UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

South African isiZulu and siSwati News Corpus Creation, Annotation and Categorisation

by

Andani Madodonga

Submitted in partial fulfillment of the requirements for the degree
Masters in Information Technology (Big Data Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria, South Africa

July 2022

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Publication data:

Andani Madodonga. South African isiZulu and siSwati News Corpus Creation, Annotation and Categorisation. Master's dissertation, University of Pretoria, Department of Computer Science, Pretoria, South Africa, June 2022.

Electronic, hyper-linked versions of this dissertation are available on-line, as Adobe PDF files, at:

http://dsfsi.github.io//

https://repository.up.ac.za/

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

South African isiZulu and siSwati News Corpus Creation, Annotation and Categorisation

by

Andani Madodonga
E-mail: u18114564@tuks.co.za

Abstract

South Africa has eleven official languages and amongst the eleven languages only 9 languages are local low-resourced languages. As a result, it is essential to build the resources for these languages so that they can benefit from advances in the field of natural language processing. In this project, the focus was to create annotated datasets for the isiZulu and siSwati local languages based on news topic classification tasks and present the findings from these baseline classification models. Due to the shortage of data for these local South African languages, the datasets that were created were augmented and oversampled to increase data size and overcome class classification imbalance. In total, four different classification models were used namely Logistic regression, Naive bayes, XGBoost and LSTM. These models were trained on three different word embeddings namely Count vectorizer, TFIDF vectorizer and word2vec. The results of this study showed that XGBoost, Logistic regression and LSTM, trained from word2vec performed better than the other combinations.

Keywords: South African Local Languages, Low Resources Languages, Data Augmentation, Topic Classification, Logistic regression,Naive Bayes, LSTM, XGBoost, Count Vectorizer, TFIDF Vectorizer, Word2vec.

Supervisors	**:**	Prof. V. Marivate
		Dr. M. Adendorff
Department	**:**	Department of Computer Science
Degree	**:**	Masters in Big Data Science

© University of Pretoria

Acknowledgments

- I would like to thank my family[Ndivhuwo Madodonga, Violet Madodonga, Mavhungu Madodonga and Masala Malotsha] for the support and words of encouragement.

- I would like to thank my supervisors, Prof V. Marivate and Dr M. Adendorff for being patient with me, offering proper guidance and courage from the beginning until the completion of this dissertation. I'll forever be thankful.

- I would like to thank my friends, Phillemon Senoamadi, Sello Matjie and Chris Nembidzane for the support and technical assistance they gave me since the beginning of my masters up until the completion. Much appreciated.

- I would like to thank Sedzani Mathoho for the inspiration and being the go-to person when I am not certain about school work and bursary applications. Much appreciated.

© University of Pretoria

Contents

© University of Pretoria

© University of Pretoria

© University of Pretoria

List of Figures

© University of Pretoria

© University of Pretoria

© University of Pretoria

List of Graphs

© University of Pretoria

List of Algorithms

© University of Pretoria

List of Tables

© University of Pretoria

Chapter 1

Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence, linguistics and computer science that focuses on making computers understand natural languages[1]. With the help of NLP, intelligent machines are built and people are benefiting from them. One of the cases where NLP has been beneficial to people is where it has been used for machine translations, performing the task of translating from one language to another. In this case, NLP helps the computer or machine to understand conversion between the two languages. NLP can also assist in learning sentiment from sentences or text and this NLP capability is utilized by companies to understand how customers feel and their opinion about the company's products and services through the analysis of their social media posts and comments. Furthermore, the chatbots that are used in the customer services space are one of the examples of NLP application [1].

Contextual chatbots and Virtual Text Assistant are now widely used but they mostly understand a limited number of languages, such as English. South African local languages do not have enough resources to be used to built such contextual Chatbots and Virtual Text Assistant. Therefore, the resources for local languages need to be created so that they can be used to build software agents that understand South African local languages [2].

South Africa is a multilingual country with nine African and two European languages; the African languages are Sepedi, Sesotho, Setswana, siSwati, Tshivenda, Xitsonga, isiZulu, isiNdebele and isiXhosa and on the other hand, European languages are

© University of Pretoria

English and Afrikaans. It is important to note that these languages are official in South Africa [3]. In South Africa, we have a challenge with the nine African languages because they are resource-poor. There is a shortage of curated and annotated corpora to enable them to benefit from Natural Language Processing. Therefore, the purpose of this study is to focus specifically on the corpus creation and annotation for isiZulu and siSwati and perform a topic classification tasks on the data.

1.1 Motivation

Within the context of South Africa, there are eleven official languages of which English, Afrikaans and isiZulu are the most represented languages across various news article publications, Although isiZulu is the third most represented language in the country;it is still low-resourced language since the availability of data is minimal on internet[4]. Most low-resourced languages do not have enough digitized text material and sometimes it is due to the small population size of the people who speak that language, but that is not always the case because in some cases, the languages that are spoken by a large-sized population may not have enough digitized text materials, hence those languages are called low-resourced languages [5]. Processing low-resourced languages that have a shortage or no annotated data and have a smaller size of native speakers, etc. is a challenge in Natural Language Processing [3].

There are over 1250 languages in Africa and recently most of them are getting attention in NLP space, the common approach that is used to create data for the low-resources language is cross-lingual word embedding[6] which is the joint representation of words of the two or more languages in one vector space to be able to compare the semantic meaning of the words across the languages and also to perform language transfer learning between the languages, this is usually done between low-resourced and resource-rich languages. However, the cross lingual transfer does not perform well in the case of machine language translation[7]. Building systems that can communicate with people in their native language is one of the reasons why it is necessary to tackle NLP problems, for instance, the development of a system that can read the news that was written in a

© University of Pretoria

different language; or a system that can accurately perform question and answer in any language. However, building such systems for low-resourced languages is a challenge. Moreover, creating a viable dataset for low-resourced languages and storing the information in a common repository like SADILAR (South African centre for digital language resources), will assist enhancing low-resourced languages for NLP tasks and strengthen the field [6].

The creation of resources for resource-poor languages presents the opportunity for those languages to be incorporated in modern technologies such as translation systems and other technologies, enable the communication and increase access to information between groups that speak different languages since the information will be presented in any language of your choice[8], for instance, English news can be translated into siSwati or any other language so that even people who cannot read English also understand the same information. In conclusion, Building resources for low-resourced languages have a good impact on the NLP research community and enables the languages incorporation into the set of NLP capable technologies. Hence, it is imperative to develop the resources for low-resourced languages [9].

1.2 Objectives and Contributions

The broader focus for this work is to create NLP resources for the South African low-resourced languages, where the language focus is on isiZulu and siSwati. Therefore, the below items are contributions to resource building:

- Creation of curated datasets for isiZulu and siSwati
- Annotation of datasets for isiZulu and siSwati that can be used to train classification models
- Demonstration of Data Augmentation and OverSampling(SMOTE) on annotated isiZulu and siSwati datasets to mitigate class-imbalance problem and increase data size
- Creation of topic classification and set baseline models for isiZulu and siSwati languages

© University of Pretoria

The data for isiZulu and siSwati will be scrapped from the internet and curated, then made ready for annotation. The annotated datasets will therefore be augmented and oversampled to increase the their size and balance the class categories prior to be used to perform topic classification and set baseline models.

Creating and annotating corpora for isiZulu and siSwati and then performing text classification will open the path for the other researchers to study further and create more resources for these low-resourced languages. This work will provide resources that other researchers can use to build downstream applications. The built resources for these two languages are; Annotated datasets which is the labelled data, Pre-trained vectorizers which are word vectorizers that are already trained on isiZulu and siSwati and Baseline Models which are simple models that present minimum expected performance in a similar classification task.

1.3 Dissertation Outline

A description of the material covered in each chapter is included below:

- **Chapter 2** focuses on the prior work that has been done and techniques/methods that are utilized in the low-resourced languages.
- **Chapter 3** covers the technical background of the methods that are applied in this work.
- **Chapter 4** covers the non-technical applied methodology together with the outcomes of the exploratory data analysis and unsupervised modeling that has been done in this work.
- **Chapter 5** covers the results obtained in this work from the supervised models.
- **Chapter 6** gives the overall conclusion derived from the results and also outlines the necessary future work.

Short introduction of each appendix:

- **Appendix A** provides information about the datasets that are used in this work.

© University of Pretoria

1.4 Publications

I have an accepted paper

© University of Pretoria

Chapter 2

Literature Review

In this chapter, we explain NLP critical components that are required to build resources for low-resourced language, and explain the prior work on NLP projects, data generation/sampling, model building and model evaluation methods.

Section 2.1 covers the components for building the language resources, Section 2.2 covers the models and evaluation techniques that are utilised in the NLP arena, Section 2.3 defines and explains Data Augmentation, SMOTE oversampling and their application, Section 2.4 explains some of related prior work that has been done on low-resourced languages and Section 2.5 provides a summary of this work.

2.1 Critical Natural Language Processing Components

Globalisation and the increase in digital communications have created the demand for NLP systems that enable fast communication across different language-speaking people. However, some languages are missing in these systems. For instance, there are roughly 7000 spoken languages on the planet. Most of them still are not included in the NLP systems, primarily because they do not have the labelled corpora to build those NLP systems [10]. These languages with scarce or no resources are low-resourced languages [11]. The language resources include (but are not limited to) the annotated corpora and core technologies. Examples of core technologies include lemmatisers, part of speech

© University of Pretoria

tagger and morphological decomposers [9].A lemmatizer is a tool that finds the inflected form of the word [12], part of speech tagger is a tool that identifies the part of speech that the words belong to, where part of speech can be a verb, adverb, pronoun, noun etc. [13]. On the other hand, the languages with high resources are the ones that have most of the resources needed to build the NLP technology [14].

The high-resourced languages include English, French, Finnish, Italian, German, Mandarina, Japanese, etc. [15] [14] and low-resourced languages include languages such as isiZulu, isiXhosa, siSwati etc. [16]. The study focused on the low-resourced languages, namely, isiZulu and siSwati; Eiselen and Puttkammer [9] stated that annotated corpora are one of the things that low-resourced languages lack. Thus, the isiZulu and siSwati datasets need annotation as part of enriching these two languages. Hsueh, Melville, and Sindhwani [17] defined data annotation as the process of labelling the dataset(s), an important step when building machine learning models. Stenetorp et al. [18] stated that manual data annotation is the most important, time-consuming, costly, and tedious task for NLP researchers. Therefore, automation tools are developed to perform these annotations. Computers and machine learning models don't understand texts like humans do [19], hence the data has to be represented in a vector form called word embedding.

The vector representation (word embedding) caters for the semantic and syntactic relationship of the words [20] and in a lower dimension space [21]. There is an increase in number of word embeddings that are being developed [22]. These word embeddings include the bag of words model, term-frequency inverse-document frequency(TFIDF), and word2vec among others [23] [24]. For this study, the text datasets for isiZulu and siSwati are transformed into word vector representation using three different word embeddings. These are the bag of words, TFIDF and word2vec. The purpose is to build resources for the South African local languages.

Text classification identifies the category that a textual document belongs to. It is applied on digital documents. Text classification requires a machine learning algorithm [25]. The algorithm can either be supervised or unsupervised learning algorithms (or others), that is, the data can have input data and corresponding expected results(supervised learning). When there are input data without the desired results, we have unsupervised learning scenario[26].

Moreover, frequently used machine learning algorithms are Decision trees, Naïve Bayes, Rule Induction, Neural Networks, Nearest Neighbours, Support Vector Machines(SVM) etc. [26]. The machine learning algorithms used for this work will be Logistic Regression, LSTM. Naïve Bayes is the baseline for classification. Figure 2.1 below shows text classification process from data collection to modeling.

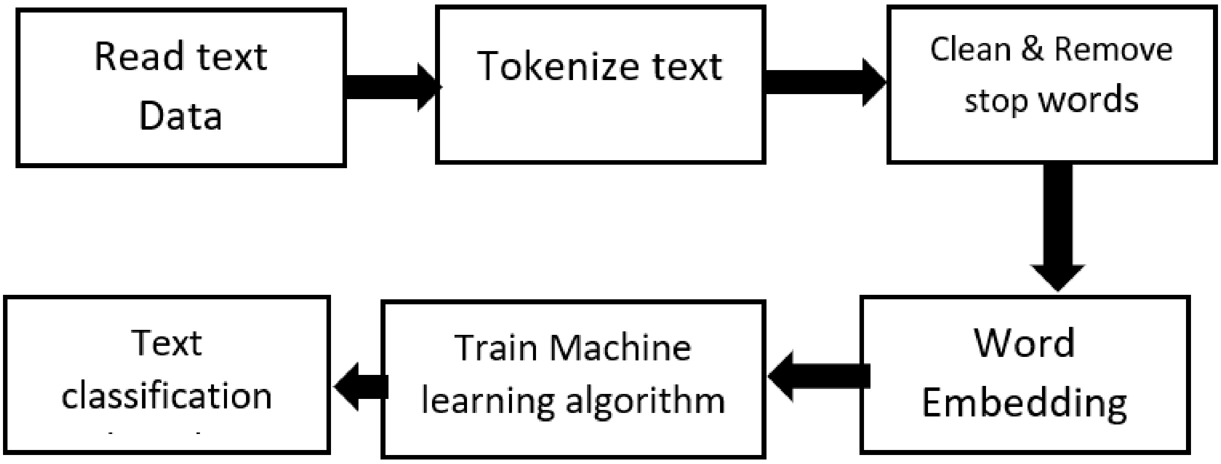


Figure 2.1: Data preparation steps for text classification.

The lack of curated and annotated data impede the process of fighting the shortage of resources for low-resourced languages in the NLP space[27]. Besides, established NLP methods often cannot be transferred on or to these languages without these corpora[27].Therefore, annotated data is a resource for language technology.

Niyongabo et al. [27] collected the datasets of two closely related African languages - Kirundi and Kinyarwanda from two different sources. A total of 21268 and 4612 articles were annotated for Kinyarwanda and Kirundi respectively. The two datasets underwent a cleaning process that involved the removal of special language characters and stopwords. The removal of special characters cleans non-alphabetical characters (e.g. @!) and URLs from the textual dataset [27]. On the other hand, stopwords removal excludes the words that carry little information in the text such as ‘is’, ’at’, ’the’ etc. [28]. The removal of

© University of Pretoria

stop words during data pre-processing improves performance in downstream NLP tasks [29].

The sources were newspapers and websites. These datasets were annotated, based on the title and content of the contained articles, into the following categories:

Politics
Sport
Economy
Health
Entertainment
History
Technology
Tourism
Culture
Fashion
Religion
Environment
Education
Relationship.

2.2 Model and Evaluation techniques

Machine learning refers to the learning of tasks without being directly programmed. Machine learning algorithms achieves machine learning[30]. Many an algorithms solves different problems. They are grouped based on how they learn from the data, namely, supervised, semi-supervised, unsupervised[30]. To expand, supervised algorithms predicts the output given an input, using inference from the labelled training data set, for example, Naïve Bayes, Decision tree, support vector machine, etc[30]. Unsupervised algorithms refer to the algorithms predicts the output of the input data when there's no available labelled training data. In this case, the algorithm discovers the patterns from the data on its own, such as, k-means clustering, principal component analysis etc[30].

© University of Pretoria

On the other hand, semi-supervised algorithm uses both labelled and unlabelled data to perform the predictions[30]. Examples of these algorithms include generative models, self-training models etc.[30].

The machine learning algorithms require evaluation after performing a prediction or classification task. Metrics like F1-score and confusion matrix measures performance [31]. In general, F1-score measures the performance of the classifiers. The confusion matrix summarises classification results in a tabular format[31].[32].For instance, in the case of binary(0 and 1) classification, the confusion matrix will look like the one below in table 2.1 [33]

Table 2.1: Confusion Matrix [33].

