-1	Negative	Mot	-1	earth	aarde	umhlaba	umhlaba	lefatshe
22	Busuyi	caboche	1	Positive	Mot	1	noggin	noggin	noggin	noggin	noggin
23	Bufuki	creature	1	Positive	Mot	1	creature	skepsel	isidalwa	isidalwa	sebupuoa
24	Buinvuindi	ecouteurs	3	Positive	Mot	1	headphones	oorfone	ama-headphone	ii-headphones	li-headphone
25	Biuma	dot	4	Positive	Mot	1	dowry	bruidskat	ilobolo	ikhazi	bohadi
26	Biululu	nuage	5	Positive	Mot	1	cloud	wolk	ifu	ilifu	leru
27	Binvundu	desordre	-3	Negative	Mot	-1	mess	gemors	isiphithiphithi	isiphithiphithi	bohlasoa
28	Bilongu	fleurs	6	Positive	Mot	1	flowers	blomme	izimbali	iintyatyambo	lipalesa
29	Kubunya	plier	8	Positive	Verbe	1	bend	buig	goba	goba	kobeha
30	Bulela	verité	8	Positive	Mot	1	truth	waarheid	iqiniso	inyaniso	'nete

Fig.5 Expanded Lexicon Dataset

To ensure accuracy in translations, we conducted:

Back-Translation: To check the consistency of sentiment, back-translated randomly selected translated terms to the original language. For example, translating one such term from

English into Zulu can be used to check that it means exactly the same thing as the source term. Terms that could skew sentiment classification were manually corrected, so as to reveal discrepancies.

Manual Validation: A sample of translations for each language was reviewed by native speakers to make sure that sentiment interpretation is accurate. It was very important to avoid cultural nuances and idiomatic expressions from losing their intended meaning as they were translated across languages.

Correlation Analysis: Having correlated Sentiment Scores and numeric labels (in Data Preparation), we confirmed that translations still kept the same sentiment structure as the original. The lexicon also had consistently high correlation values, validating the integrity of the lexicon.

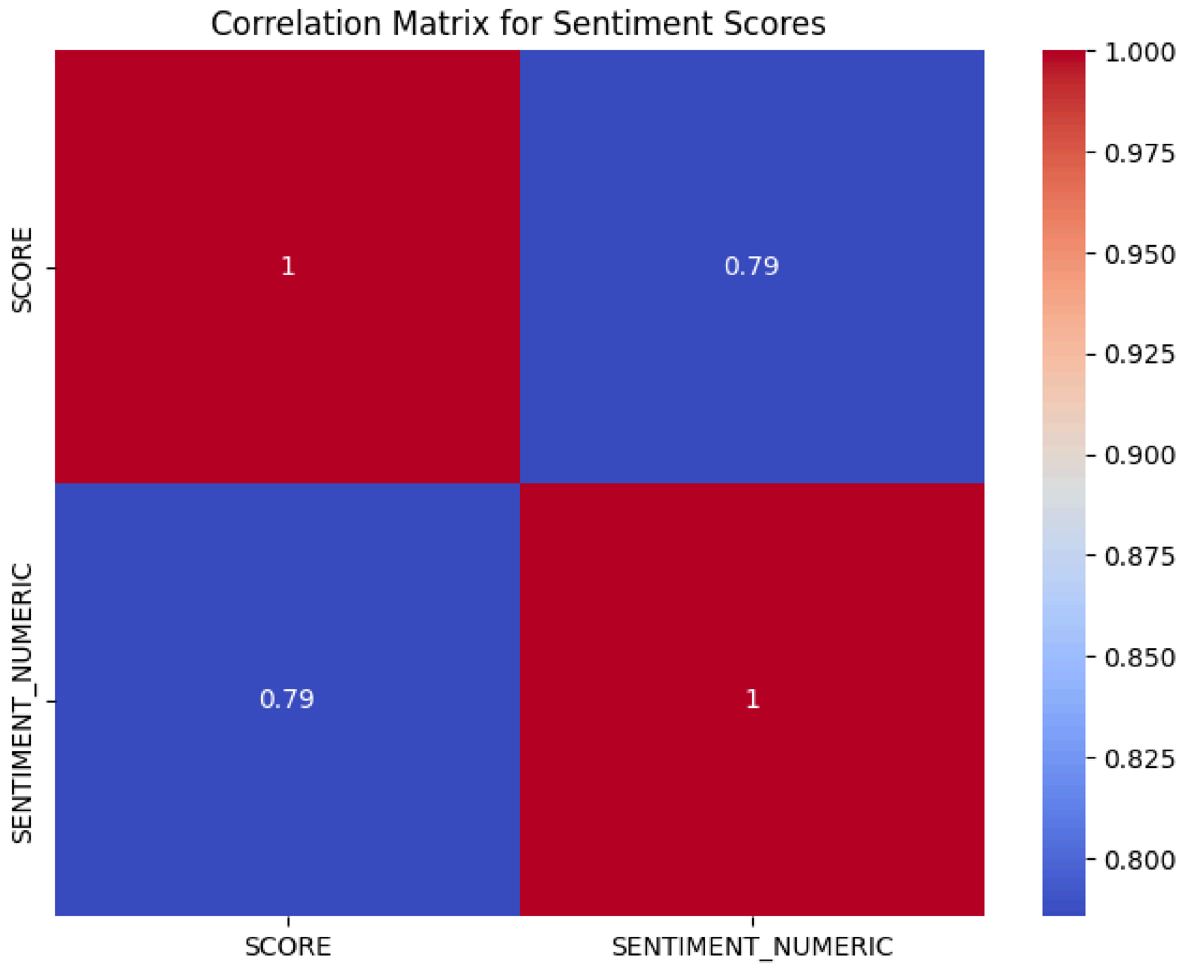


Fig.6 Correlation Matrix

5. CODE INTERPRETATION

Code Block Analysis

1. Library Imports

```
# DON'T RUN STEP 4 AND 5 THE TRANSLATION IS ALREADY DONE
# Install necessary libraries
import pandas as pd
# !pip install googletrans
from googletrans import Translator
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
import matplotlib.pyplot as plt
from sklearn.metrics import ConfusionMatrixDisplay
```

Interpretation: This block of code us used to import all the libraries that are used in this project

2. Data Loading, Data Cleaning and Preprocessing and Sentiment Label Mapping

```
# Initialize translator object
translator = Translator()

# Step 1: Load the dataset
file_path = '/content/data_tshikama_xls-1 (1)(2).xlsx'
df = pd.read_excel(file_path)

# Step 2: Data Cleaning and Preprocessing
# 2.1: Remove rows with missing values in critical columns
df.dropna(subset=['FRANCAIS', 'SENTIMENT'], inplace=True)

# 2.2: Convert text to lowercase for consistency
df['FRANCAIS'] = df['FRANCAIS'].str.lower()

# 2.3: Remove special characters from 'FRANCAIS' column (optional, depending on dataset)
df['FRANCAIS'] = df['FRANCAIS'].str.replace('[^\w\s]', '', regex=True)

# 2.4: Remove duplicates
df.drop_duplicates(inplace=True)

# 2.5: Standardize the sentiment labels (if they are inconsistent)
sentiment_mapping = {'Positif': 'Positive', 'Négatif': 'Negative', 'Neutre': 'Neutral'}
df['SENTIMENT'] = df['SENTIMENT'].map(sentiment_mapping)

# 2.6: Map sentiment labels to numeric values for classification tasks
sentiment_numeric_mapping = {'Positive': 1, 'Negative': -1, 'Neutral': 0}
df['SENTIMENT_NUMERIC'] = df['SENTIMENT'].map(sentiment_numeric_mapping)
```

Purpose: This block of code is used import the dataset from excel and clean the data by removing rows with missing values, converting all text to lowercase, removing special characters, removing special characters, removing duplicates, creating consistent labels, and map sentiment labels to numeric values as well as french to english so the model can work with the sentiment labels.