	Actual class 0	**Actual class 1**
Predicted class 0	True Positive(TP)	False Negative(FN)
Predicted class 1	False Positive(FP)	True Negative(TN)

To derive the F1-score, the recall and precision must be computed first. Recall refers to the measure of how many of all positive classes were predicted correctly and whereas, precision outputs the value that tells us that, out of all the predicted classes that are positive, how many are indeed positive[33]. The other metric that is commonly used in Natural Language Processing systems for performance measurement is the BLEU(Bilingual Evaluation Understudy) score [34], which is a measure of similarity between the machine translated texts and the expected texts ranging from 0 to 1 strength[35].

2.3 Data generation techniques for low-resourced languages

The existing approach that is utilised to mitigate the challenges of low-resourced such as shortage of data, is the language translation approach, that is, the low-resourced language gets translated into the resource-rich language[36]. However, in most cases, this approach suffers from language biases and may be impractical to achieve in real life[36]. Sometimes the direct translation may be impossible or inaccurate due to language dif-

© University of Pretoria

ferences. Hence, the translated data will require manual processing thereafter, which is tedious and time-consuming. Manually creating data for low-resourced languages is time-consuming but a good approach [37]. It introduces minimal language biases and more accurate than translated datasets[37].

Cross-lingual and transfer learning is one of the combinations of techniques frequently used or preferred in NLP due to its speed and efficiency. However, it works best on languages closely related to each other because it transfers the word embedding of the resource-rich language to the low-resourced language[37]. This further serves to highlight why all languages must have NLP resources such as annotated data to avoid data simulations that have unfavourable effects.

Data Augmentation is a method that generates a copy (or unique data) of the data by slightly altering the existing data [38]. It increases the size of small training data in ways that improve model performance[39].Model performance is highly dependent on the quality and size of the training data. Data Augmentation addresses the issue of small training data that leads to the models losing their generasibility[40].

Marivate et al. [4] had a small data size of Sepedi and Setswana local languages, and incorporated word embedding(word2vec) based-contextual augmentation to increase the dataset used to train classification models. Each training dataset is augmented 20 times while the test dataset remains unchanged. In their study, the new data created replaced the words (based on context) in the sentences. Hence a new sentence is formed. Furthermore, the Data Augmentation improved the performance of the classifiers [4]. In this current study, the same Data Augmentation (word embedding-based augmentation) will be performed on the siSwati and isiZulu dataset to increase the training data size.

Rizos, Hemker, and Schuller [41] adopted a different Data Augmentation technique based on three items: (a) synonym replacement based on word embedding vector closeness, (b) warping of the word tokens along the padded sequence or (c) class-conditional, recurrent neural language generation. This augmentation was applied to the social media hate-speech dataset to minimise the imbalance of target classes and maximise the information on the text. Moreover, this technique performed better than the baseline technique.

The Synthetic Minority Oversampling Technique (SMOTE) is a technique adopted where the learning is done on an imbalanced dataset since it solves the problem of class imbalance[42].SMOTE works by generating synthetic examples through inserting different values(words) of minority classes chooses them from a defined neighbourhood within feature space, that is, a minority class is selected, then obtain the k-nearest neighbours of the same minority class and therefore utilises the k- neighbours to create the new synthetic examples.[42]. Rupapara et al. [43] implemented the SMOTE technique when performing the classification of toxic comments extracted from social media platforms and microblogs websites, the dataset contained two classes, namely, toxic and non-toxic. While the toxic classes had 15294 comments, the non-toxic classes had 143346 comments. The lower ratio from the non-toxic classes creates a class imbalance. Bag of words and TFIDF vectorizers were trained to create a feature space for the implementation of SMOTE. SMOTE eliminates the class imbalance in the dataset[43].

2.4 Prior work on Low-resourced languages

Supervised learning models perform better on larger labelled datasets, which presents a challenge for low-resourced languages as they don't have enough data and annotating data can be expensive[44]. Most prior studies focused on developing parallel corpora between low and resource-rich languages, but parallel corpora are often unavailable for some low-resourced languages[44]. Zoph et al. [45] identified low-resourced languages and investigated the idea of distance learning on machine translation. Since English and French are resource-rich languages, the two languages trained a neural machine translation (NMT)[45]. The NMT trained on a language pair-the initial model trained on English-French pair. Afterwards, the NMT model initialized another NMT model to be used on a low-resourced and high-resourced pair(e.g. Uzbek-English)[45]. The technique used in this study is called transfer learning which means that knowledge learnt from another task is applied to the other task to improve the performance. In this case, the low-resourced languages investigated were Uzbek, Hausa, Turkish and Urdu. The study combined an encoder-decoder, with Long- short term memory to allow the decoder to propagate back to the encoder[45]. As a result, the transfer learning improved the BLEU (bilingual evaluation understudy) for low-resourced Neural machine

© University of Pretoria

translation [45].

Nguyen and Chiang [46] explored transfer learning between the two low-resourced languages Turkey and Uzbek by first pairing each language with English and then generating the parallel data. Then, split the words with Bytes Pair Encoding (BPE) to maximise the overlapping vocab[46]. The model and word embedding are trained on the first language pair (Turkey-English) and then the same model parameters and word embeddings were transferred to the other model that trained the second language pair (Uzbek-English). This technique improved the BLEU by 4.3% [46]. Parallel corpora are not always available; hence the low-resource tagging technique was proposed. The technique utilised the bilingual dictionary, monolingual corpora of high and annotated low-resourced languages[44]. The bilingual dictionary and monolingual corpora assisted the cross-lingual distant learning method, thus, removing the need for parallel corpora[44]. For instance, the bilingual dictionary will act as annotations for the monolingual dataset[44]. The neural network was trained and evaluated on the dataset and compared with the other models(trained using parallel corpora), the models include: Minitagger, bidirectional long short-term memory (BiLSTM) and bidirectional long short-term memory-Conditional random field (BiLSTM-CRF)[44]. The neural network(new technique) outperformed the rest of the benchmark methods for low-resourced languages[44].

The datasets of low-resourced South African languages, isiZulu collected from isolezwe and National Centre for Human Language Technology; and Sepedi collected from National Centre for Human Language Technology were used to evaluate the performance of open-vocabulary models on the small datasets, the evaluated models include n-grams, LSTM, RNN, FFNN, and transformers. The performance of the models was evaluated using the byte pair encoding (BPE). BPE uses the subword based tokenisation splitting the rare words into smaller meaningful words. For instance, ‘girls’ will be split into ‘girl’ and ‘s’ to make the model understand that the word ‘girls’ derives from the word ‘girl’; and as a result, the RNN performed better than the rest of the models on both the isiZulu and Sepedi datasets [47].Nyoni and Bassett [48] explored the machine translation capability from the zero-short learning, transfer learning and multilingual learning on

© University of Pretoria

two South African languages, namely, isiZulu and isiXhosa; and one Zimbabwean language, that is Shona. The datasets were in language pair(parallel text), that is, English -to- Shona, English -to- Zulu, English -to- Xhosa and Zulu -to- Xhosa, with the pair English -to- Zulu being the target pair since it has the smallest datasets(sentence pair). The transfer learning and zero-short learning did not outperform the multilingual model which produced the Bleu score of 18.6 for the English-to-Zulu pair.Moreover, these results provide an avenue for the development and improvement of low resource translation techniques [48].

Marivate et al. [4] have identified and addressed the issue of lack of clear guidelines for low-resources languages in terms of collecting and curating the data for specific use in the Natural Language Processing domain. In their investigation, two datasets of news headlines written in Sepedi and Setswana were collected, curated, annotated, and fed into the machine learning classification models to perform text classification. The news headlines of Sepedi and Setswana datasets collected from online websites and social media platforms were counted to be 219 and 491(count of news articles) respectively and the datasets were annotated by means of categorising the articles into the following categories based on context:

Legal
General News
Sports
Politics
Traffic News
Community Activities
Crime
Business
Foreign Affairs.

The word embeddings, namely, TFIDF, word2vec, a bag of words and fasttext were constructed using the data from different sources. These sources include JW300, Bible and SADILAR so that news articles can be classified. Furthermore, machine learning

© University of Pretoria

algorithms for classification tasks were selected, namely, Logistic regression, Support vector classification, XGBoost and MLP neural network and the word embedding based contextual augmentation was applied to the datasets since the size of the datasets was small. The text classification used different machine learning models. The evaluation metric was the F1-score, which is a model performance measure. One of the models, Xgboost, performed well as compared to other models[4].

2.5 Summary

The current study is closely related to what Marivate et al. [4] have done. Our studies are similar in terms of annotation, and word embedding. Some of the classification models and the purpose of the study are similar. However, the difference is the focus of the languages. While they focused on Sepedi and Setswana languages[4], we worked on isiZulu and siSwati. The main aim is to curb the shortage of resources for South African low-resourced languages. We selected the two Nguni languages (isiZulu and siSwati). There were no criteria used in the selection of these two languages. The models used to perform the text classification are:

- Naïve Bayes
- Logistic Regression
- Xgboost
- LSTM

These models combine classical models with neural networks and machine learning algorithms. This provides variability in selecting the model that performs better in this problem, thus creating a baseline for both classical models and neural network areas. Moreover, the annotated datasets and three(3) word embeddings for both isiZulu and siSwati will be available for future researchers interested in further creating the resources for these two languages and other research usages. The outcomes of this study will be beneficial in the NLP research community as the annotated isiZulu and siSwati datasets with their word embeddings will be made public to other researchers to mitigate the lack of annotated corpora problem and enable annotated data and word embedding for these two low-resourced languages become easily accessible. In conclusion, this study addressed the lack of resources for the two South African local languages, namely, isiZulu and siSwati and created resources for them. It curated and annotated the corpora, and performed text classification . Therefore, the findings will provide a baseline for the other researchers interested in enhancing South African local languages in the NLP space.

In short, this current study will address the problem of lack of resources of the two South African local languages, namely, isiZulu and siSwati. The resources will be created, that is, the isiZulu and siSwati curated and annotated corpora, and perform text classification. Therefore, the findings will provide a baseline for the other researcher who are interested in enhancing South African local languages in NLP space.

© University of Pretoria

Chapter 3

Methodology Technical Background

This section focuses on the technicalities of methods that are employed in this study from the data collection until the model building and evaluation. This includes the data cleaning, word embedding creation, Data Augmentation, data split, text classification and model evaluation.

Section 3.1 covers the data cleaning process, section 3.2 explains each word embedding that was implemented in this work, section 3.3 explains the Data Augmentation process, section 3.4 explains the process to split the data into training and test sets, section 3.5 covers the mathematical part of the algorithms that are used in this work, section 3.6 covers the derivation of the model evaluation measure and lastly, section 3.7 summarises the whole chapter.

3.1 Data Cleaning

Data cleaning is the process of detecting, handling, or removing bad data; it is basically the process of dealing with the abnormalities in a data [49]. Removal of errors and inconsistency in data improves the data quality [50] and failure to perform data cleaning may result in performing analysis on the data that has errors which then lead to inaccurate results and conclusions about the data [51]. The data cleaning process was applied on the isiZulu and siSwati datasets to enable proper processing and improve the quality of the results. There's a shortage of language processing tools that can perform tasks such

© University of Pretoria

as tokenization, lemmatization and stop words removal for low-resourced languages [27], and isiZulu and siSwati also don't have processing tools, hence the English processing tools were adopted to achieve the data cleaning.

3.1.1 Stop words and Special character removal

Textual data from the internet usually comes with some noise, hence it is important to clean it before processing on the models. There are two aspects that are part of text data cleaning, that is, stopwords removal and special characters removal [27]. Stop words are the words that are meaningless to the analysis, though not intrinsically meaningless for the purposes of communication; and removing the stopwords reduces the noise [52]. The other data cleaning aspect is special character removal which refers to the removal of non-alphabetical characters such as "*!@ " [27]. Therefore, the isiZulu and siSwati stopwords were collated and extend with the list of English stopwords then used to remove the stopwords from the text data. Lastly, the special characters were removed from our data.

3.2 Word embedding

A word embedding is the representation of text into a word vector form [53] and word representation is the fundumental step in Natural Language Processing [54]. In this current study, three different word embeddings were used, that is, Bag of words, term frequency inverse document frequency and Word2vec. Each word embedding is explained below.

3.2.1 Bag Of Words

Bag of words is a simple text representation in machine learning that counts the appearance of each word in the documents regardless of the structure of the inputs such as paragraph, sentence, format e.t.c. The bag of words is created by initially tokenizing the texts, that is, splitting the texts in the documents into words; and secondly, building the vocabulary by collecting and numbering all the words in all the documents; and lastly,

© University of Pretoria

counting how often each word appears on the vocabulary.

The bag of words creation requires three steps, namely, tokenization, vocabulary building and encoding [55]. The above-mentioned steps are explained below using examples.

“UGXEKWE nxazonke umfundisi wodumo eThekwini ngokudayisa amapeni athandazelwe awakhangisa ngokuthi uma ulisebenzisa uzophasa ngamalengiso”

Step 1. Tokenization:
[‘UGXEKWE’, ‘nxazonke’, ‘umfundisi’, ‘wodumo’, ‘eThekwini’, ‘ngokudayisa’, ‘amapeni’, ‘athandazelwe’, ‘awakhangisa’, ‘ngokuthi’, ‘uma’,’ulisebenzisa’,’uzophasa’, ‘ngamalengiso’]

Step 2. Vocabulary building (overall documents):
[‘ngoba’,’Kuthiwa’,‘UGXEKWE’, ‘nxazonke’, ‘umfundisi’, ‘wodumo’, ‘eThekwini’, ‘ngokudayisa’, …… ‘amapeni’, ‘athandazelwe’, ‘awakhangisa’, ‘ngokuthi’, ‘uma’,’ulisebenzisa’ ,’uzophasa’, ‘ngamalengiso’]

Step 3. Encoding:
ngoba kuthiwa ugxekwe nxazonke… ulisebenzisa uzophasa ngamalengiso [0, 1, 1, 0 …, 1, 0, 1]

The last step results in a vector of word counts which is the numeric representation of the vocabulary [55]. In the above example, the tokenization process splits the text into one word, however, the text can be split into more than one word. The pairing of strings in sequence is called grams where a pair of two words is referred to as bigram, three words as trigram and so forth, generally the pairing of n-word is called n-gram [55]. In this current study, the bag of words was created using isiZulu and siSwati datasets independently, to obtain the numeric representation of each dataset and later utilized the machine learning algorithms to perform the text classification.

© University of Pretoria

3.2.2 Term Frequency Inverse Document Frequency (TF-IDF)

TFIDF is another text representation that represent texts in a vector form, however, it weighs how often the word appears in the document, therefore, if the word appears frequently in a particular document then it is informative in that document. TFIDF creates the bag of words and transforms it into the term frequency-inverse document vector [55]. The term frequency of a word in a document is calculated by getting the count of each word in a document and the inverse document is calculated by the occurrence of the word in all the documents. These measures are utilized to obtain the TFIDF score which is given by the formula

- T -term
- d-document
- D -documents
- f-frequency
- tf-term frequency
- idf-inverse document frequency
- tfidf-term frequency inverse document frequency

tfidf(t,d,D)=tf(f,d).idf(t,D)) (3.1)

where

$$(tf(t,d) = log(1 + freq(t,d)) \tag{3.2}$$

and

$$idf(t,D) = log(\frac{N}{count(d \in D : t \in d)}) \tag{3.3}$$

© University of Pretoria

The tfidf score is interpreted as follows: The score approaches zero if the word is appearing frequently in all the documents and scores approaches 1 if the word doesn't appear frequently in the documents [56]. Again, this approach will be applied in isiZulu and siSwati dataset in order to obtain the vector representation of the data in the above explained version before performing the text classification.

3.2.3 Word2vec

Word2vec is a word embedding technique that represent the words in a vector form, where each word is linked to one vector and that vector stores the characteristics of the words as compared to the other words; the characteristics of the words refer to the context, definition, semantic relationship of the word. Moreover, the vector representation of each word is then become the input or output in a neural network based on the chosen architecture, that is, CBOW (continuous bag of words) or Skip-gram (continuous skip gram).

CBOW:

Word2vec is capable of grouping together associated words and get the meaning of the words based on the position of the word on the text. CBOW uses the words surrounding the target word to predict the target word, for instance, given a sentence [We are happy this side], then the surrounding words can be used to predict the word 'are' as the target word, that is, the input [we, happy, this] can be utilized to predict the target word 'are'. Therefore, the surrounding words are used to predict the middle word [57]. Below Figure 3.1 shows the graphical representation of how the CBOW Model works in the background and illustrates that the model consists of fully connected two layers neural network, the two layers are hidden layer and output layer. The model make use of these two layers to make the prediction of target word given the context words, this happens through the process of feeding the context words to the hidden layer and each word from the context words go through the back propagation (training) while updating the error vector(and hidden layer weights) which will then be averaged/summed element-wise to obtain the output that will be fed into the activation function for probability score calculation, therefore, the output layer will consists of the predicted target word [58].

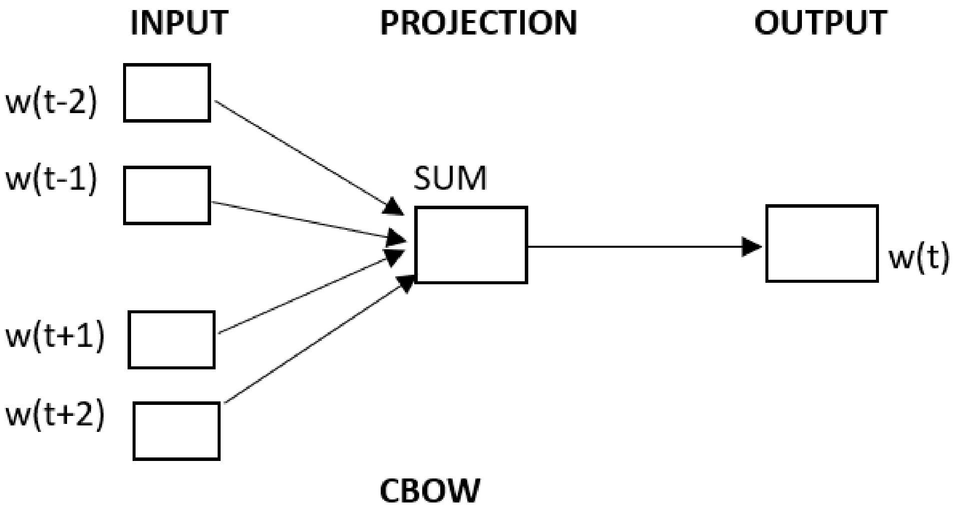


Figure 3.1: CBOW Model [59].

Skip gram:

Skip gram models do the opposite of the CBOW in terms of the predictions, that is, instead of using the surrounding words to predict the target word, it uses the target word to predict the surrounding words. Skip gram model predicts the word before and after the given word in the sentence, for instance, given a sentence, [We are happy this side], then the word ‘this’ can be used to predict surrounding words [happy, side][57].

Below Figure 3.2 shows the graphical representation of how the Skipgram Model works in the background and illustrates that the model consists of fully connected two layers neural network, the two layers are hidden layer and output layer. The model make use of these two layers to make the prediction of context given the input word, this happens through the process of feeding a word vector (input word) to the hidden layer for the back propagation(as part of training) while calculating the error vector associated to each target word and then use the cumulative error vector to update the hidden layer weights [58]. The hidden layer outputs are passed into the activation function for the computation of probability score of each target word and the vector of probability scores will be the output of the output layer [58].

© University of Pretoria

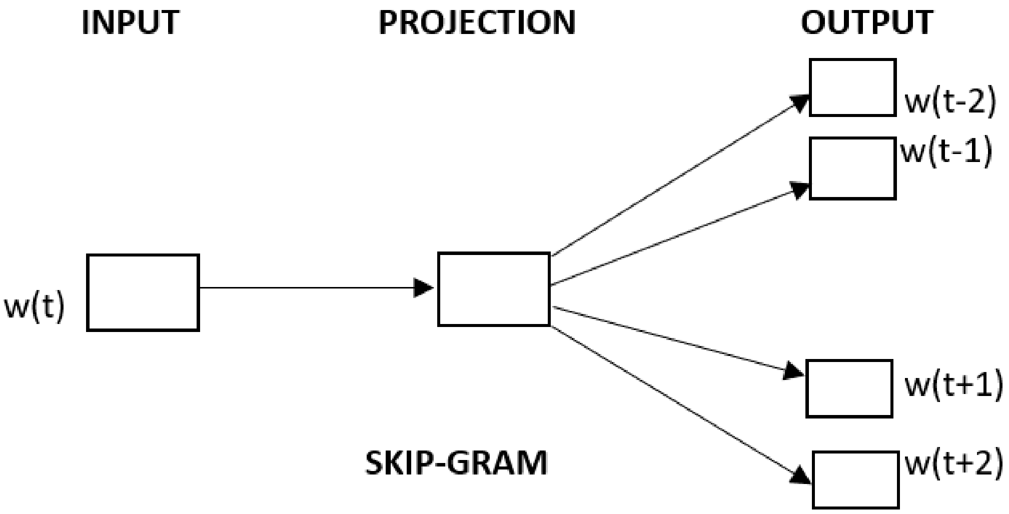


Figure 3.2: Skipgram Model [59]

3.3 Data Generation

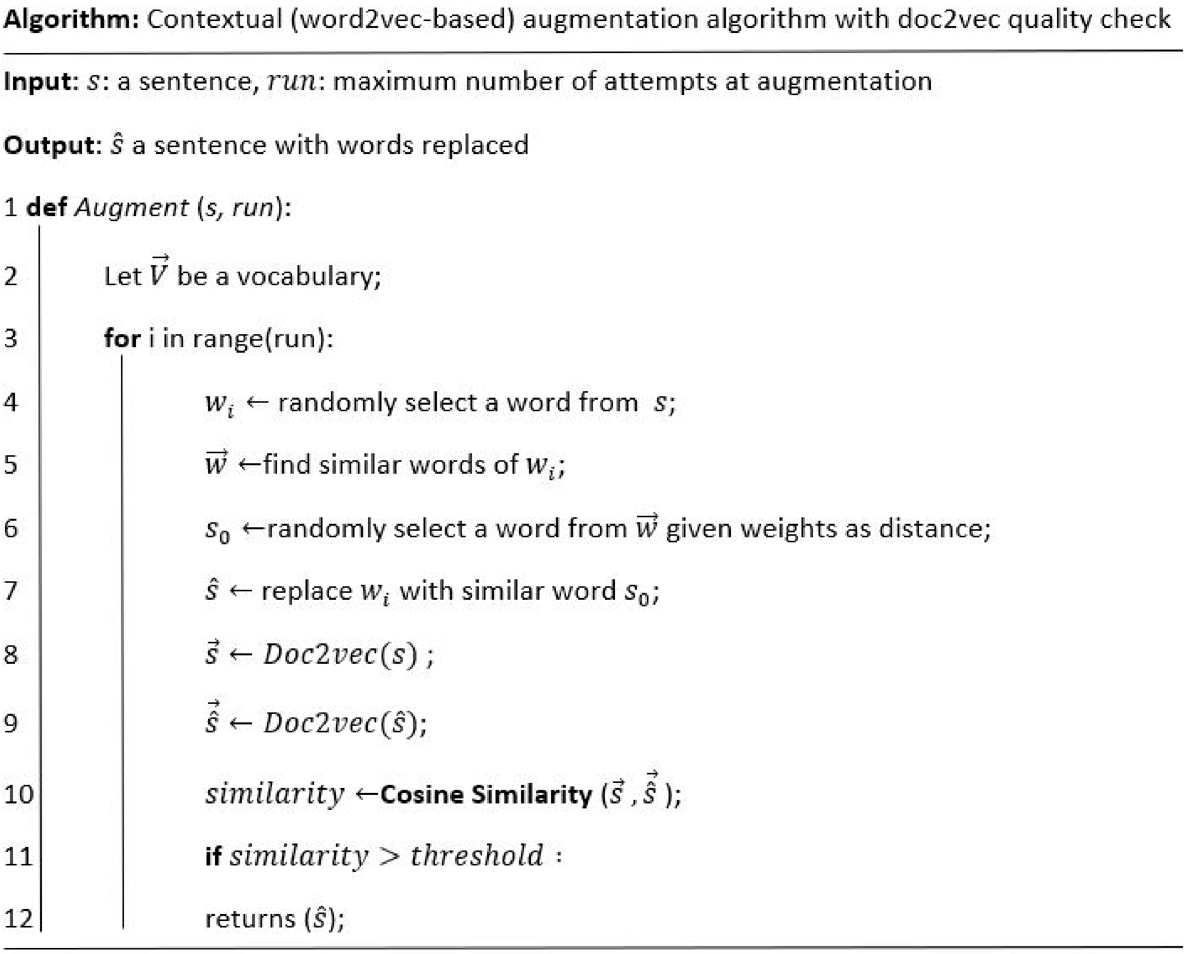
The size of the siSwati dataset is small due to the limited available siSwati dataset on internet, and the categories/classes are imbalanced and on the other hand, isiZulu dataset is enough but the categories/classes are imbalanced. Kobayashi [40] stated that the model generalization mostly depends on the size and quality of the data. Therefore, size of siSwati dataset must be increased for the classification models to be able to produce high generalization. The probelm of imbalanced classes needs to be addressed for both isiZulu and siSwati datasets, hence Data Augmentation and Synthetic Minority Oversampling Technique (SMOTE) explained below will be utilised to address the class imbalance and data size problems.

3.3.1 Data Augmentation

Data Augmentation is the process that is used to increase the data size to improve the performance of the machine learning classifiers [60]. The most common way to augment the data is by means of replacing the words or phrases in a sentence by their synonyms where the synonym is derived by obtaining the semantically similar/related words[61]

© University of Pretoria

but then Kobayashi [40] stated that the synonyms are limited and therefore, introduced a contextual based augmentation where the original word will be replaced based on the contextual meaning instead of the synonym[40]. Therefore, the siSwati and isiZulu datasets will be augmented using the same approach where the original words on the sentence are replaced based on their contextual meaning. The augmentation will be done through referencing the words similarity from the Word2vec word embedding, refer to the below algorithm 3.1 that will be adopted to achieve the augmentation



```
Algorithm: Contextual (word2vec-based) augmentation algorithm with doc2vec quality check
Input: $s$: a sentence, $run$: maximum number of attempts at augmentation
Output: $\hat{s}$ a sentence with words replaced
def Augment (s, run):
    Let $\vec{V}$ be a vocabulary;
    for i in range(run):
        $w_i$ ← randomly select a word from $s$;
        $\vec{w}$ ←find similar words of $w_i$;
        $s_0$ ←randomly select a word from $\vec{w}$ given weights as distance;
        $\hat{s}$ ← replace $w_i$ with similar word $s_0$;
        $\vec{s}$ ← $Doc2vec(s)$ ;
        $\vec{\hat{s}}$ ← $Doc2vec(\hat{s})$;
        $similarity$ ←Cosine Similarity ($\vec{s}$ , $\vec{\hat{s}}$ );
        if $similarity > threshold$ :
        returns ($\hat{s}$);
```

Algorithm 3.1: Contextual augmentation[4]

The above algorithm was utilized to increase the siSwati and isiZulu datasets. The same process was adopted to solve the class imbalance problem, that is, the minor classes

© University of Pretoria

were augmented so that they can be increase to the size of the majority classes.

3.3.2 SMOTE

SMOTE is an oversampling technique used to rebalance the original training set through the creation of synthetic samples of the minority class[42]. This technique works by selecting the minority class and the total amount of oversampling to balance the classes, then the k-nearest neighbours for that particular class are obtained , therefore, iteratively the k nearest neighbours are randomly chosen to create new instances[42].This oversampling technique was used to balance the classes.

3.4 Data Split

Machine learning algorithms solve the problem without the use of a fixed algorithm but instead they learn the pattern from the data [62]. However, for the models to learn from the data, they must be trained and tested. The training and testing processes play an important role in developing a good machine learning model and the training and testing is done using a separate dataset. The success of the machine learning is highly dependent on the amount of training data, if the features are highly correlated, the training-testing sets are divided into 50% - 50% ratio, meaning that the one half to the data will be used to train the machine learning model and the other half will be used to test the model, however, the training - testing ratio depends on the data structure and the training set data should not be less than 50% but it can be above 50% [62]. K-fold cross validation is a technique that is used to split the data into training and test sets, this technique split the data into k random subsets, where the other subset will be used to train the model and the other one to assess the model performance, it does that k number of times(iteratively)[63]. Therefore the split that will be applied on isiZulu and siSwati datasets during traing and evaluation is 5-fold cross validate, meaning that each model will be trained and evaluated five time on different training and test datasets and then average results of each performance measure produced from each iteration.

The below Figure 3.1 describes the training and testing process.

© University of Pretoria

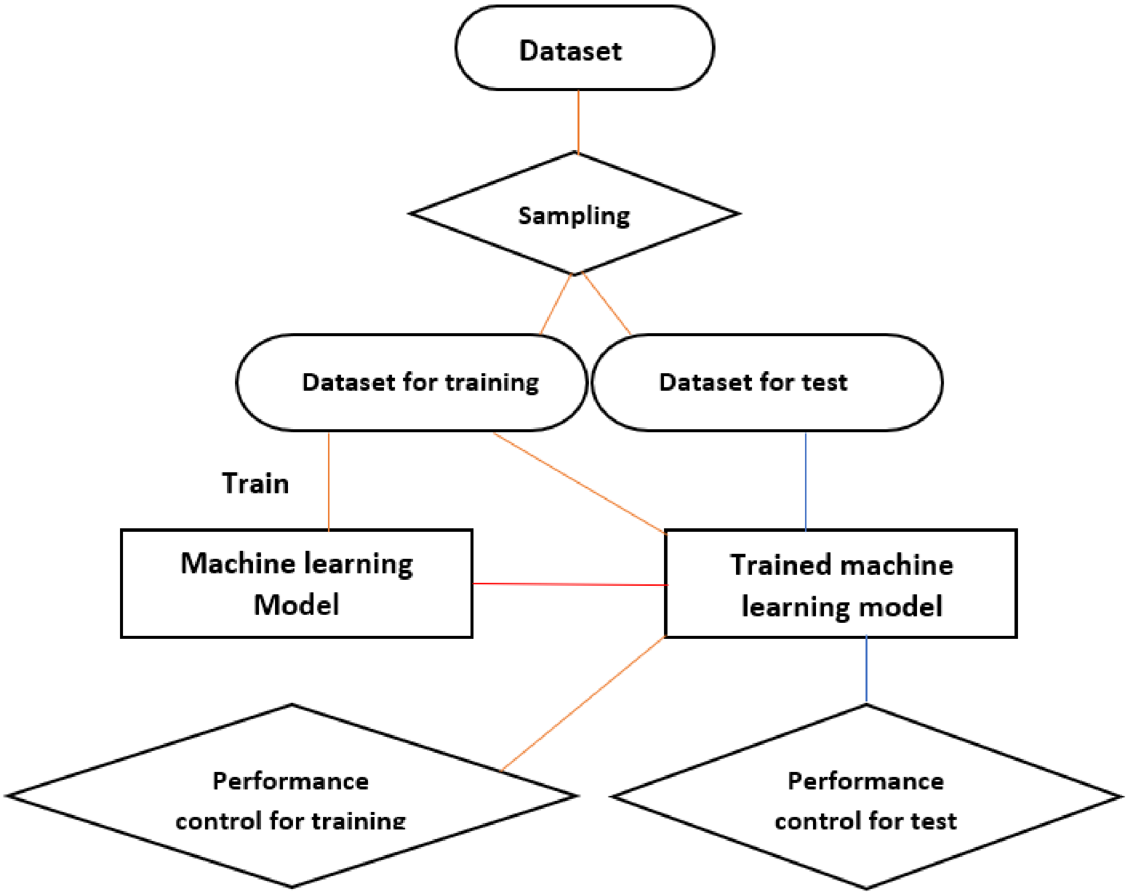

Dataset
Sampling
Dataset for training
Dataset for test
Train
Machine learning Model
Trained machine learning model
Performance control for training
Performance control for test

Figure 3.3: Training and Testing process [62].