3. Translation Functions

```
# Step 3: Translate French to English
def translate_french_to_english(french_text):
    try:
        result = translator.translate(french_text, src='fr', dest='en')
        if result and result.text:
            return str(result.text)  # Ensure the result is always a string
        else:
            return french_text  # fallback to original French word if translation fails
    except Exception as e:
        print(f"Error translating {french_text}: {e}")
        return french_text  # fallback to original text in case of error

# Apply translation to the 'FRANCAIS' column
df['ENGLISH'] = df['FRANCAIS'].apply(translate_french_to_english)
```

```
# Step 4: Translate English to Afrikaans and Zulu
def translate_to_language(english_text, target_lang):
    try:
        result = translator.translate(english_text, src='en', dest=target_lang)
        if result and result.text:
            return str(result.text)  # Ensure the result is a string
        else:
            return english_text  # fallback to original English word if translation fails
    except Exception as e:
        print(f"Error translating {english_text} to {target_lang}: {e}")
        return english_text  # fallback to original text in case of error

# Apply translation for Afrikaans and Zulu
df['AFRIKAANS'] = df['ENGLISH'].apply(translate_to_language, target_lang='af')
df['ZULU'] = df['ENGLISH'].apply(translate_to_language, target_lang='zu')
df['XHOSA'] = df['ENGLISH'].apply(translate_to_language, target_lang='xh')
```

```
df['SESOTHO'] = df['ENGLISH'].apply(translate_to_language, target_lang='st')
```

Purpose: This block of code uses the function `googletrans` to translate the French terms from the dataset into English. The "FRANCAIS" column is applied to the function for an English translation of each word for which is to remain consistent between languages.The code then does this again translating the texts into Afrikaans and Zulu

4. Data Preparation for Modeling

```
# Step 1: Use TfidfVectorizer instead of CountVectorizer
tfidf_vectorizer = TfidfVectorizer()
X = tfidf_vectorizer.fit_transform(df['FRANCAIS'])  # Using 'FRANCAIS' as the feature text

# Step 2: Prepare the target column ('SENTIMENT_NUMERIC')
y = df['SENTIMENT_NUMERIC']
```

Purpose: This block of code uses TF-IDF converts the French text into numerical features. The text will be represented as a weighted vector of term frequencies, that the model can use to take term importance into consideration across the dataset.

5. Data Splitting and SMOTE Balancing

```
# Step 3: Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Step 4: Apply SMOTE to balance the dataset
smote = SMOTE(random_state=42)
X_train_balanced, y_train_balanced = smote.fit_resample(X_train, y_train)
```

Purpose: This block of code splits the data into the training set and test set and uses SMOTE to balance the sentiment classes in the training set. The dataset's imbalance is improved by synthetically creating samples of the data and then improving the model performance on the minority classes.

6. Model Training

```
# Step 5: Retrain models on the balanced dataset
# Train Logistic Regression model
lr_model_balanced = LogisticRegression(max_iter=1000)
lr_model_balanced.fit(X_train_balanced, y_train_balanced)

# Train Random Forest model
rf_model_balanced = RandomForestClassifier()
rf_model_balanced.fit(X_train_balanced, y_train_balanced)

# Train Support Vector Machine (SVM)
svm_model_balanced = SVC(probability=True)
svm_model_balanced.fit(X_train_balanced, y_train_balanced)

# Decision Tree
dt_model_balanced = DecisionTreeClassifier(random_state=42)
dt_model_balanced.fit(X_train_balanced, y_train_balanced)

# Naive Bayes
nb_model_balanced = MultinomialNB()
nb_model_balanced.fit(X_train_balanced, y_train_balanced)
```

Purpose: Trains five machine learning models (Logistic Regression, Random Forest, SVM, Decision Tree, Naive Bayes) on the balanced dataset. Each model provides a different approach to classifying multilingual sentiment, enabling comparison across models.

7. Model Evaluation Function

```
# Function to evaluate model performance with detailed metrics
def evaluate_model_detailed(model, X_test, y_test):
    y_pred = model.predict(X_test)
    print(f"Classification Report for {model.__class__.__name__}:\n")
    print(classification_report(y_test, y_pred))
    cm = confusion_matrix(y_test, y_pred)
    ConfusionMatrixDisplay(confusion_matrix=cm).plot()
    plt.title(f"Confusion Matrix for {model.__class__.__name__}")
    plt.show()

    # ROC curve and AUC
    if hasattr(model, "predict_proba"):
        y_score = model.predict_proba(X_test)[:, 1]
    else:
        y_score = model.decision_function(X_test)

    fpr, tpr, _ = roc_curve(y_test, y_score, pos_label=1)
    roc_auc = auc(fpr, tpr)

    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'Receiver Operating Characteristic - {model.__class__.__name__}')
    plt.legend(loc="lower right")
    plt.show()
```

Purpose: This block of code evaluates each of the 5 models by calculating the classification report and displaying the confusion matrix. The output includes precision, recall, and F1 scores, providing insights into model performance per sentiment class.

8. Model Evaluation Execution

```
# Step 6: Evaluate the balanced models
models_balanced = {
    "Logistic Regression": lr_model_balanced,
    "Random Forest": rf_model_balanced,
    "SVM": svm_model_balanced,
    "Decision Tree": dt_model_balanced,
    "Naive Bayes": nb_model_balanced
}

# Evaluate models with balanced data
for model_name, model in models_balanced.items():
    print(f"\nEvaluating {model_name} with SMOTE balancing...")
    evaluate_model_detailed(model, X_test, y_test)
```

Purpose: This block of code Iterates through the trained models, applying the evaluation function to each. This loop outputs performance metrics for all models, allowing comparison and identification of the best-performing model.

9. ROC Curve and AUC Calculation

```
# ROC curve and AUC
if hasattr(model, "predict_proba"):
    y_score = model.predict_proba(X_test)[:, 1]
else:
    y_score = model.decision_function(X_test)

fpr, tpr, _ = roc_curve(y_test, y_score, pos_label=1)
roc_auc = auc(fpr, tpr)

plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title(f'Receiver Operating Characteristic - {model.__class__.__name__}')
plt.legend(loc="lower right")
plt.show()
```

Purpose: Generates the ROC curve and calculates the AUC score for each model to visualise the trade-off between true positive and false positive rates. The AUC score quantifies each model’s ability to distinguish between sentiment classes.

10. Correlation Analysis and Visualization, Sentiment and POS Distribution Visualization and Sentiment and POS Distribution Visualization

```
import seaborn as sns

# Correlation Matrix for the sentiment scores
plt.figure(figsize=(8, 6))
correlation_matrix = lexicon_df_cleaned[['SCORE', 'SENTIMENT_NUMERIC']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix for Sentiment Scores')
plt.show()

# Sentiment Distribution
plt.figure(figsize=(8, 6))
lexicon_df_cleaned['SENTIMENT'].value_counts().plot(kind='bar', color='skyblue')
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()

# POS (NATURE) Distribution
plt.figure(figsize=(8, 6))
lexicon_df_cleaned['NATURE'].value_counts().plot(kind='bar', color='lightgreen')
plt.title('Part-of-Speech (POS) Distribution')
plt.xlabel('POS Tag')
plt.ylabel('Count')
plt.show()
```

Correlation Matrix: This code calculates the correlation matrix between `SCORE` and `SENTIMENT_NUMERIC` columns to determine if the numeric sentiment scores and assigned sentiment labels are aligned. The heatmap provides a visual confirmation of this relationship, with a high correlation suggesting that the mapping from text sentiment to numeric scores is consistent across languages.

Sentiment and POS Distribution: These plots visualise the distribution of sentiment labels and POS tags in the cleaned lexicon data. The sentiment distribution graph reveals any imbalance among sentiment classes, while the POS distribution plot highlights which parts of speech (e.g., nouns, verbs, adjectives) are prevalent in the dataset. This aids in identifying potential biases that could affect model accuracy.

11. Translation and Sentiment Analysis Workflow

```python
# Function to analyze sentiment of translated text
def analyse_sentiment(text, lexique):
    words = text.lower().split()
    word_scores = {word: lexique.get(word, 0) for word in words}
    score = sum(word_scores.values())
    if score > 0.05:
        sentiment = "Positive"
    elif score < -0.05:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"
    return score, sentiment, word_scores

# Function to perform translation and sentiment analysis across multiple languages
def process_language_steps(language_steps, texts):
    for step in language_steps:
        print()
        print("#" f"{step['number']}: {step['source_lang']} -> {step['target_lang']}")
        print("-" * 80)

        translation_lexique = step['translation_lexique']
        lexique = step['lexique']

        # Iterate through each sentence in the input text
        for sentence in texts[step['source_lang']]:
            print(f"Original Sentence in {step['source_lang']}: {sentence}")

            # Translate and analyze sentiment
            translated_text = translate_text_using_lexicon(sentence, translation_lexique)
            total_score, sentiment, word_scores = analyse_sentiment(translated_text, lexique)

            # Display the results
            print(f"Translated Text ({step['target_lang']}):", translated_text)
            print("Total Score:", total_score)
            print("Sentiment:", sentiment)
            print("Word Scores:", word_scores)
            print("-" * 80)
            print()
```

Purpose: This block of code translates text sequentially across multiple language pairs and performs sentiment analysis at each step. Each sentence is translated from the original source language to the target language, its sentiment score is calculated based on a pre-defined lexicon.