3.5 Machine learning Models

Machine learning algorithms have the learning ability to change according to the data pattern and remember previous events, unlike the traditional algorithms that follow specified steps to accomplish the given tasks [62]. The machine learning algorithms are divided into branches, that is, supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transduction, and learning to learn [64]. In this study only supervised and unsupervised learning were utilized.

3.5.1 Unsupervised Learning:

Unsupervised learning algorithms uses the unlabeled data to do the prediction (or clustering), and these algorithms basically extract and group together hidden features and structures that are found in the datasets [62]. Furthermore, the unsupervised learning algorithms learn in the absence of the labeled examples [64]. Below is the unsupervised learning algorithm that is applied on the isiZulu and siSwati datasets to find the clusters.

Topic Modelling

It is common to have high dimensional data in NLP space because of the word embeddings that are created using the text datasets, however, there are techniques that are there in place to reduce the data dimensionality, such as principal component analysis [65]. Principal component analysis is an algorithm that is utilized to perform the reduction from higher dimension to smaller dimension without losing most of the important data information and also segment the uncorrelated important components/features [66]. The isiZulu and siSwati datasets will be independently used to create TFIDF word embedding and PCA method will be applied on the resulting TFIDF matrix to reduce the dimension and produce the principal components, therefore, apply kmeans model and elbow method to extract the top 10 words from the resulting principal components. This is the unsupervised approach to extract out the important features from our isiZulu dataset.

Non-negative matrix factorization (NMF) which is a useful data representation technique that extract hidden patterns and reduce the data dimensionality [67], it does similar task as PCA; therefore, it will be employed to perform the topic extraction for siSwati dataset.

Prior to applying the PCA and NMF on the TFIDF data matrix, the number of clusters had to be identified. Therefore, the unsupervised clustering algorithm called Kmeans clustering algorithm, which determine and group the data points based on the minimum means square error calculated from their center points. Given the data, the kmeans algorithm will iteratively, determine the centroids coordinates, compute the distance of each data point to the centroids, and lastly group the data points based on minimal distance to the centroids. This process split the large datasets into groups

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

based on the commonalities amongst the data points [68]. The kmeans algorithm has high latency when the large dataset is used, then the minibatch kmeans algorithm is used to overcome the latency problem; the minibatch kmeans works like kmeans except that instead of passing the entire datasets as an input to the model, it divides the data into randomly selected batches and then pass each batch as an input to the model[69].

The elbow method is the most popular used method to determine the optimal number of clusters, this method compute the percentage of variation for different number of clusters. The variation changes faster for small number of clusters and slows down when the number of clusters increase leading to a curve that looks like an elbow, the curve can be visualized by plotting variance against number of cluster [70]. Furthermore, Calinski-Harabasz Index which is a measure that explains the compactness of clusters and how well the clusters are spaced; given by

$$CH(K) = \frac{B(K)(N-K)}{W(K)(K-1)} \tag{3.4}$$

where

$$B(K) = \sum_{k=1}^{K} a_k ||(\bar{x_k}) - \bar{x}||^2 \tag{3.5}$$

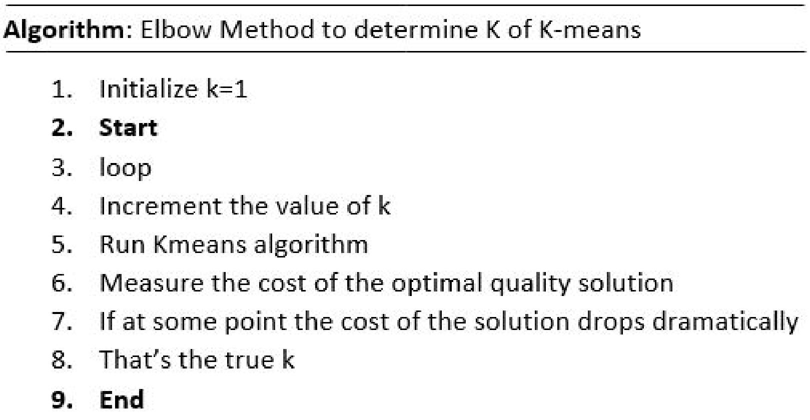
and

$$W(K) = \sum_{k=1}^{K} \sum_{c_j=k} ||x_j - (\bar{x_k})||^2 \tag{3.6}$$

[71] ,the interpretation and variables of the formula are explained next:
The variable K represents the number of clusters, B(K) is the inter-cluster covariance which explains the degree of dispersion between the clusters(the larger the B(K) value, the higher the dispersion) and W(K) is the intra cluster covariance which explains the relationship in the clusters (the smaller the W(K) value the closer the relationship), therefore, the higher the value of CH(K), the better the clustering effect is[71]. The evaluated clusters for Siswati were created using k-means algorithm [70] , hence the elbow model (shown in algorithm 3.2) and Calinski-Harabasz was applied on clusters created by K-means algorithm(shown in algorithm 3.5. And for isiZulu Articles titles and content(news), the minibatch algorithm(shown in algorithm 3.3 was used to create clusters that will also be evaluated using the same criteria.

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA



Algorithm: Elbow Method to determine K of K-means

```
1. Initialize k=1
2. Start
3. loop
4. Increment the value of k
5. Run Kmeans algorithm
6. Measure the cost of the optimal quality solution
7. If at some point the cost of the solution drops dramatically
8. That's the true k
9. End
```

Algorithm 3.2: Elbow Method Algorithm [70] .

3.5.2 Supervised Learning

Supervised learning algorithms create a function that maps the inputs to an outputs based on the training input-output example [64].These supervised learning algorithms get trained by the datasets , they are divided into four, namely, classification algorithms, deep learning, deep transfer learning, and regression methods [62].However, only classification and deep learning will be utilized.

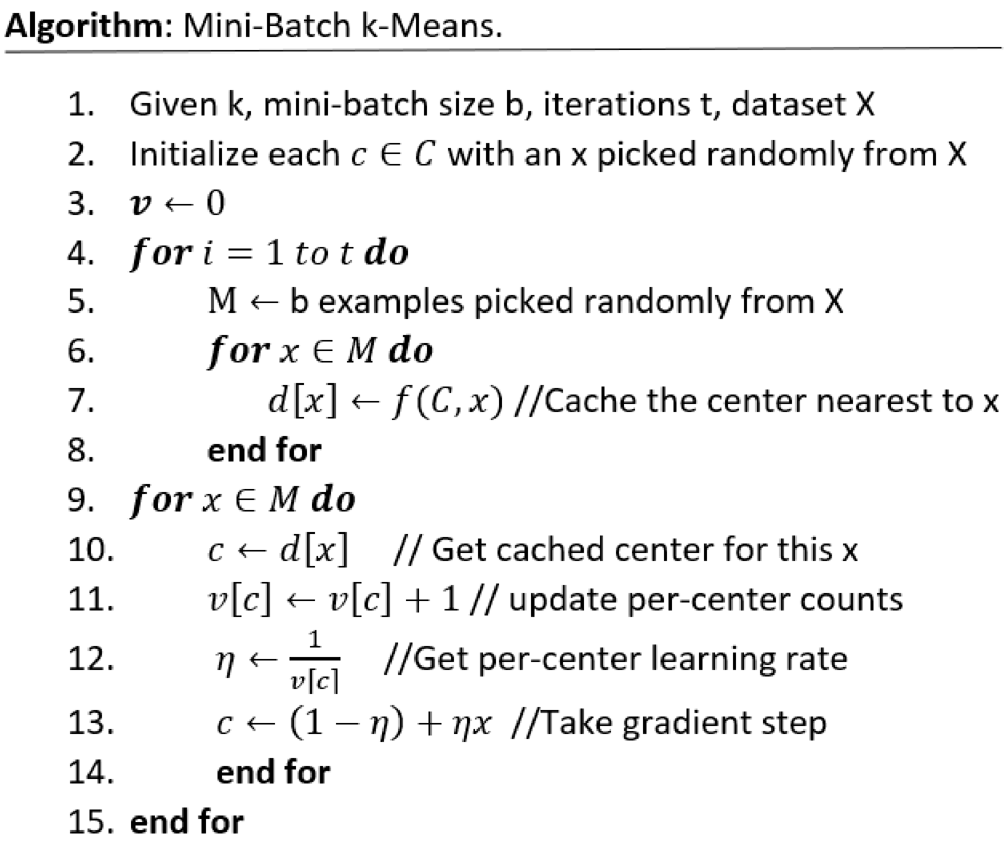
Classification algorithms

Classification algorithms focus on classifying the data into desired output labels, for example, the labels could be Boy or Girl, Cat or Dog, e.t.c. There are many classification algorithms but then one algorithm has to be chosen based on its effectiveness to perform the classification for suitable problem [62]. In this study, the below are the selected classification algorithms for our problem.

Logistic Regression

Logistic regression model is a statistical model that is used to solve classification problems, this model is like linear regression model except that the logistic regression model outputs are binary [73]. Logistic regression model predicts whether the given input belongs to a particular category based on the probability, that is, the prediction is made based on the category that yield the highest probability. For example, if the

© University of Pretoria



Algorithm: Mini-Batch k-Means.

```
Given k, mini-batch size b, iterations t, dataset X
Initialize each $c \in C$ with an x picked randomly from X
$v \leftarrow 0$
for $i = 1$ to $t$ do
    M ← b examples picked randomly from X
    for $x \in M$ do
        $d[x] \leftarrow f(C, x)$ //Cache the center nearest to x
    end for
for $x \in M$ do
    $c \leftarrow d[x]$    // Get cached center for this x
    $v[c] \leftarrow v[c] + 1$ // update per-center counts
    $\eta \leftarrow \frac{1}{v[c]}$    //Get per-center learning rate
    $c \leftarrow (1 - \eta) + \eta x$  //Take gradient step
    end for
end for
```

Algorithm 3.3: Mini Batch Kmeans Algorithm[69]

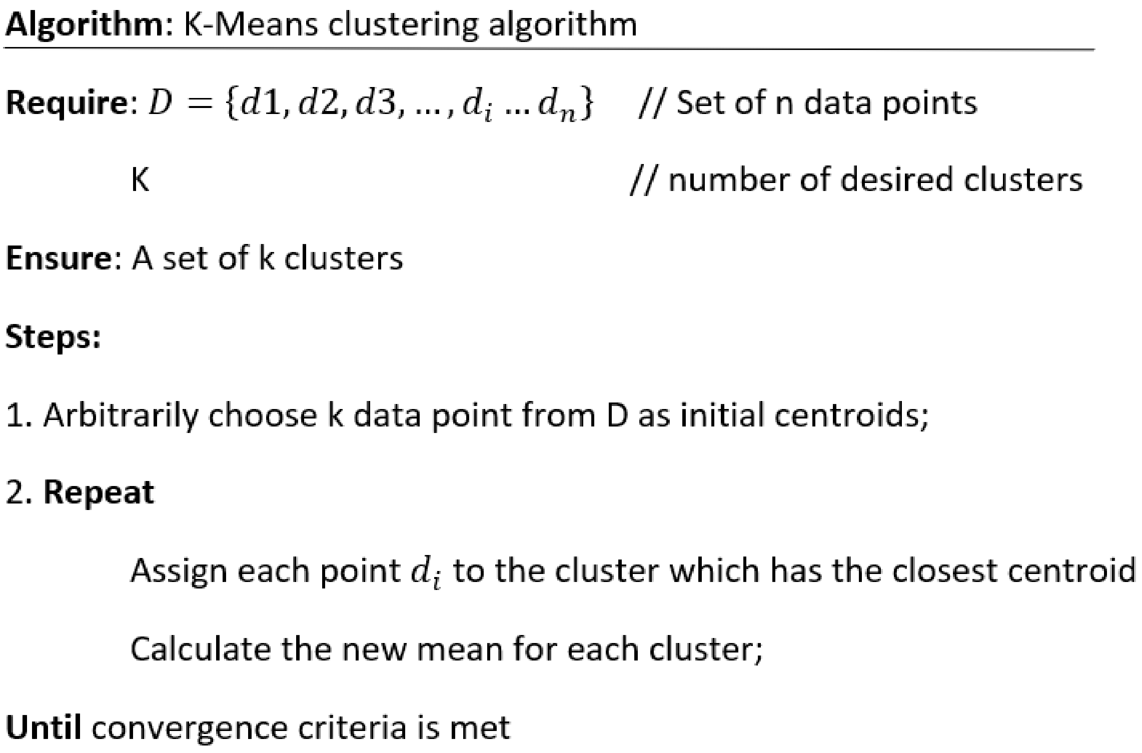
categories are Yes (1) or No (0) and the input is X, then the probability will be expressed as Pr(yes|X) which reads as the probability of category 'yes' given input 'X'[74]. Logistic regression is used to model the relationship between the input and responses, mathematically, logistic regression is given by

$$Pr(X) = Pr(Y = 1|X) \tag{3.7}$$

, in this case, the relationship between response Y and independent variables X are being modeled. Logistic regression is derived from linear regression which is given by

$$Pr(X) = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n, \tag{3.8}$$

then equating these probabilities 3.7 and 3.8,



```
Algorithm: K-Means clustering algorithm
Require: $D = \{d1, d2, d3, \ldots, d_i \ldots d_n\}$     // Set of n data points
         K                                               // number of desired clusters
Ensure: A set of k clusters
Steps:
1. Arbitrarily choose k data point from D as initial centroids;
2. Repeat
         Assign each point $d_i$ to the cluster which has the closest centroid
         Calculate the new mean for each cluster;
Until convergence criteria is met
```

Algorithm 3.4: Kmeans algorithm [72]

$$Pr(X) = Pr(Y = 1|X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n \tag{3.9}$$

[74] In classification model, the odds are used instead of the probability (used in regression model). Odds are defined as the probability that the event will happen over the probability that the event will not occur, and they are given by the formula

$$odds = \frac{p}{(1 - P)} \tag{3.10}$$

, where p is the probability of success. Therefore, the odds are substituted in place of the probability in equation (3), that is,

$$odds = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n \tag{3.11}$$

$$ln(\frac{p}{(1 - P))} = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n \tag{3.12}$$

© University of Pretoria

$$\frac{p}{(1-P)} = \exp\left(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n\right) \tag{3.13}$$

after applying Taylor series and making P the subject of the formula,

$$P = \frac{\exp\left(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n\right)}{\exp\left(1 + \exp\left(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n\right)\right)} = \frac{1}{\left(1 + \exp\left(-(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n)\right)\right)} \tag{3.14}$$

Therefore this sigmoid function formula will produce the probability scaled between 0 and 1 [75] Where P is response or dependent variable, X is the independent variable,β_0 is the gradient of X and $\beta_{\geq 1}$ are the change of Y with respect to X [73]. The logistic regression model is known to predict binary classes; however, it can be made to take multi class and such model that takes more than two classes is called multinomial logistic regression. In this case, we would want to predict the target variable y which constitute of more than two classes, given x. Mathematically it is expressed as p(y=c|x), the Softmax function is used to compute the probability of p(y=c|x). Softmax is the generalization of sigmoid function so it takes a vector $x = [x_1, x_2, x_3, , x_n]$ and fit them in a probability distribution where as a results they range between 0 and 1 , and all values in vector x sum up to 1. It is expressed as

$$Softmax(z_i) = \frac{\exp(z_i)}{(\sum_{j=1}^{n} \exp(z_i))}, 1 \leq i \leq n$$

(3.15)

When we input the vector x,

$$Softmax(x) = \frac{\exp(z_1)}{(\sum_{j=1}^{n} \exp(z_i))}, \frac{\exp\left(z_2\right)}{(\sum_{(j=1)}^{n} \exp\left(z_i\right))}, , \frac{\exp\left(z_k\right)}{(\sum_{(j=1)}^{n} \exp\left(z_i\right))}.$$

(3.16)

$$where z_1 = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n \tag{3.17}$$

© University of Pretoria

Therefore, the multinominal regression uses the generalization of sigmoid function which is called softmax [76]. The logistic regression model was fitted and used to perform the classification on isiZulu and siSwati news dataset, based on the above explained algorithm.

Naïve Bayes

Multinomial naïve bayes classifier is a probabilistic classifier that make uses of Bayesian probability and naïve assumptions about how the features interact. Naïve bayes classifier is applied in natural Language Processing problems to perform the classification, for instance, given document that may belong to a category $c \in C$ (possible categories) then the classifier will return c^ category with the highest posterior probability. Note that the notation represent the estimated category, which is mathematically represented as

$$\hat{c} = \underset{c \in C}{argmax}\, p(c|d) \tag{3.18}$$

[76] Naïve bayes is derived from Bayesian probabilities, given by

$$p(y|x) = \frac{p(x|y)p(x))}{(p(y)} \tag{3.19}$$

therefore, the equation 3.18 can be substituted into equation 3.19 ,

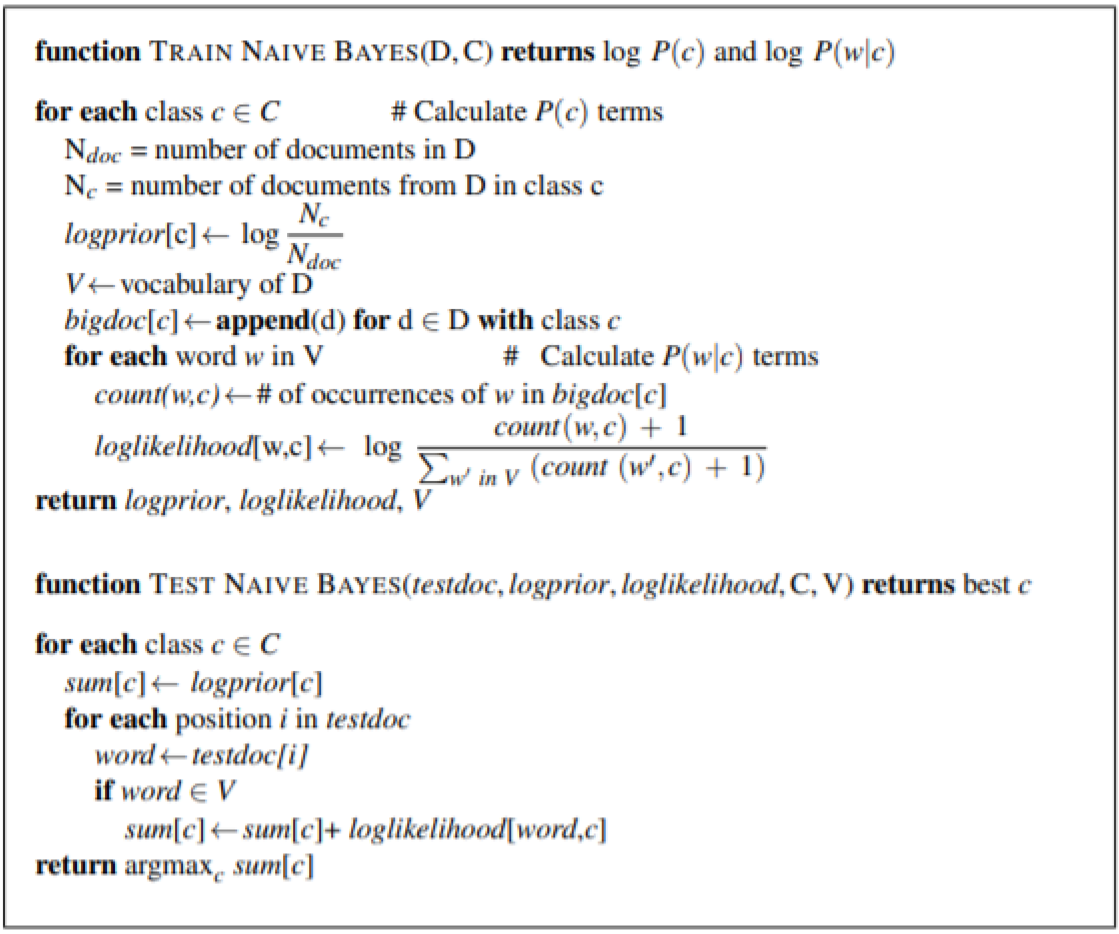
$$c = \underset{c \in C}{argmax}\, p(c|d) = \underset{c \in C}{argmax}\, \frac{(p(d|c)p(c))}{p(d)}. \tag{3.20}$$

Now this equation needs to be maximized, then the denominator p(d) can be dropped since it is the same across all category, so the equation to be maximized is

$$c = \underset{c \in C}{argmax}\, p(c|d) = \underset{c \in C}{argmax}\, p(d|c)p(c). \tag{3.21}$$

The part of the formula p(d|c) represents the likelihood of data given the class and p(c) represents the prior probability of the class c, so the probable category $\hat{c}$ is the category with the highest product of likelihood and prior probability. In simpler terms, the estimated category will be the one with the highest probability [76].

© University of Pretoria



```
function TRAIN NAIVE BAYES(D, C) returns log $P(c)$ and log $P(w|c)$

for each class $c \in C$            # Calculate $P(c)$ terms
   $N_{doc}$ = number of documents in D
   $N_c$ = number of documents from D in class c
   $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
   $V \leftarrow$ vocabulary of D
   $bigdoc[c] \leftarrow$ append(d) for d $\in$ D with class $c$
   for each word $w$ in V                  #  Calculate $P(w|c)$ terms
      $count(w,c) \leftarrow$ # of occurrences of $w$ in $bigdoc[c]$
      $loglikelihood[w,c] \leftarrow \log \frac{count(w,c)\ +\ 1}{\sum_{w'\ in\ V}\ (count\ (w',c)\ +\ 1)}$
return $logprior$, $loglikelihood$, $V$

function TEST NAIVE BAYES($testdoc$, $logprior$, $loglikelihood$, C, V) returns best $c$

for each class $c \in C$
   $sum[c] \leftarrow logprior[c]$
   for each position $i$ in $testdoc$
      $word \leftarrow testdoc[i]$
      if $word \in V$
         $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$
```

Algorithm 3.5: Naïve bayes algorithm [76]

The naïve bayes model was fitted and used to perform the classification on isiZulu and siSwati news dataset, based on the above explained algorithm.

XGBOOST

XGBoost is a machine learning algorithm that makes use of boosted trees to predict target variable y given an input x. The XGBoost algorithm make use of two objective function, training loss and regularization functions, to find the best parameters for train data and measure how well the model is performing. The objective function is given by

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{3.22}$$

, where L is the training loss function that measures how predictive the model is and Ω is the regularization function which prevents the model from overfitting[77]. Consider

© University of Pretoria

an ensembled tree that consists of classification and regression trees (CART), therefore, after formulating the trees then each leave on a tree will have a score, moreover, each leaf will have a score which they will be summed up to get a final score that will be used to make a decision around the prediction. Now the predicted target value is given as

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{3.23}$$

Where K is the number of trees, f is a function in F, and F is the set of all possible Carts. Objective function that must be optimized is given by

$$obj(\theta) = \sum_{(i)}^{n}(l(y_i, \hat{y}_i^t)) + \sum_{i}^{t} w(f_k) \tag{3.24}$$

Applying additive training, we express the prediction equation at t step as

$$\hat{y}_i^0 = 0 \tag{3.25}$$

$$\hat{y}_i^1 = (y_i^0) + f_1(x_i) \tag{3.26}$$

$$\hat{y}_i^2 = (y_i^1) + f_2(x_i)... \tag{3.27}$$

$$\hat{y}_i^t = \sum_{(k=1)}^{t} f_k(x_i) = (y_i^{(t-1)}) + f_t(x_i) \tag{3.28}$$

$$obj^t = \sum_{i}^{n} l(y_i, (y_i^{(t-1)}) + \sum_{i}^{t} w(f_i) = \sum_{i}^{n} l(y_i, (y_i^{(t-1)}) + f_t(x_i)) + \sum_{i}^{t} w(f_i) + constant. \tag{3.29}$$

$$obj^t = \sum_{i}^{t}(y_i - ((y_i^{(t-1)}) + f_t(x_i)))^2 + \sum_{i}^{t} w(f_i) + constant. \tag{3.30}$$

$$= \sum_{i}^{t}[2(((y_i^{(t-1)}) - y_i)f_t(x_i)) + (f_t(x_i))^2] + \sum_{i}^{t} w(f_i) + constant \tag{3.31}$$

Now we take the taylor form,

$$obj^t = \sum_{i}^{n}[l(y_i, y_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \sum_{i}^{t} w(f_i) + constant \tag{3.32}$$

© University of Pretoria

Where

$$g_i = \delta_{(y_i^{(t-1)})} l(y_i, (y_i^{(t-1)}))and \tag{3.33}$$

$$h_i = \delta_{(y_i^{(t-1)})}{}^{2}\, l(y_i, y_i^{(t-1)}) \tag{3.34}$$

, then after removing the constants, the objective function at step t is

$$obj^t = \sum_{(i=1)}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \sum_{i}^{t} w(f_i). \tag{3.35}$$

The training loss function has been proven and now we need to find the second term of the objective function which is the regularization term. The model complexity can be defined as follows, given a tree f(x),

$$f_t(x) = w_{(q(x))}, w \in R^t, q : R^d \hookrightarrow 1, 2, 3, T \tag{3.36}$$

where w is the vector score on each leaf, q is the function that maps the data point to the corresponding leaves, and T is the number of leaves. Therefore, the complexity is given by

$$w(f) = \gamma T + \frac{1}{2} \lambda \sum_{(j=1)}^{T} w_j^2. \tag{3.37}$$

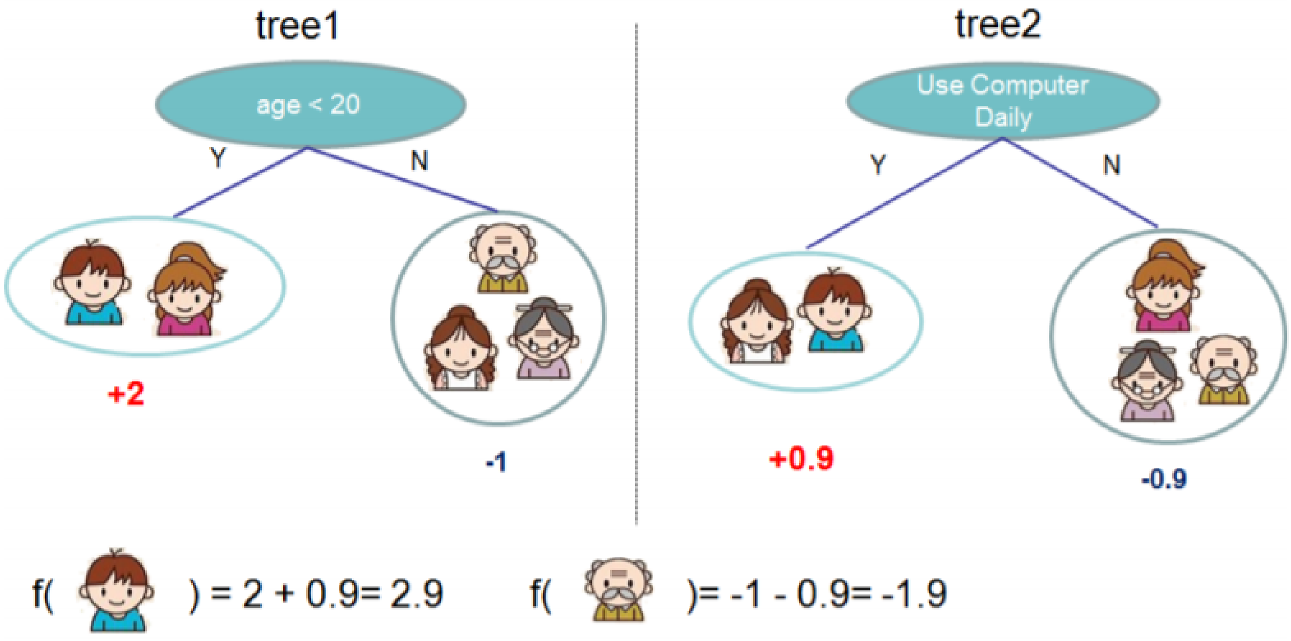
The two terms of the objective function have been derived and the full form of the objective function is

$$obj^t \approx \sum_{(i=1)}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2} \lambda \sum_{(j=1)}^{T} w_j^2 \tag{3.38}$$

$$= \sum_{(j=1)}^{T} [(\sum_{(i \in I_j)} g_i) w_j + \frac{1}{2} (\sum_{(i \in I_j)} h_i + \lambda) w_j^2] + \gamma T \tag{3.39}$$

Where $I_j = \{i | q(x_i) = j\}$ is the set containing the indices of data points assigned to the j^{th} leaf and the w_j is the score on the leaf and they are not dependent of each other. This equation checks the structure of the tree and output how good the structure is, the smaller the score, the better the structure of the tree[77].

© University of Pretoria



Algorithm 3.6: Ensembled tree of two trees[77].

Deep learning

Deep learning is a machine learning subfield that imitates the human brains learning technique to learn from the data using the artificial neural networks [78] and it is different from the traditional machine learning algorithms from the data representation perspective since it represents it in nonlinear form [79]. The models of this family that will be utilized to perform the classification is explained below.

Long Short-Term Memory (LSTM)

Neural networks are composed of a network of small processing units (or nodes) connected by the weighted connections. The neural networks were developed imitating the human brains where the nodes are neurons and the connection weights between the neurons are the strength of the synapses. Once the input has been fed into the networks then the network gets activated while the data flow from one node to another spreading through the weighted connections. There are varieties of neural network, the variation is based on the connection shapes, some form acyclic shape which are referred to as feedforward neural network, and on the other hand, the ones that form a cycle shape are

© University of Pretoria

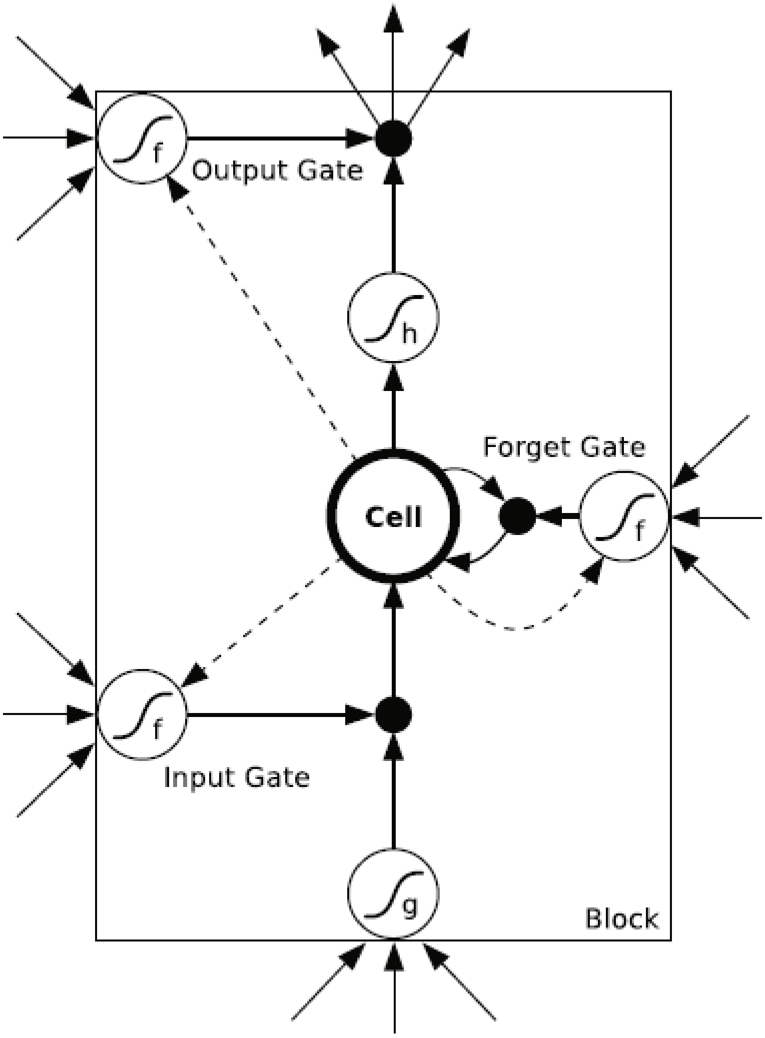
referred to as recurrent neural network [80].