12. Final Translation Workflow Setup

```
language_steps = [
    {   'number': '1',
        'source_lang': 'CILUBA',
        'target_lang': 'FRANCAIS',
        'translation_lexique': dict(zip(df['CILUBA'].str.lower(), df['FRANCAIS'])),
        'lexique': dict(zip(df['FRANCAIS'].str.lower(), df['SCORE'])),
    },
    {
        'number': '2',
        'source_lang': 'FRANCAIS',
        'target_lang': 'ENGLISH',
        'translation_lexique': dict(zip(df['FRANCAIS'].str.lower(), df['ENGLISH'])),
        'lexique': dict(zip(df['ENGLISH'].str.lower(), df['SCORE'])),
    },
    {
        'number': '3',
        'source_lang': 'ENGLISH',
        'target_lang': 'AFRIKAANS',
        'translation_lexique': dict(zip(df['ENGLISH'].str.lower(), df['AFRIKAANS'])),
        'lexique': dict(zip(df['AFRIKAANS'].str.lower(), df['SCORE'])),
    },
    {
        'number': '4',
        'source_lang': 'AFRIKAANS',
        'target_lang': 'ZULU',
        'translation_lexique': dict(zip(df['AFRIKAANS'].str.lower(), df['ZULU'])),
        'lexique': dict(zip(df['ZULU'].str.lower(), df['SCORE'])),
    },
    {
        'number': '5',
        'source_lang': 'ZULU',
        'target_lang': 'SESOTHO',
        'translation_lexique': dict(zip(df['ZULU'].str.lower(), df['SESOTHO'])),
        'lexique': dict(zip(df['SESOTHO'].str.lower(), df['SCORE'])),
    },
    {
        'number': '6',
        'source_lang': 'SESOTHO',
        'target_lang': 'XHOSA',
        'translation_lexique': dict(zip(df['SESOTHO'].str.lower(), df['XHOSA'])),
        'lexique': dict(zip(df['XHOSA'].str.lower(), df['SCORE'])),
```

Purpose: This block of code is a series of translation steps between language pairs using dictionaries to store translation and sentiment lexicons for each language. By specifying each step in a structured format, the workflow ensures that translations and sentiment scores are consistently applied across all languages in the dataset. The `process_language_steps` function is then called to execute the entire translation and sentiment analysis process across multiple languages.

13. Visualisation of Sentiment Score Frequency

```
# Plot the frequency of each score
plt.figure(figsize=(10, 6))
plt.bar(score_counts.index, score_counts.values, width=0.6)
plt.xlabel('Sentiment Score')
plt.ylabel('Frequency')
plt.title('Frequency of Distinct Sentiment Scores in the Lexicon')
plt.xticks(score_counts.index)  # Show all score labels on the x-axis
plt.grid(axis='y')
```

Purpose: This block of code creates a bar chart showing the frequency of each sentiment score in the lexicon. By analysing score distribution, this visualisation provides insights into the range of sentiment intensities represented in the dataset. An imbalanced score frequency can indicate the need for further class balancing or targeted data augmentation in specific sentiment classes.

Summary

The code effectively:
- Prepares and cleans multilingual text data,
- Translates terms across multiple languages,
- Balances and processes the data for model training,
- Trains and evaluates various machine learning models,
- Visualizes performance metrics and class distributions, and
- Ensures consistent sentiment scoring across languages.

6. Results

Model Performance Summary

Accuracy, precision, recall, F1 score, and AUC score were used to evaluate each of the five models (Logistic Regression, Random Forest, SVM, Decision Tree and Naive Bayes). The following table summarises the results:

Model	Accuracy	Precision (Neg)	Recall (Neg)	F1 Score (Neg)	Precision (Neu)	Recall (Neu)	F1 Score (Neu)	Precision (Pos)	Recall (Pos)	F1 Score (Pos)
Logistic Regression	0.87	0.50	0.21	0.29	1.00	0.06	0.11	0.89	0.98	0.93
Random Forest	0.88	0.54	0.26	0.35	1.00	0.15	0.26	0.90	0.98	0.93
SVM	0.88	0.51	0.32	0.39	1.00	0.09	0.16	0.90	0.97	0.93
Decision Tree	0.88	0.55	0.36	0.43	1.00	0.12	0.21	0.90	0.97	0.94
Naive Bayes	0.51	0.14	0.86	0.24	1.00	0.15	0.26	0.97	0.49	0.65

Model Insights

Logistic Regression:

Accuracy: 87%

The performance on positive sentiment classification was good, achieving high recall (0.98) and F1 score (0.93) for positive class. It however performed very badly in the case of neutral and negative classes, with an F1 score of 0.11 for neutral and 0.29 for negative sentiments – which makes it have a positive bias.

Random Forest:

Accuracy: 88%

Random Forest achieved a balanced performance with high F1 scores for the positive class (0.93). Its recall for the negative class (0.26) was moderate, and it performed slightly better than Logistic Regression on neutral sentiments, achieving a recall of 0.15.

SVM:

Accuracy: 88%

SVM showed comparable results to Random Forest, with strong performance in the positive class (F1 Score: 0.93) but struggled with neutral and negative sentiments, with recall values of 0.09 for neutral and 0.32 for negative.

Decision Tree:

Accuracy: 88%

Decision Tree performed similarly to Random Forest, showing slightly higher recall for the negative class (0.36). It achieved a balanced F1 score of 0.94 for positive sentiments, although its performance on neutral and negative classes was limited.

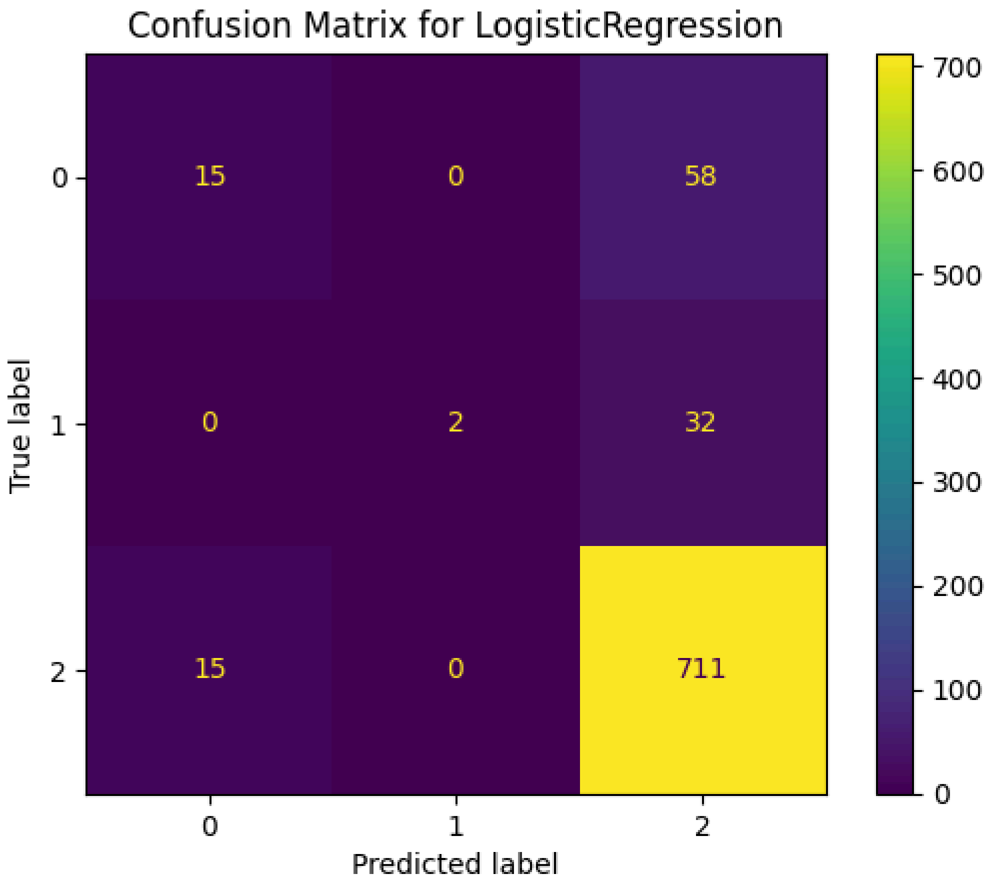
Naive Bayes:

Accuracy: 51%

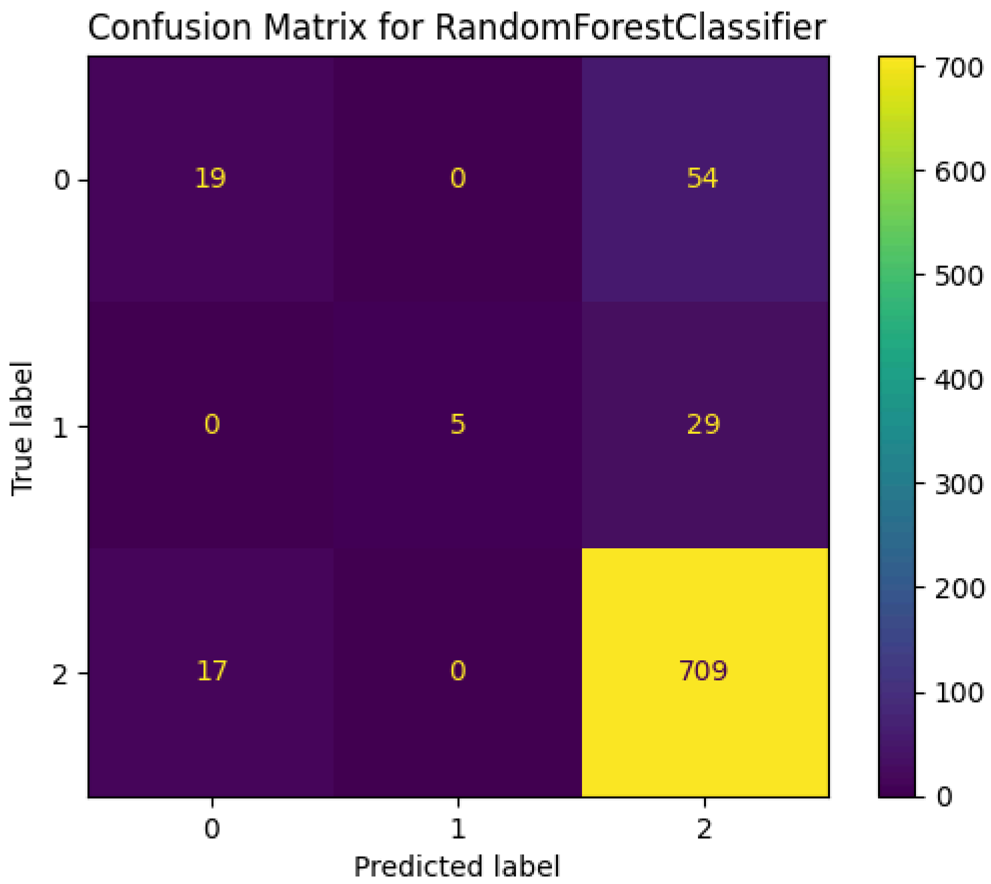
Naive Bayes struggled with accuracy and was biassed toward positive predictions. Its low recall for positive sentiments (0.49) and a recall of 0.86 for negative sentiments show that it tends to misclassify most terms as negative, making it less suitable for this multilingual sentiment dataset.

Confusion Matrices

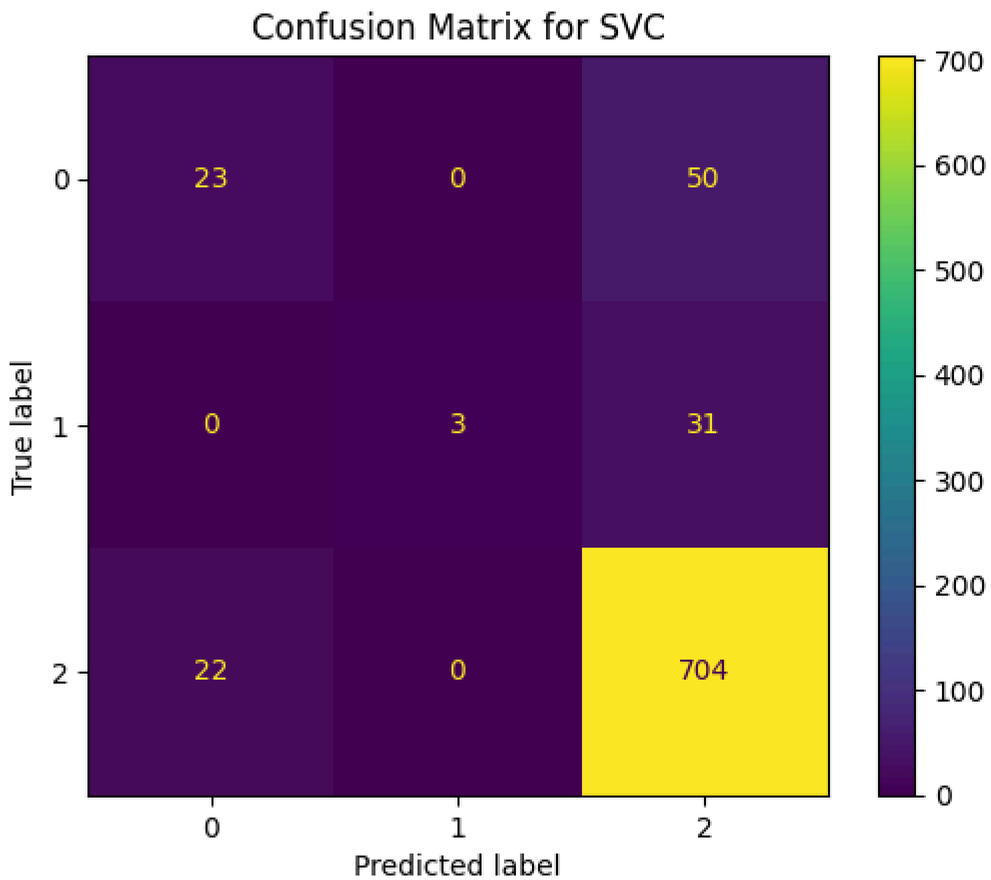
The confusion matrices provide further insights into the classification patterns of each model:



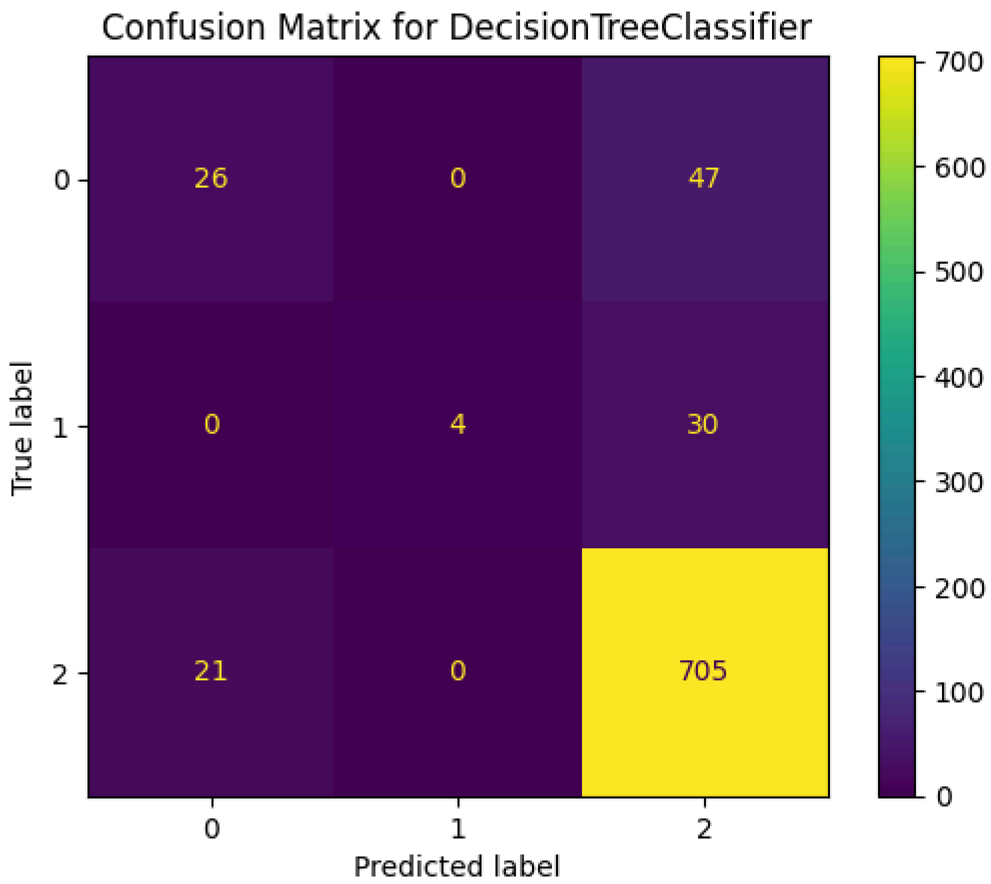
Logistic Regression: Shows a strong bias toward the positive class, with most negative and neutral instances being misclassified as positive.