The recurrent neural network (RNN) differ from feedforward neural network (FFNN) by input and output mapping, the RRN maps all the previous inputs history to each output which gives it the capability to keep memory of the previous inputs in the network's internal state, whereas the FFNN only maps input to output vectors. Furthermore, RNN has the drawback of vanishing gradients that happens when the given input cycles around the recurrent networks [80].

Long short-term memory (LSTM) is a neural network that uses the recurrent neural network (RNN) architecture to temporally model the sequences of data and their long-range relationships[81]. The LSTM is composed of recurrently connected subnets knows as memory blocks; and each memory block have at least one self-connected memory cells and the multiplicative gates that allows the LSTM to remember long time processed information and therefore, that solves the problem of vanishing gradient. The LSTM is like the RNN except that memory blocks are used on LSTM instead of the summation units on the hidden layers [80].

The memory block with one cell will have three gates, namely, output gate, input gate and forget gate as shown in Figure 3.7. These gates collect the activation from within and outside the block and activates the cell through multiplication outcomes. The input and output gates apply the multiplication on the cell's input and output and forget gate takes care of the multiplication of the cell's previous state. Moreover, the gates are usually activated using sigmoid activation function [80].

© University of Pretoria



Algorithm 3.7: LSTM memory block [80].

The above model will be applied on isizulu and siSwati datasets to perform the classification and the below section describes how all the models will be evaluated in order to obtain the best performing model.

© University of Pretoria

3.6 Model Evaluation

The machine learning models are developed and the techniques to assess the effectiveness and performance of the machine learning models is called model evaluation. There are many ways that are used to assess the quality of models[82]; however, the confusion matrix and F-score defined and explained below will be utilized since we would be assessing the classification models.

3.6.1 Confusion matrix and F1-score

The confusion matrix is used to measure the quality and effectiveness of the machine learning models and usually utilized to assess the performance of classification models [82]. Confusion matrix is represented in a tabular version of size n x n where n represents the number of classes to be predicted, for instance, confusion matrix of n=2(classes) shown in table 4.2, where

- a is the number of correct negative predictions.
- b is the number of incorrect positive predictions.
- c is the number of incorrect negative predictions.
- d is the number of correct positive predictions.

The confusion matrix produces the prediction accuracy, given by,

$$accuracy = \frac{(a+d)}{(a+b+c+d)} \tag{3.40}$$

[83].

Table 3.1: 2x2 confusion matrix [83].

	Predicted Negative	**Predicted Positive**
Actual Negative	a	b
Actual Positive	c	d

© University of Pretoria

The classification accuracy is a one value ranging from 0(bad prediction) to 1(good prediction) that represents the ratio of the correctly predicted values against the entire dataset. The bigger the accuracy value (i.e close or equals to 1) means the model is performing well else the model is under performing[31]. In some cases, when the data is imbalanced, the accuracy measure may produce wrong classification results, however, the F1-score measure which is also derived from confusion matrix does not get affected by the imbalance data. The other measures that could be derived from the confusion matrix are precision and sensitivity which are defined as follows: Precision which calculates the ratio of correctly predicted positive labels against all predicted positive labels, given by the formula

$$precision = \frac{d}{(b+d)} \tag{3.41}$$

and on the other hand, sensitivity calculates the ratio of the correctly predicted positive labels over all actual positive labels, given by

$$sensitivity = \frac{d}{(c+d)} \tag{3.42}$$

. Therefore, F1-score is defined as the harmonic mean of precision and sensitivity, mathematically represented as,

$$F1score = 2\frac{(precision.recall)}{(precision + recall)} \tag{3.43}$$

, and it is interpreted the same way as the accuracy , that is, it ranges between 0 and 1; and the closer (or equals) the value to 1 the good the model or the smaller the value, the bad the model[31]. The two performance measures, accuracy and f1 score were used to assess the quality of our models.

3.6.2 LIME Model

Local interpretable Model-Agnostic Explanations (LIME) is a popular algorithm that shows the decision-making process of the black box machine learning models [84] [85]. LIME models derive the explanation by sampling the instances and provide the prediction of each instance using the classifier function then weighs them to the neighboring explained instance. Therefore, the local interpretation of the black box model will be

© University of Pretoria

provided together with the influential factors that led to the model decision [84]. This algorithm was utilized to explain all the four black box models that are trained in this work to get the insight of the decision-making process.

3.7 Summary

The detailed technical part from the data cleaning, Data Augmentation up until the model evaluation were explained to provide a background understanding of each model, algorithm and processes that were employed to deliver the objective of this work. This chapter provided the technical aspect of each tool that was used and the non-technical details will be provided in the next chapter.

© University of Pretoria

Chapter 4

Applied Methodology

This chapter describes all the steps / methods that were employed in this study to accomplish our goal, namely:

- Data collection and annotation
- Data preparation
- Data analysis
- Text classification

The above-mentioned tasks were executed to achieve our main goal which was corpus creation, annotation and text classification for South African low-resourced languages. Section 4.1 covers the data collection process and the data sources, Section 4.2 covers the data annotation process that was taken to label the datasets, Section 4.3 covers the data preparation processes that was done on the data before putting it in the model, Section 4.4 covers the word embeddings that were created in this work, Section 4.6 provides the unsupervised and supervised models that were built in this work together with the results obtained from unsupervised mini experiment, section 4.7 provides an understanding of the model evaluation measures that we used to select the best model, and lastly, section 4.8 summarises the whole chapter and highlight on what to expect in the next chapter.

© University of Pretoria

4.1 Data collection

The isiZulu news data was collected from Isolezwe, which is a Zulu-language local newspaper. The news article data published online on Isolezwe website was scraped and stored in a csv file for further processing. In turn, the siSwati dataset was collected from public broadcaster for South Africa, that is, SABC news LigwalagwalaFM Facebook page and the scraped data was the news headline posts posted online and it was also scraped online and stored on a csv file. Lastly, the other isiZulu and siSwati datasets were collected from Sadilar(www.sadilar.org) and Leipzig(https://wortschatz.uni-leipzig.de) for the purpose of vectorizers creation to increase the token variety.

The size of each dataset is as follows:

Table 4.1: Original News Datasets

	isiZulu Full Articles	isiZulu Titles	siSwati Titles
Corpus size	752	752	80
Number of tokens	43023	3495	3418

Table 4.2: Vectorizer Corpora Sizes in number of tokens

	Tokens	
Source	**isiZulu**	**siSwati**
Sadilar	770845	399800
Leipzig	4296659	134827
Total	5067504	534627

4.2 Annotation

The isiZulu and siSwati news article datasets was annotated so that they can be used to perform the classification. The categories that are possibly associated with the news articles are listed below,

- arts, culture, entertainment and media

© University of Pretoria

- conflict, war and peace
- crime, law and justice
- disaster, accident and emergency incident
- economy, business and finance
- education
- environment
- health
- human interest
- labour
- lifestyle and leisure
- politics
- religion and belief
- science and technology
- society
- sport
- weather

Therefore, the datasets were given to the annotators (three linguistic experts) to perform the data annotation. The data was annotated by more than one annotator hence there was an issue where the categories conflicts for some articles, then in that case, the majority voting technique was adopted. That is, the category that appeared most for that article was be considered, and since it was three annotators then there was no tie in terms of conflicting labels.

© University of Pretoria

The original class distrubition for isiZulu(both articles and titles) is shown below in Figure 4.1 and it was observed that the classes are imbalanced, for instance, the class category *'crime, law and justice'* have 292 records whereas class category 'weather' has only 6 records. on the other hand, the class distribution for siSwati dataset is shown in Figure 4.2 and it resembles the same class imbalance as the one observed in isiZulu dataset.

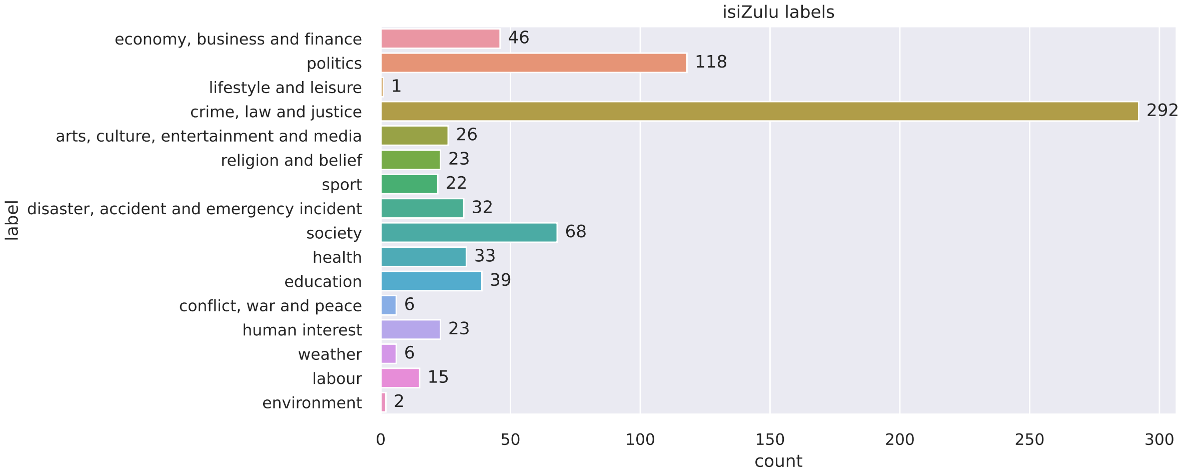


Figure 4.1: isiZulu initial Class Distribution

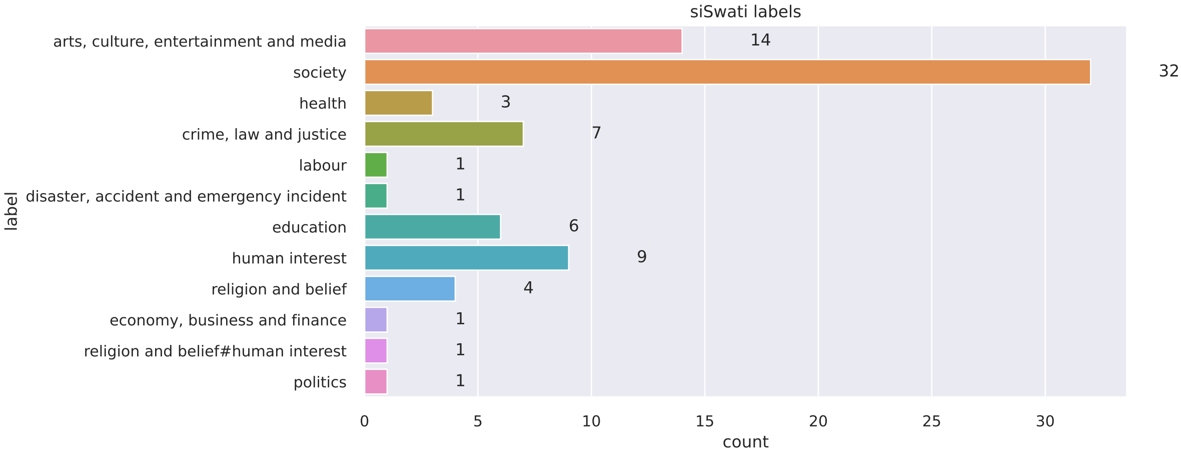


Figure 4.2: siSwati initial Class Distribution

Some class categories have much fewer records than the other and it affects the classification model since the class imbalance introduce the bias in the clasification model.However, only class categories with at least 35 records for isiZulu and at least 6 records for siSwati were selected, the other class categories were discarded,therefore, the remaining class categories for isiZulu were:

- crime, law and justice
- economy, business and finance
- education
- politics
- society

and the 5 major classes for siZwati are:

- crime, law and justice
- arts, culture, entertainment and media
- education
- human interest
- society

since the number of class categories have dropped to 5 categories, the news dataset size also dropped to 563 for isiZulu and 68 for siSwati as shown in the table 4.3 below

Table 4.3: 5 major categories News Data Sets

	isiZulu Full Articles	**isiZulu Titles**	**siSwati Titles**
Corpus size	563	563	68
Number of tokens	29217	2532	2485

© University of Pretoria

4.3 Data preparation

The datasets collected in this work contained some noise such as single characters, white spaces, encoded characters, meaningless words, and special characters. The noise had to be removed before the datasets are fed into the models. All these noises on the datasets were removed with the use of Python code as follows.

- **Single characters**: The single characters carry less meaning, so they were also removed from the datasets using the below line of code

```
    document = re.sub(r'\s+[a-zA-Z]\s+', ' ', document)
```

- **White spaces**: There were instances where there are multiple spaces between two words, so those spaces were substituted with a single space

```
    document = re.sub(r'\s+', ' ', document, flags=re.I)
```

- **Encoded characters**: There were some characters/words that were not ASCII encoded then those characters were decoded back to ASCII using the code:

```
    document = [word.encode('ascii', 'ignore') for word in document]
    document=[word.decode() for word in document]
```

- **Special characters**: Special characters refer characters such

```
    document = re.sub(r'\W', ' ', str(data[x]))
```

- **Meaningless words**: The data contained combination of letters that don't make any existing isiZulu/siSwati word and some of the identified isiZulu and siSwati stopwords hence those words were listed in a list and extend the existing english stopwords list. Therefore, the stopwords(containing nonexistence words) were excluded from the dataset

```
stop = stopwords.words('english')

newStopWords = ['udkt','unksz','unkk','19','ngo','kodwa','uma','be','
    kusho','noma','fighters','zonke','kusho','la','lakhe','mina','
    ngesikhathi','nje','ukuba','u','ukuthi','ukuze','uma','wakhe','
```

© University of Pretoria

```
    wami','wase','wathi','yakhe','unkk','zonke','ngoba','uthe','noma
    ','njengoba','nje','bese','uma','ku','futsi','utsi','kwekutsi','
    kutsi','video','sabcnews','ingabe','lelive','kulelive','whatsapp
    ','sabctindzaba','njani','nge','natsi','wakho','ne','na','wena','
    sabc','nga','live']

stop.extend(newStopWords)
document = [word for word in document if not word.lower() in stop]
```

Once the datasets are noise free, each letter in the datasets was set to lowercase, resulting in clean datasets to be used in word embedding creation and machine learning models building.

4.4 Word embeddings

The word embeddings for isiZulu and siSwati were independently created using the two datasets for each respective language, that is, for isiZulu the two datasets were used: isiZulu news dataset, Sadilar and Leipzig isiZulu datasets; on the other hand, for siSwati: siSwati news dataset, siSwati Sadilar and Leipzig datasets. The two datasets for each language were combined to create one dataset to be used to create word embeddings.

The three word-embeddings, namely, TFIDF, bag of words and Word2vec were created with the use of the combined Sadilar and leipzig datasets for each language. The combined datasets were fed into the three word embedding models to create three different vectorizers.