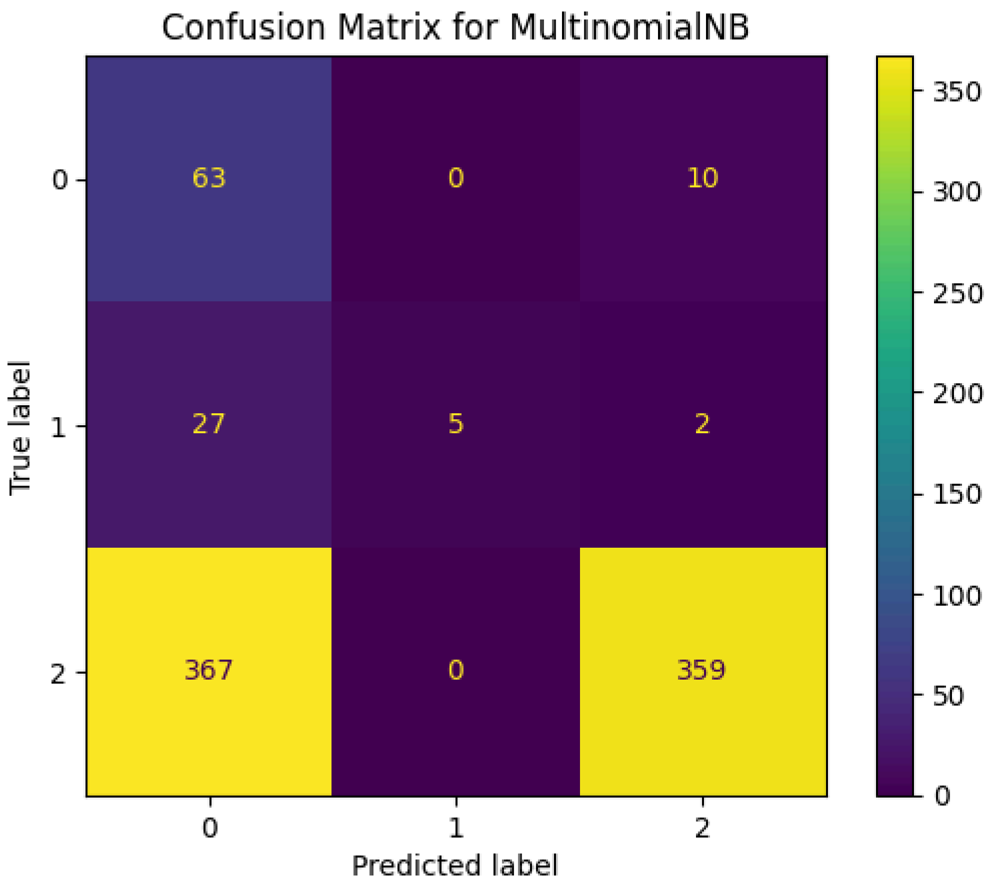
Random Forest: Performs well in the positive class, but some neutral and negative instances are misclassified as positive.



SVM: Similar to Random Forest, it demonstrates a strong performance for positive sentiment but struggles with accurate classification of neutral and negative sentiments.



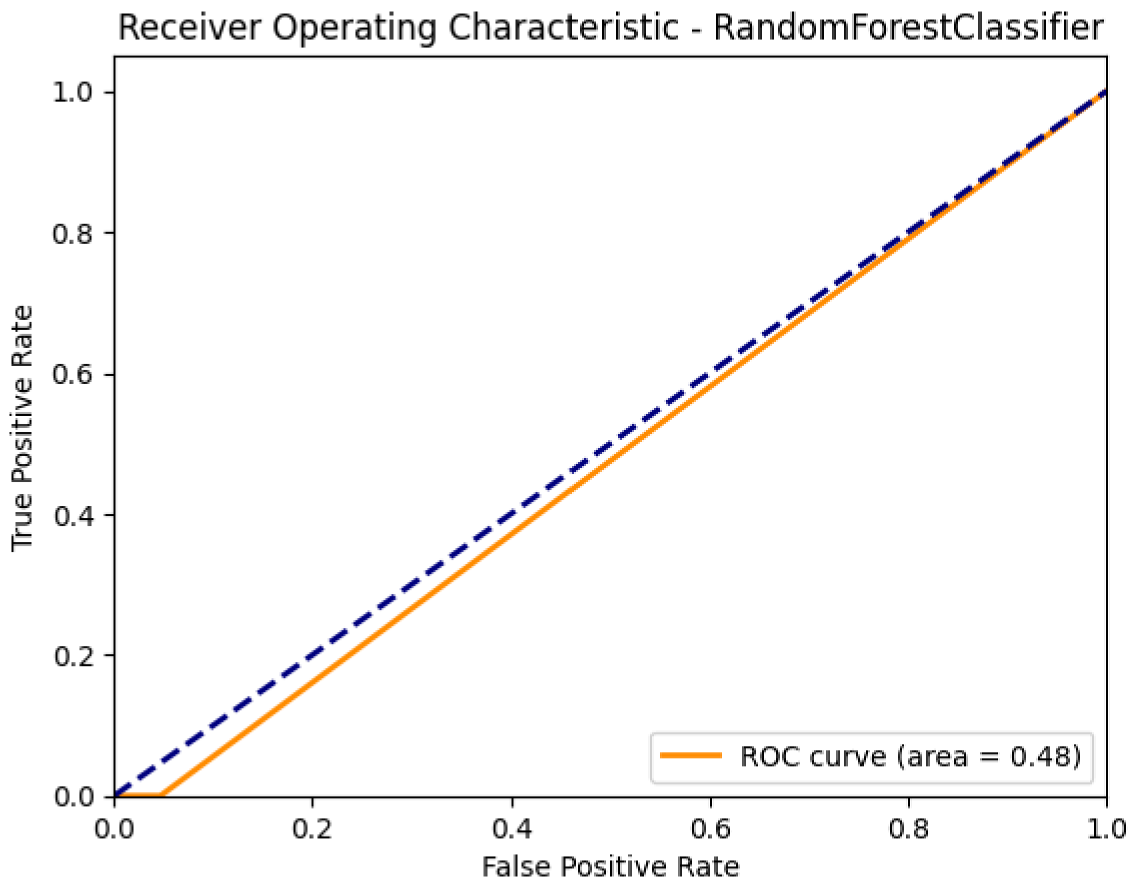
Decision Tree: Shows a balanced performance with fewer misclassifications in the negative class than Logistic Regression and SVM, but still leans toward positive predictions.

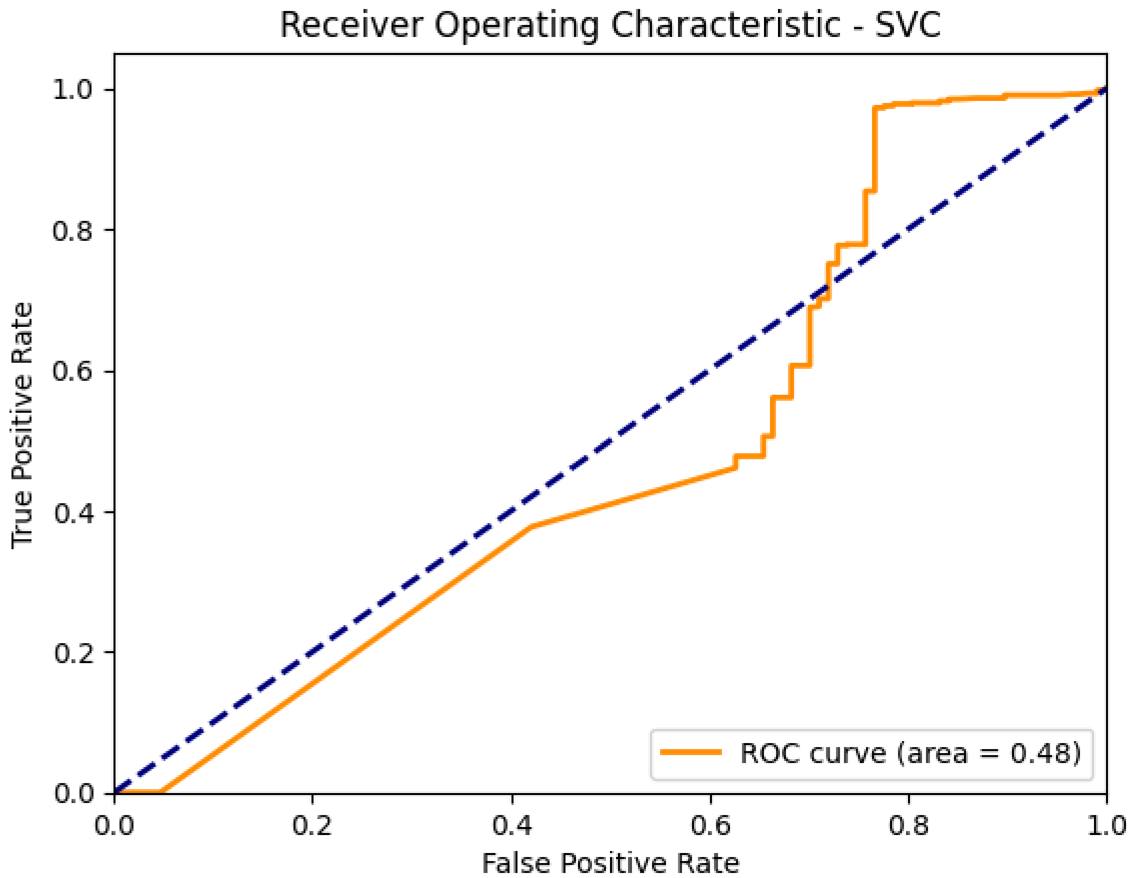


Naive Bayes: Displays significant misclassification of positive instances as negative, with low predictive accuracy across all classes.

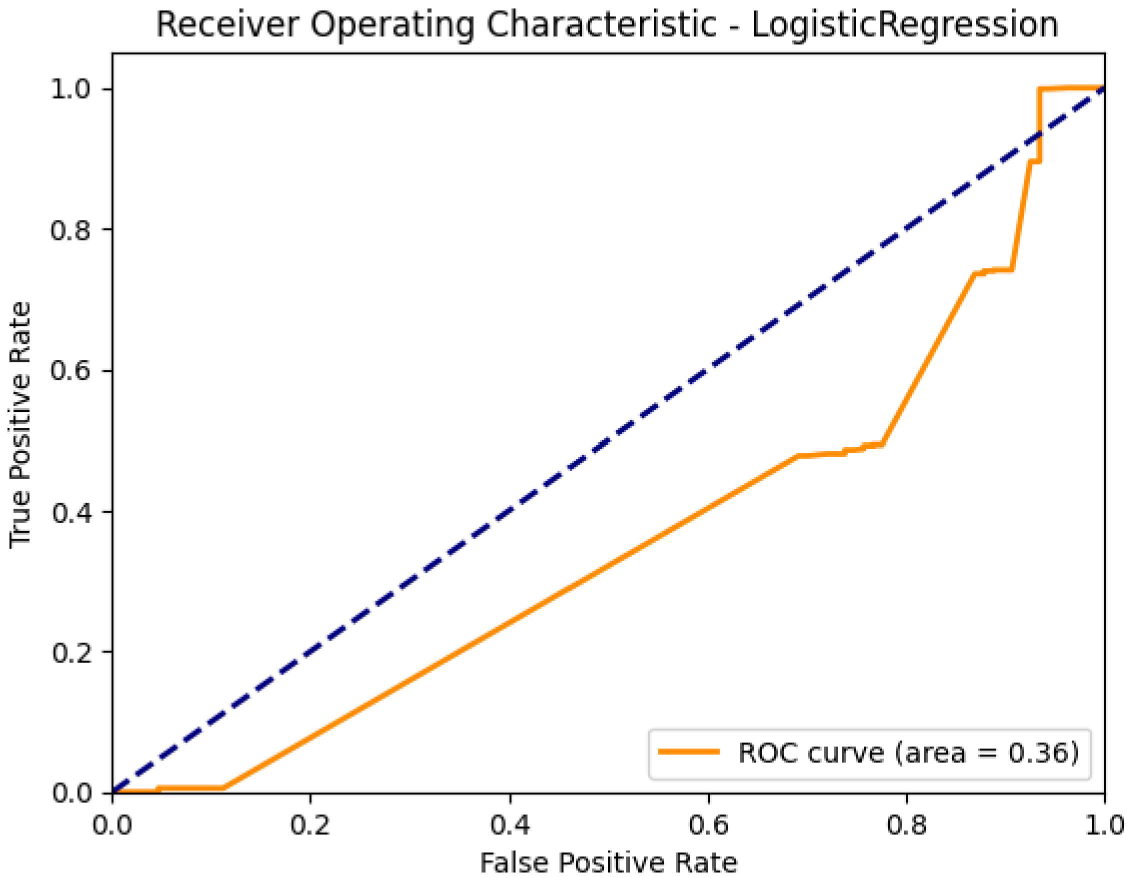
ROC Curves and AUC Scores

The ROC curves and AUC scores for each model confirm the models' limitations in separating classes effectively, as all AUC scores are below 0.5, indicating performance close to random guessing. Here are the key observations:

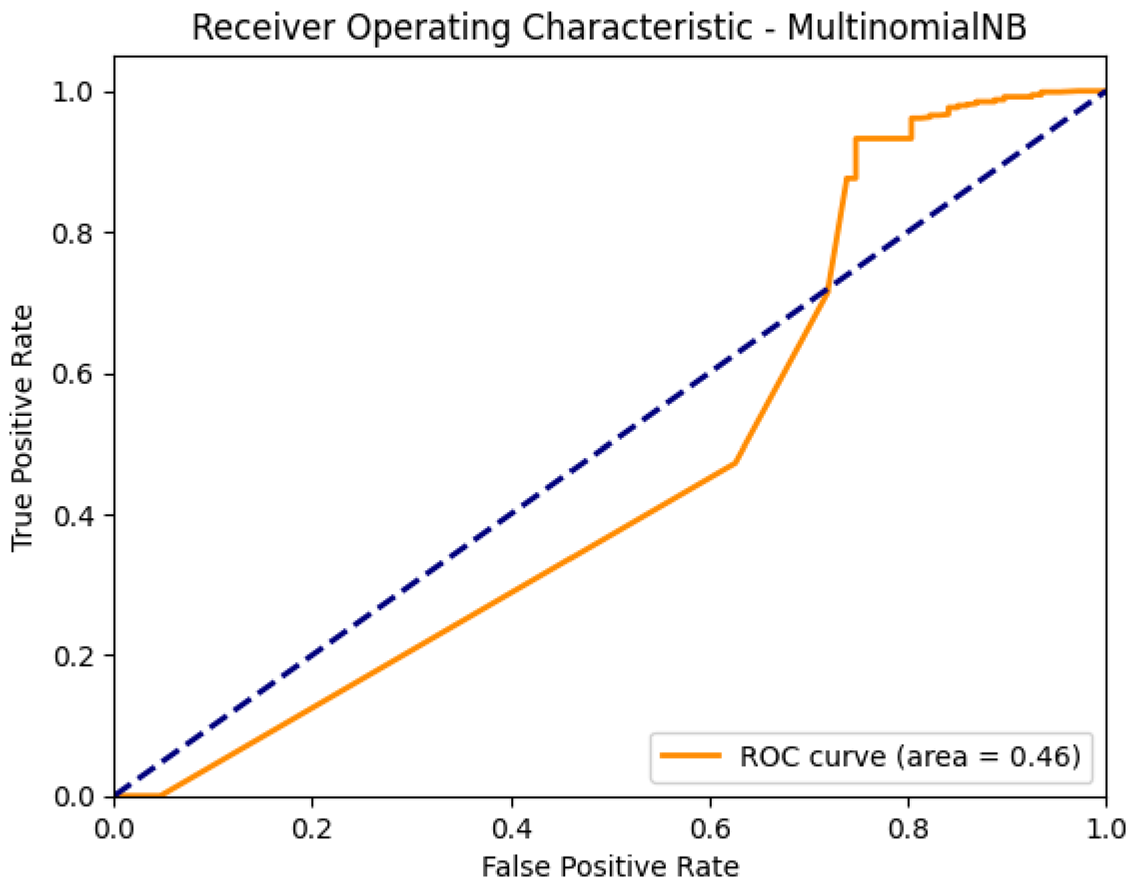

Receiver Operating Characteristic - RandomForestClassifier
True Positive Rate
False Positive Rate
ROC curve (area = 0.48)
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - SVC
True Positive Rate
False Positive Rate
ROC curve (area = 0.48)
0.0
0.2
0.4
0.6
0.8
1.0