The default parameters were used for the creation of 1-gram bag of words except for the vocabulary since it was provided. Whereas for 1-gram TFIDF, everything remained as default except for the vocabulary; and the max_df and min_df threshold were set to 65 and 5 respectively, to eliminate the words that occurred most and less frequently on the datasets, this works as a mechanism to remove stopwords. Lastly, the default parameters for Word2vec were kept unchanged in the process of creating a vectorizer.

The three word-embedding were created to be used on the classification models, however, the bag of words a TFIDF were used on the following classical models

- Logistic regression model

© University of Pretoria

- Naïve Bayes model
- Xgboost model

And Word2vec was used on all the models, including LSTM model.

4.5 Data Generation

The isiZulu and siSwati datasets consist of only 5 class categories, however, the data was still imbalanced,hence we applied the sampling techniques to balance the class categories and then feed the data into the models. The two techniques, namely, Data Augmentation and SMOTE were employed to solve the class imbalance problem and as a results, the class distribution after applying Data Augmentation is shown below in Figure 4.3. The aim was to bring all the minor class categories close to the major category(oversampling) which is *'crime,law and justice'*, hence this class category was not augmented.Moreover, it was observed that the count of class categories are in the same range, that is, within 200-300.The isiZulu Titles class distribution is shown in Figure 4.4, however, it was observed that the class categories *'education' and 'economy, business and finance* still have records less than 200 due to lack of unique word since titles contains short texts. In the case of siSwati, the oversampling process was performed with the aim to equal the records of minor classes to the major class which is class category *society*. The class distribution post Data Augmentation for siSwati is shown in Figure 4.5, the classes are equal.

© University of Pretoria

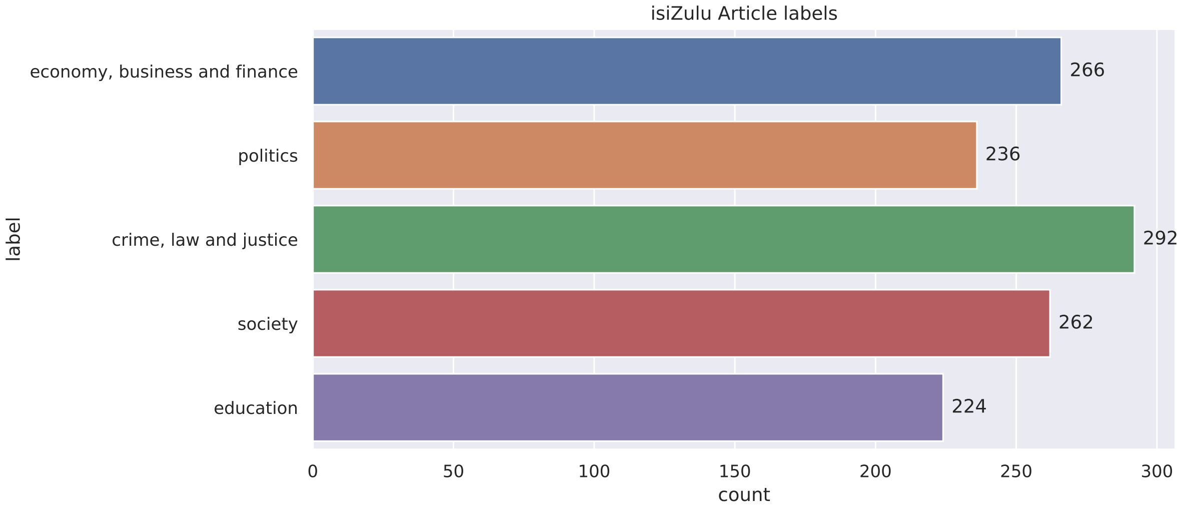


Figure 4.3: isiZulu Post Data Augmentation Class Distribution

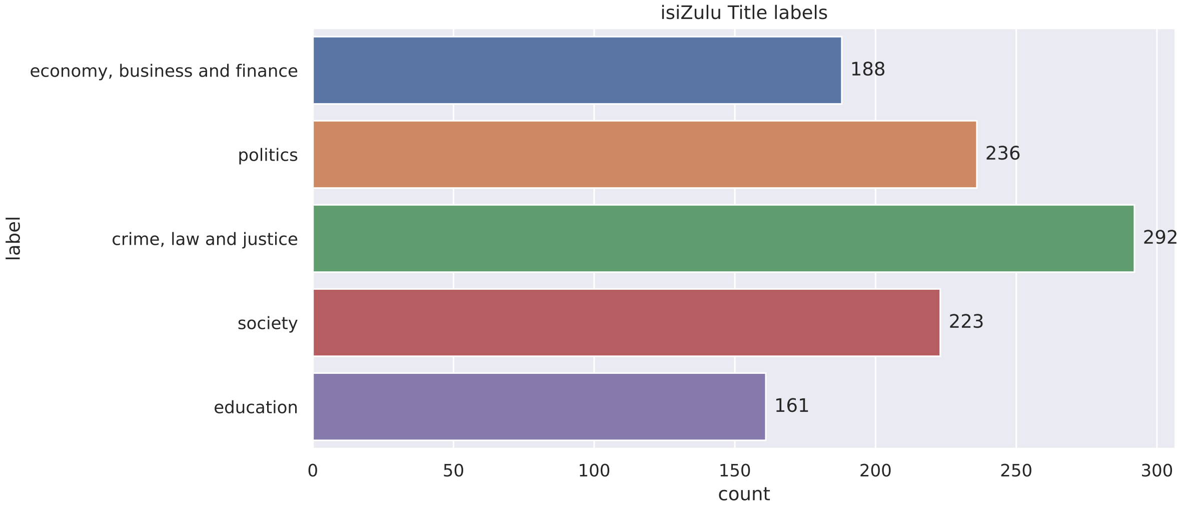


Figure 4.4: isiZulu Titles Post Data Augmentation Class Distribution

© University of Pretoria

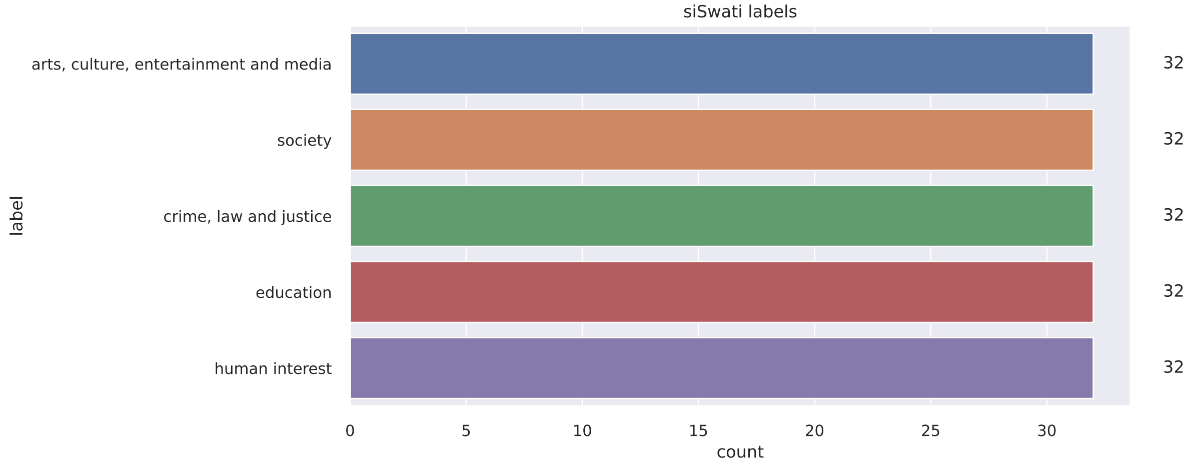


Figure 4.5: siSwati Post Data Augmentation Class Distribution

The same 5 class categories were balanced using SMOTE technique, the class categories were made to be equal. Before applying SMOTE the dataset class distribution were imbalanced as shown in Figures 4.1 and 4.2 for isiZulu news dataset(both article and title) and siSwati news dataset respectively, and thereafter applying SMOTE the class categories were balanced as shown in the below Figures 4.6 and 4.7.

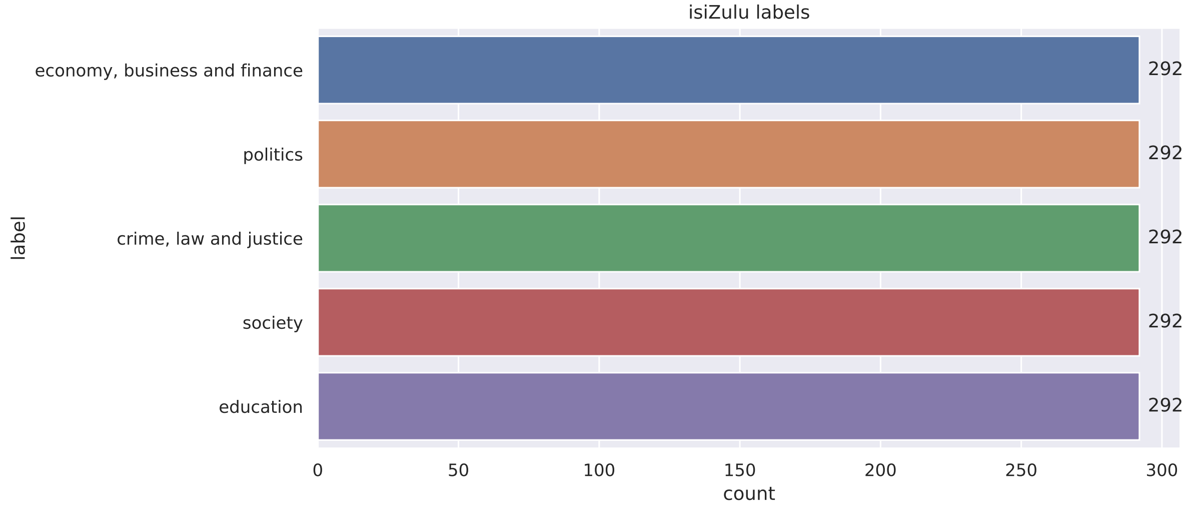


Figure 4.6: isiZulu Articles & Titles Post SMOTE Class Distribution

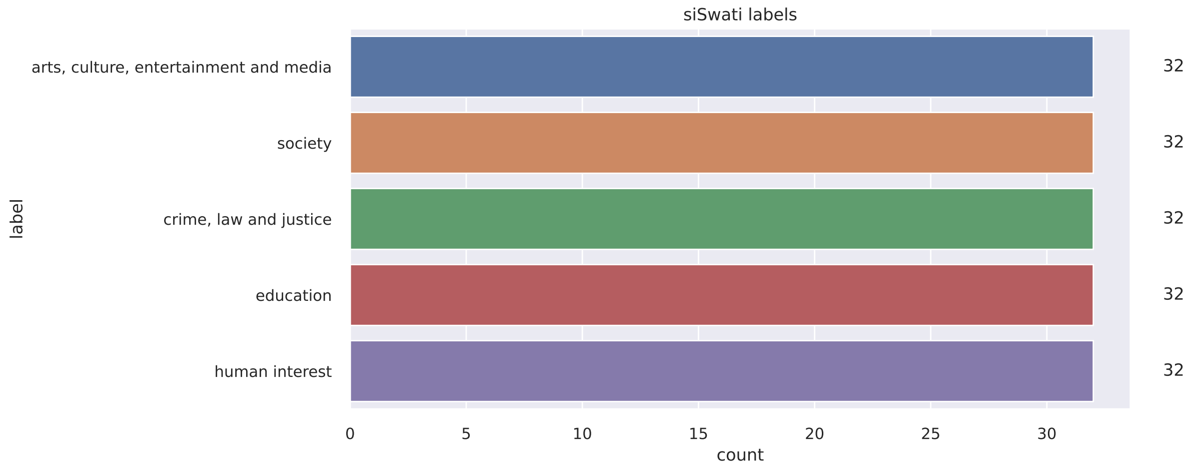


Figure 4.7: siSwati Post SMOTE Class Distribution

The models were trained using the balanced datasets and thereafter, the performances were recorded for analysis purposes.

4.6 Modelling

This section covers the unsupervised and supervised models that were implemented on isiZulu and siSwati dataset to perform clustering and classification.

4.6.1 Unsupervised

The unsupervised machine learning algorithms were applied on the isiZulu and siSwati clean datasets to perform the topic modelling. Below are the models applied on the datasets and the results.

PCA

Topic modelling for isiZulu news articles was performed using the PCA model. Prior to fitting the PCA model, the minibatch kmeans algoruthm and elbow method were utilized to determine the number of clusters(k). The clusters number was decided using the elbow graph that was produced through the number of clusters against the Calinski_Harabasz score, the optimal number of cluster was found to be four as shown in Figure 4.8. Therefore, the PCA model was fitted to produce three topics. Hence, the topics obtained from the PCA model are listed below.

isiZulu News Article Topics

The generated topics can be categorised, for instance, topic 0 and topic 3 are more associated to the politics, whereas topic 1 and topic 2 are more associated to education since they speak of school and students.The PCA model is able to extract meaningful topics that can be categorized. The model performed very well since we managed to categorise all the topics.

- Topic 0: anc umnuz we natal kwazulu umengameli izolo ramaphosa cyril be
- Topic 1: kuthiwa okuthiwa abantu lo esikoleni ngenxa imali umndeni abafundi emuva
- Topic 2: izitshudeni abasebenzi ungqongqoshe inational sikhwama izolo africa sezimali ethekwini million
- Topic 3: umnuz zuma jacob wezwe uzuma kombuso yidlanzana kwikhomishini ephenya zondo

© University of Pretoria

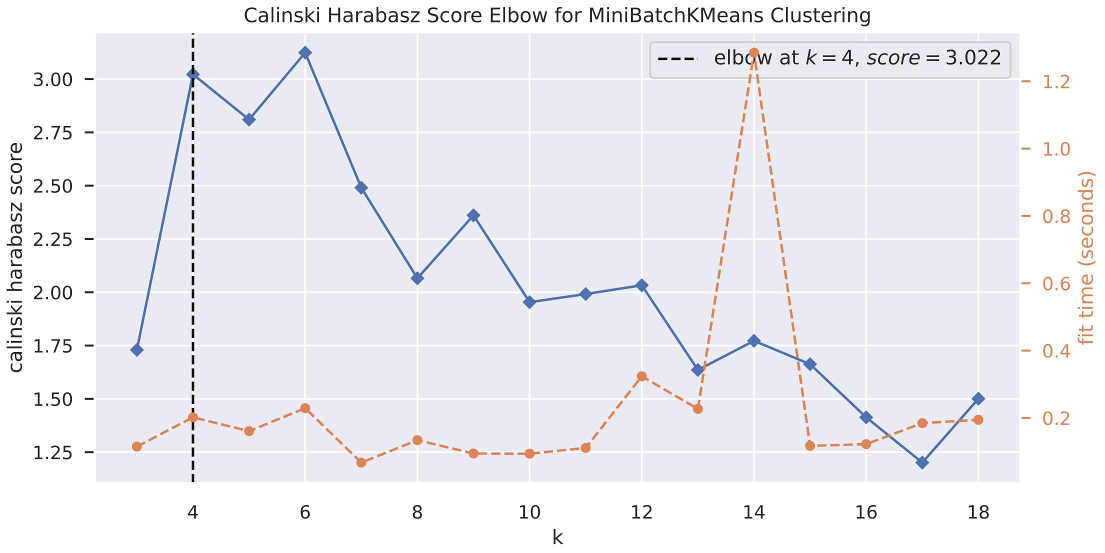


Figure 4.8: MiniBatch Kmeans Clustering for isiZulu Articles

NMF

Topic modelling for siSwati and isiZulu news titles was performed using the NMF since the data is small(contains short text). The elbow method together with Calinski_Harabasz(measure) score were used to decide on the number of clusters, however, the Kmeans algorithm was utilized instead of minibatch kmeans. The datasets were vectorised and fed into the NMF algorithm to extract the topics. The number of clusters obtained from the elbow method is three for both isiZulu and siSwati news titles as shwon in Figure 4.1 and 4.2 respectively. The topics equivalent to the number of clusters obtained from the elbow method were extracted out using the NMF method. The topics extracted from the NMF algorithm for isiZulu news title and siSwati are listed in the sub-sections below.

isiZulu News Title Topics

PCA and NMF models were utilised to group the isiZulu and siSwati news datasets into topics based on the similarities shared across the texts. below we are going to examine whether the models were able to extract topics accurately; that was done through as-

© University of Pretoria

sessing whether the top words in each topic are related and can be matched to one of the existing/pre-defined topics.