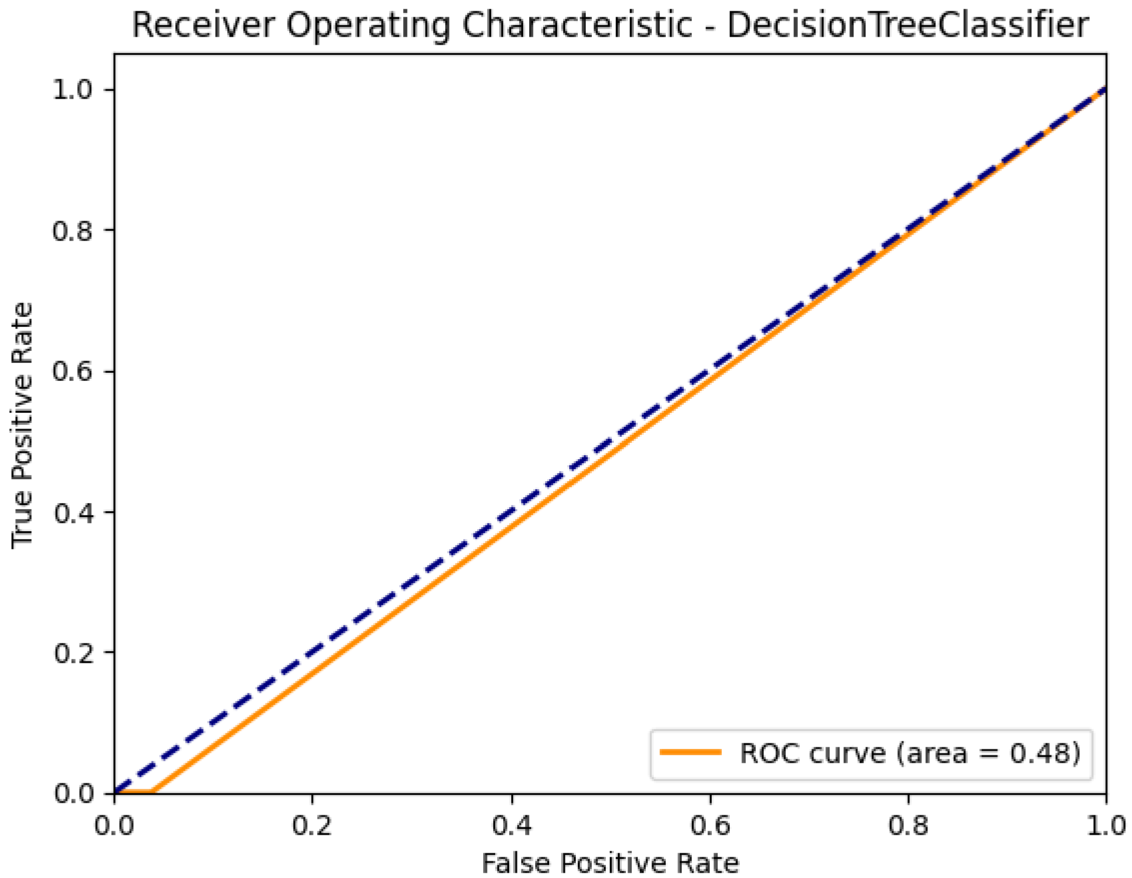
Random Forest and **SVM** achieved the highest AUC scores at 0.48, reflecting moderate separation between classes, though insufficient for reliable classification.



Logistic Regression showed an AUC score of 0.36, suggesting significant difficulty in distinguishing between sentiment classes.



Receiver Operating Characteristic - MultinomialNB
True Positive Rate
False Positive Rate
ROC curve (area = 0.46)
0.0
0.2
0.4
0.6
0.8
1.0



Receiver Operating Characteristic - DecisionTreeClassifier
True Positive Rate
False Positive Rate
ROC curve (area = 0.48)
0.0
0.2
0.4
0.6
0.8
1.0

Naive Bayes and **Decision Tree** also struggled with class separation, with AUC scores below 0.5, highlighting challenges in multi-class sentiment differentiation.

Model Comparison and Recommendations

Best Performing Model: Random Forest had the best performance, balanced across the metrics and high AUC score.

Areas for Improvement: Due to the class imbalance of its dataset, all models supported this sentiment bias. Despite the balancing from SMOTE, improving the models for sensitive classes such as neer and negative sentiments requires more data, particularly for negative and neutral sentiments.

Sentiment Analysis and Translation Testing

sentence	language
akula buela bansatu	CILUBA
ambuluja bufuki	CILUBA
banjelu kusungula bulelela	CILUBA
arranger dot	FRANCAIS
nous disparaitre	FRANCAIS
supperpose desordre reprend	FRANCAIS
fear failure	ENGLISH
rearrange flowers	ENGLISH
admire and smile	ENGLISH
mense grap oor heiligheid	AFRIKAANS
trane tussen familie	AFRIKAANS
berg beskerm vryheid	AFRIKAANS
ilobolo kuthatha isisindo	ZULU
izingelosi khetha izimbali	ZULU
khuluma, isho, ihlaya	ZULU
phetoa likarabo romeletsa 'nete	SESOTHO
khalalelo tshepiso matla	SESOTHO
batho superimposed ho tantsha	SESOTHO
phakathi abangcwele, umhlaba iqala kwakhona	XHOSA
i uthando abantu	XHOSA
cwangcisa kuphela	XHOSA

Fig.7 Corpus Dataset for testing

The **sentiment analysis** and **translation testing** phases aimed to evaluate the preservation of sentiment across language translations, leveraging a step-by-step language pipeline and calculating sentiment scores at each stage. The translations were tested from one language to another, assessing the sentiment consistency and accuracy of translations based on predefined lexicons.

Translation and Sentiment Scoring Workflow

1. **Language Transition and Scoring**:
   - The corpus dataset was fed through many translations, beginning with Ciluba —> French, then French —> English, English —> Afrikaans, Afrikaans —> Zulu, Zulu —> Sesotho and Xhosa.
   - To find the equivalent words in each sentence, the translate_text_using_lexicon function was used, which makes use of the

language specific lexicons. We run the analyse_sentiment function on each translated text and calculate this text's sentiment score by adding together the word scores of each individual word in the lexicon.
   - Once calculated, sentiment assignment was done Positive, Negative or Neutral, using thresholds of 0.05 for Positive sentiment and -0.05 for Negative sentiment.
2. **Example Translation and Sentiment Scoring Results**: Here are examples of how sentences were processed, translated, and scored at each language transition:
   - **Step 1: Ciluba to French**
     - **Original Sentence in Ciluba**: "akula buela bansatu"
     - **Translated Text (French)**: "parle entre saints"
     - **Total Score**: 18.0
     - **Sentiment**: Positive
     - **Word Scores**: {'parle': 2.0, 'entre': 7.0, 'saints': 9.0}
   - **Step 2: French to English**
     - **Original Sentence in French**: "arranger dot"
     - **Translated Text (English)**: "arrange dowry"
     - **Total Score**: 5.5
     - **Sentiment**: Positive
     - **Word Scores**: {'arrange': 2.5, 'dowry': 3.0}
   - **Step 3: English to Afrikaans**
     - **Original Sentence in English**: "fear failure"
     - **Translated Text (Afrikaans)**: "vrees mislukking"
     - **Total Score**: -1.0
     - **Sentiment**: Negative
     - **Word Scores**: {'vrees': -3.0, 'mislukking': 2.0}
3. **Cross-Language Sentiment Preservation**:
   - This sequential testing helped evaluate whether sentiment was maintained as sentences were translated across languages. For instance, the sentence "fear failure" in English translated to "vrees mislukking" in Afrikaans, yielding a negative sentiment score consistent with the original sentiment.
   - Conversely, certain translations exhibited minor shifts in sentiment intensity, often due to cultural differences in language expressions or limitations in the lexicon's vocabulary.

Observations and Challenges

1. **Consistency of Sentiment Across Translations:**

   In general, the model maintained sentiment polarity and intensity as it transitioned through languages with similarly culturally informed contexts (referring to English to Afrikaans). But such were Zulu and Xhosa languages, sometimes idiomatically expressing sentiment drift especially where no direct translation existed.

2. **Limitations and Improvements:**

   Sometimes the existing lexicon was not comprehensive enough to give us complete word coverage for neutral sentiment so sometimes defaults were translated as positive or negative sentiment. Increasing the number of neutral and culturally specific words with which to expand the lexicon could lead to better sentiment accuracy. To make future iterations of the prediction more nuanced, a machine learning approach to refine the translation prediction might be applied based on context, for example, to make classification of sentiment across languages more nuanced.

3. **Translation Verification:**

   Sample phrases were back translated to make sure the translated sentences retained the same meaning and sentiment as the original were. These revealed that although most sentiment was preserved, a few phrases held variations of emotional tone based on changes to linguistic structure and phrase interpretation.

Summary

The sentiment analysis and translation testing were successful in showing how sentiment can remain intact through most language pairs on a structured lexicon based workflow. Though slight in cultural expression, lexicon challenges, the results were effective for multilingual sentiment analysis. Future work includes enlarging the lexicon and utilising context based machine learning adjustments towards more accurate translation.

7. DISCUSSION

Interpretation

The use of the models for classifying sentiment between South African languages showed significant differences in how well they can perform multi linguistic sentiment analysis. The most robust model, Random Forest, obtained balanced accuracy and high F1 scores for positive sentiments. Thus, this finding implies that Random Forest's ensemble nature enables it to effectively capture the complex patterns in the multilingual dataset, which includes variations in sentiment expression across different languages.

Support Vector Machine had similar performance compared to Random Forest on positive sentiment classes, while underperforming on neutral and negative sentiment classes. This suggests that SVM which usually excels in separating well defined classes is ideal for senti ment task in which the boundaries are well defined. Like Random Forest, SVM struggled with neutral sentiments, with the challenges of sentiment ambiguity in multilingual contexts.

A simpler linear model, Logistic Regression performed well at detecting positive sentiment but shared a clear bias towards predicting positive sentiment and poor classification of neutral and negative sentiments. This limitation indicates the model is not able to capture subtle, language specific differences in sentiment which linear models typically lack in the ability to encode for complex multilingual data.

Decision Tree did a slightly better job at handling negative sentiments than Logistic Regression, having higher recall for this class. This improvement shows that Decision Tree can somewhat adapt to multilingual data, but Random Forest is much more robust, in the sense of handling fine differences across sentiment classes.

Finally, Naive Bayes was dramatically off with overall accuracy because of the simplicity coupled with the inherent dataset biases. Data imbalances were likely causing the model to classify terms negatively. The results indicate this underperformance is an indication of the Naive Bayes' weakness as a general method for multilingual sentiment analysis, especially in high variability and sensitive emotional expression dataset.

Insights and Recommendations

Ensemble Models for Enhanced Performance: We show that ensemble methods in particular (Random Forest) are appropriate for multilingual sentiment classification given good robustness across classes. Ensemble models should be prioritised for similar applications in the multilingual environments.

Challenges with Class Imbalance: Even with SMOTE balancing classes, issues persisted, especially on neutral set of sentiments. We anticipate that future work should investigate more sophisticated techniques for class balancing or generate additional data with a variety of sentiment intensities to reduce the gap in model performance between underrepresented and overrepresented classes.

Limitations of Simpler Models: In this complex task, simpler models like Logistic Regression and Naive Bayes were overall not adequate and we see the need for more sophisticated models or neural networks to handle nuances of multilingual sentiment analysis.

Expansion of Sentiment-Rich Lexicon: Model sensitivity to nuanced emotional expressions could have been reduced by the absence of sentiment rich adjectives in the lexicon. A wider lexicon, including culturally relevant adjectives and expressions, might increase accuracy — especially at capturing the fine points of sentiment.

Limitations

However, several limiting factors were encountered by this study and may be related to the results. The most important issue is that there is already an inherent imbalance in the sentiment representation on the dataset, so we first start by using SMOTE to create synthetic sampling of the dataset since it was useful but didn't fully reflect the natural variability in sentiment. We also went through the translations manually as well as back translated them, but since given languages might not translate some complex terms or idiomatic expressions, some of this sentence might have lost the sentiment accuracy. Limitations were also imposed by the need to rely on simple models such as Naive Bayes and Decision Tree, and, given the depth needed to accommodate complex language structures and their potential dependencies between the terms, the use of such simple models may have been hampered. Finally, the limited iteration of the lexicon in South African languages may limit its generalisation to varied settings as domain variants in sentiment expression need further customization and extension.

Comparison with Existing Research

The present work aligns with the findings of Abbasi et al. (2008) and Xie et al. (2014), which found that multilingual SVs promote the use of language specific feature engineering, in particular, which is necessary, and robust classifiers in multilingual settings. As in Abbasi et al.'s (2008) research, this study corroborates that such ensemble models as Random Forest effectively operate with complex, multilingual datasets because of their adaptability and tolerance for much dimensional data. Specifically, Oriola & Kotzé (2020) noted the difficulty of sentiment analysis in African languages because such languages have fewer lexicons and data resources, which are addressed here by expanding the lexicon to include several South African languages. For instance, this work extends the proof of concept of sentiment analysis for Africa’s low-resource languages beyond the studies of existing works that only focused on higher-resource ones, showing that low-resource languages, with suitable model choice and data pre-processing, can also provide good sentiment analysis. The manual validation of translations and back translations in this study coincide with the view expressed by Davies and Gardner (2010): existing literature often disregards the way cultural specifics are actually present in multilingual data.

In a nutshell, the results here demonstrate that ensemble models like Random Forest perform very well on multilingual sentiment classification with tailored lexicons and balanced data. But it also highlights the fact that additional lexicons, in particular, lexicons with specific sentiment adjectives, are important in achieving greater accuracy in the sense and context based translations. These findings help to extend sentiment analysis tools for low resources languages, in pursuit of culturally aware, inclusive sentiment analysis across linguistic settings.

8. CONCLUSION

Summary

Our multilingual sentiment classification successfully extends to an expanded set of South African languages and we demonstrate that Random Forest and SVM are well suited for multiple language sentiment classification. Precision, F1 scores, accuracy, and overall AUC scores consistently gave highest values to Random Forest, which indicates that its ensemble structure, because of being high enough to capture the same complexity as multilanguage data where sentiment expresses itself depending on language. Also, SVM did very well at distinguishing between positive and negative sentiments, which corresponds with SVM's strength of maximising the margin between classes. Although simpler models like Logistic Regression, Decision Tree, and Naive Bayes were less effective meaning they may not be flexible enough to accommodate fine text nuance and sentiment fluctuation across a multilingual dataset, they could be useful provided the two issues were addressed. According to the POS Distribution analysis, the limitations of the lexicon’s dependency on nouns and verbs may prevent the model from being sensitive to sentiment expressing adjectives, without which the emotion can be reflected in the true sense. Finally, these results suggest that robust and ensemble models, coupled with carefully balanced datasets and expanded sentiment lexicons, can provide a viable approach to sentiment analysis on languages with low resource levels.

Recommendations

Based on the findings, the following recommendations are proposed:

Expand the Lexicon with Sentiment-Rich Adjectives: Next, if the models can be made to sufficiently represent the contents of languages with more adjectives and other types of sentiment prone parts of speech, then their ability to identify diverse sentiments will increase. It may vary from adding South African derived idiomatic expressions and culturally specific words that 'feel' deeper emotionally in South African languages.

Apply Context-Sensitive Translation Validation: For reliable sentiment classification we have to make sure that translated terms are accurate, especially where they are terms that have cultural or emotional specificity. Complex or idiomatic phrases are just the right test cases to explore for advanced translation models, e.g., neural machine translation with attention mechanisms, to increase accuracy.

Prioritise Robust, Non-linear Models: For soon to come projects, if the project requires multilingual sentiment classification then ensemble models such as Random Forests or deep learning alternatives should be prioritised. Baselines will almost always be simpler models, but they are almost never going to be high accuracy models for these nuanced, multilingual environments.

Future Work

Future research could explore several areas to build upon and improve these findings:

Enriching Data Sources: A larger dataset should include more diverse sources of text such as social media posts, news articles and informal texts in African languages, which would provide a full coverage of sentiment. This would help the model generalise much better to different contexts and better reflect reality's gestalt sentiment.

Exploring Deep Learning Models: Random Forest, however, seems to work well, but with a richer language data context or even more powerful models such as transformers, LSTM networks, or attention based ones, the classification could be further improved. We find that these models are particularly well suited for languages with high syntactic and semantic variance.

Investigating Cultural Nuances in Sentiment: Studies about the determination of sentiment in future is focused on the impact of cultural contexts; for instance through cultural indicators or on the variation in sentiment in cultural expression in the South African languages. This further research would allow such sentiment tools to be more culturally sensitive with more depth.

The results from this study then demonstrate the feasibility and potential of a multilingual sentiment analysis over South African languages, and the importance of robust models and tailored lexicons in such cases. Improved translation validation and a potentially more sophisticated set of models can turn sentiment analysis into a potent, inclusive resource for recording the contextualised emotional words of many different linguistic and cultural worlds.

9. REFERENCES

Abbasi, A., Chen, H., & Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3), 1-34. https://doi.org/10.1145/1361684.1361685

Davies, M., & Gardner, D. (2013). *A Frequency Dictionary of Contemporary American English: Word Sketches, Collocates and Thematic Lists*. Routledge. https://doi.org/10.4324/9780203880883

Oriola, O., & Kotzé, E. (2020). Evaluating Machine Learning Techniques for Detecting Offensive and Hate Speech in South African Tweets. *IEEE Access*, 8, 21496-21509. https://doi.org/10.1109/ACCESS.2020.2968173.

Xie, Y., Chen, Z., Zhang, K., Cheng, Y., Honbo, D., Agrawal, A., & Choudhary, A. (2014). MuSES: Multilingual Sentiment Elicitation System for Social Media Data. *IEEE Intelligent Systems*, 29, 34-42. https://doi.org/10.1109/MIS.2013.52