The topics generated can be associated with the class categories that we already know, for instance, topic 0 speaks about the ”ANC, arresting and Ramaphosa” hence it can be said to fall under class ’politics’ or ’crime category’.Topic 1 also speaks about the politics and crime, whereas topic 2 is not clear enough to be categorized. The model performed better since we managed to categorise 2 out of 3 topics.

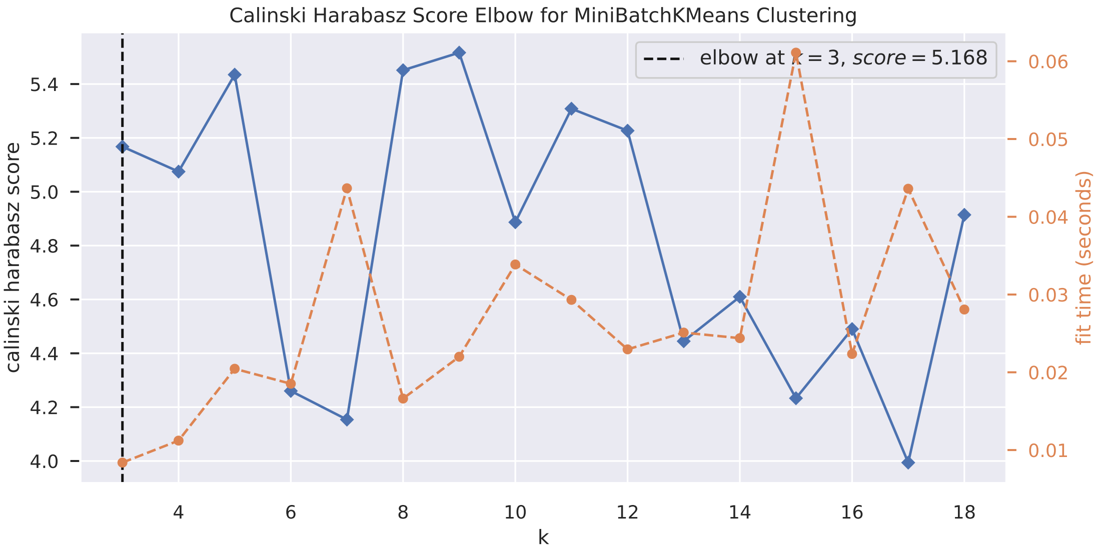
- Topic 0: anc ifp owe kwi ye ekzn uzuma abantu kuboshwe uramaphosa
- Topic 1: imibono izingane umalema ungqongqoshe enkantolo ekzn uramaphosa kusolwa bafuna umfundi
- Topic 2: izingane bafuna enkantolo ekzn ungqongqoshe ngokubulawa kuboshwe umfundi iphoyisa da

siSwati News Title Topics

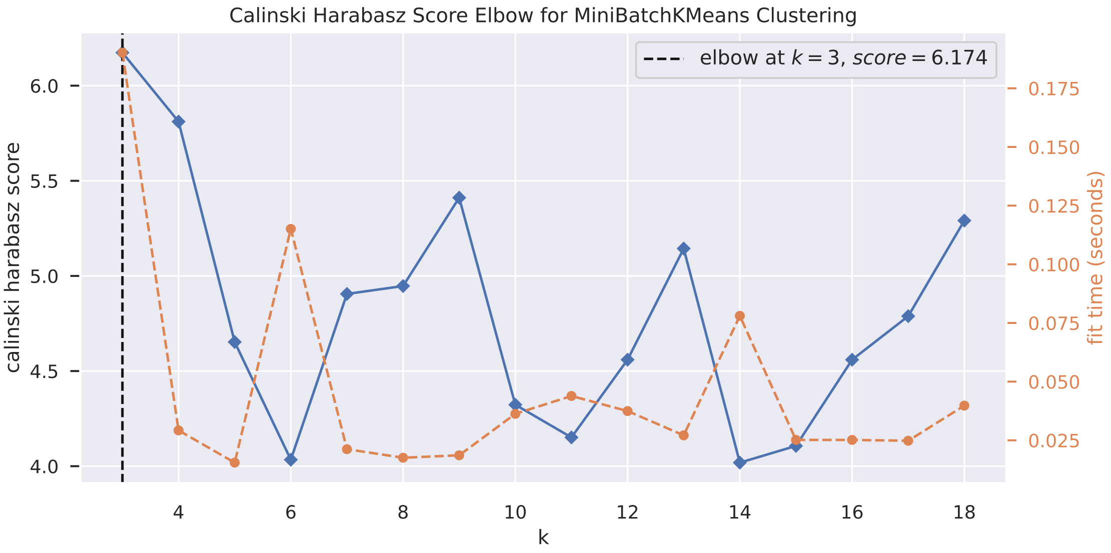
The topic 2 speaks about Madumane and Hlengiwe, who are musicians, hence this can be categorised as ’art and entertainment’.The other two topics are not clear to be categorised. NMF model performed poorly on siSwati news title datasets since we could only categorise 1 out of 3 topics, the NMF model performed better on isiZulu news title dataset as compared to siSwati,however, this could be due to the fact that siSwati dataset has small number observations.

- Topic 0: njalo nobe sikhatsi afrika yami inyandzaleyo naye ngobe wa kube
- Topic 1: njalo afrika co za isms kule ngabe wa ungaphutselwa ya
- Topic 2: madumane makwakwa ntsimbi real njalo hlengiwe afrika ungaphutselwa co za

© University of Pretoria



Graph 4.1: MiniBatch Kmeans Clustering for isiZulu Titles



Graph 4.2: Kmeans Clustering for siSwati Titles

© University of Pretoria

4.6.2 Supervised

The annotated isiZulu and siSwati news datasets were split using kfold cross validate(k=5), where the other piece of the data will be used as training dataset and the remaining piece of the dataset as test dataset for the model text classification performance. The model parameters details and the approach taken for each model is explained in detail below.

Logistic Regression Model

Multinomial logistic regression classification model was used to classify the news data for both isiZulu and siSwati.The parameters used to train the model were defined as follow:

```
    clf = LogisticRegression(solver = 'saga', multi_class = 'multinomial')
```

Naïve Bayes Model

Multinomial naïve bayes was fitted to perform the classification for news article. The default parameters were used to train the model to perform the classification. The model was fitted with the following parameters

```
    clf = MultinomialNB()
```

And the best parameters obtained from using the GridSearch were used train the multinomial naïve bayes model and predict the categories of the news article for both isiZulu and siSwati.

XGBoost model

Xgboost model was used to perform the isiZulu and siSwati news classification and the model was defined as follow

```
    clf =xgboost.XGBClassifier(num_class=4,objective='multi:softprob')
```

LSTM

LSTM model that was used to classify the news articles/titles for isiZulu and siSwati. The model was set up to have the input layer, attention layers, two bidirectional LSTM

© University of Pretoria

layer and dense layer as shown on the below python code.

The LSTM model made the prediction where each category was assigned the probability and the category with the highest probability was considered to be the suitable category for that article/title(document).

```python
def attention_layer(inputs, neurons):
    x = layers.Permute((2,1))(inputs)
    x = layers.Dense(neurons, activation="softmax")(x)
    x = layers.Permute((2,1), name="attention")(x)
    x = layers.multiply([inputs, x])
    return x

# input
x_in = layers.Input(shape=(15,))
# embedding
x = layers.Embedding(input_dim=embeddings.shape[0],
                     output_dim=embeddings.shape[1],
                     weights=[embeddings],
                     input_length=15, trainable=False)(x_in)
# apply attention
x = attention_layer(x, neurons=15)
# 2 layers of bidirectional lstm
x = layers.Bidirectional(layers.LSTM(units=15, dropout=0.2,
                         return_sequences=True))(x)
x = layers.Bidirectional(layers.LSTM(units=15, dropout=0.2))(x)
# final dense layers
x = layers.Dense(64, activation='relu')(x)
y_out = layers.Dense(38, activation='softmax')(x)
# compile
model = models.Model(x_in, y_out)
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])
```

© University of Pretoria

4.7 Model Evaluation

The model performance evaluation for the isiZulu and siSwati text classification was done using the accuracy score, F1 score and confusion matrix. The model with the highest accuracy score, f1 score and precision was regarded as the best model for the text classification of isiZulu and siSwati news article.

```
metrics.classification_report(y_test,predicted)
metrics.confusion_matrix(y_test, predicted)
```

The LIME model was used to provide insights of the black box classification models to validate that the models are using the correct words to make a prediction.The LIME model was applied on the most and the least performing models to assess why the model performed well and why the other did not performing well.Therefore, With the support of LIME model,f1-score,accuracy and confusion matrix we were able to conclude on best model for the classification of isiZulu and siSwati datasets.

4.8 Summary

This chapter included all the actual tasks that were executed to deliver the results.The tasks included data cleaning, word embedding creation, Data Augmentation, SMOTE, model development and model evaluation.Therefore, in the next chapter we present and analyse the results obtained from the above executed tasks.

Chapter 5

Results and Discussion

This chapter focuses on the performance of the classification models and the analysis of the same. The chapter is divided into three sections based on the sampling techniques used to balance the class category before training the classification models.The three sections are listed below:

- Original Data
- Augmentation
- SMOTE

Section 5.1 covers the model performance where the classification model was trained using the original dataset, section 5.2 covers the model performance where the contextual Data Augmentation was applied on the original dataset, section 5.3 covers the model performance where the SMOTE sampling technique was applied on the original dataset to balance the class categories and lastly, Section 5.4 summarises the aforementioned sections.

5.1 Original Datasets

This section contains the model classification results when the original class imbalanced datasets were used to train and test the models. Each of the isiZulu news articles,

© University of Pretoria

isiZulu Titles original and siSwati Titles datasets were preprocessed and fed into the classification model. The sampling technique were applied next sections to overcome the class imbalance problem and the model performance prior and post the application of sampling techniques was assessed.

The classification models, namely, Naive Bayes, logistic regression, Xgboost and LSTM were trained using the original dataset and below is the performance of each model and the corresponding utilised vectorizer.

5.1.1 Original Data-isiZulu Articles Model Training

The Table 5.1 shows the model performance obtained from training the models on original dataset, it was oberserved that the combination of Word2vec and Naive Bayes model performed very poor, whereas the combination of Word2vec and LSTM model outperformed all the models, obtaining the accuracy of 83.11% and f1-score of 82.78% as shown in Table 5.1. In general, all the classical models performed poorly because of the class imbalance, this can be drawn from the precision score of 21.73% from the least performing model(Naive Bayes Model),however,the original isiZulu Articles dataset suffer from class imbalance, hence, all classification models did not perform well except LSTM model. Furthermore, the confusion matrix Figure 5.1 shows that LSTM classified most of the news articles correctly regardless of the existing class imbalance. On the other hand, Model trained using TFIDF word-embedding showed a very poor performance, scoring f1-score less than 30% and it was observed from the XGBoost(TFIDF) confusion matrix Figure 5.2 below that the model classified majority of the documents into major class category *crime,law and justice.*

© University of Pretoria

Table 5.1: isiZulu Articles Original Dataset Model Performance

Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	21.73	21.12	16.34	52.4	(13.29,19.4)
Count Vectorize	Logistic Regression	41.23	34.97	36.06	54.53	(32.09,40.03)
Count Vectorize	XGBoost	49.14	31.33	32.51	54.89	(28.64,36.38)
TF-IDF 1-grams	Naive Bayes	18.41	20.34	14.35	52.22	(11.45,17.24)
TF-IDF 1-grams	Logistic Regression	32.09	26.13	24.19	54.71	(20.65,27.73)
TF-IDF 1-grams	XGBoost	40.91	29.42	29.34	52.93	(25.58,33.1)
Word2vec	Naive Bayes	61.98	50.99	53.04	68.39	(48.91,57.16)
Word2vec	Logistic Regression	70.18	62.91	65.13	75.32	(61.19,69.07)
Word2vec	XGBoost	67.69	52.23	55.83	69.1	(51.73,59.93)
Word2vec	LSTM	**83.39**	**83.11**	**82.78**	**83.11**	(79.66,85.9)

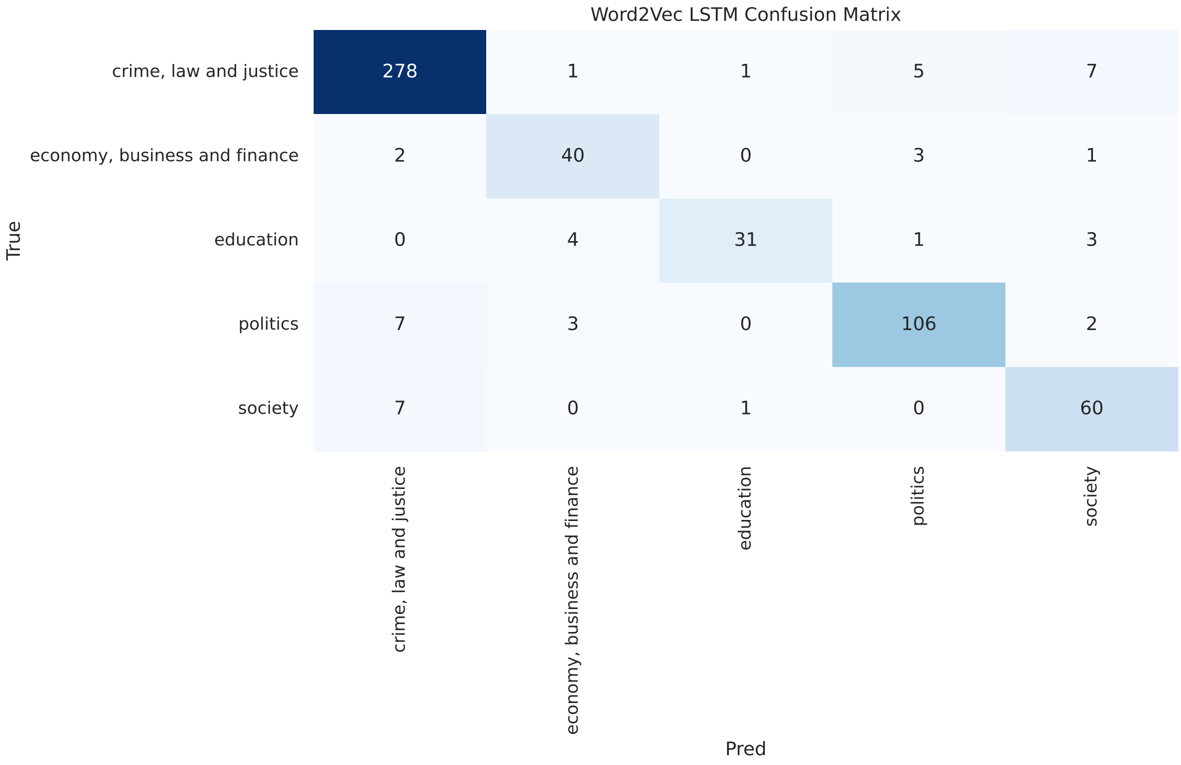


Figure 5.1: Word2vec-LSTM Confusion Matrix for isiZulu Articles Original Dataset

© University of Pretoria

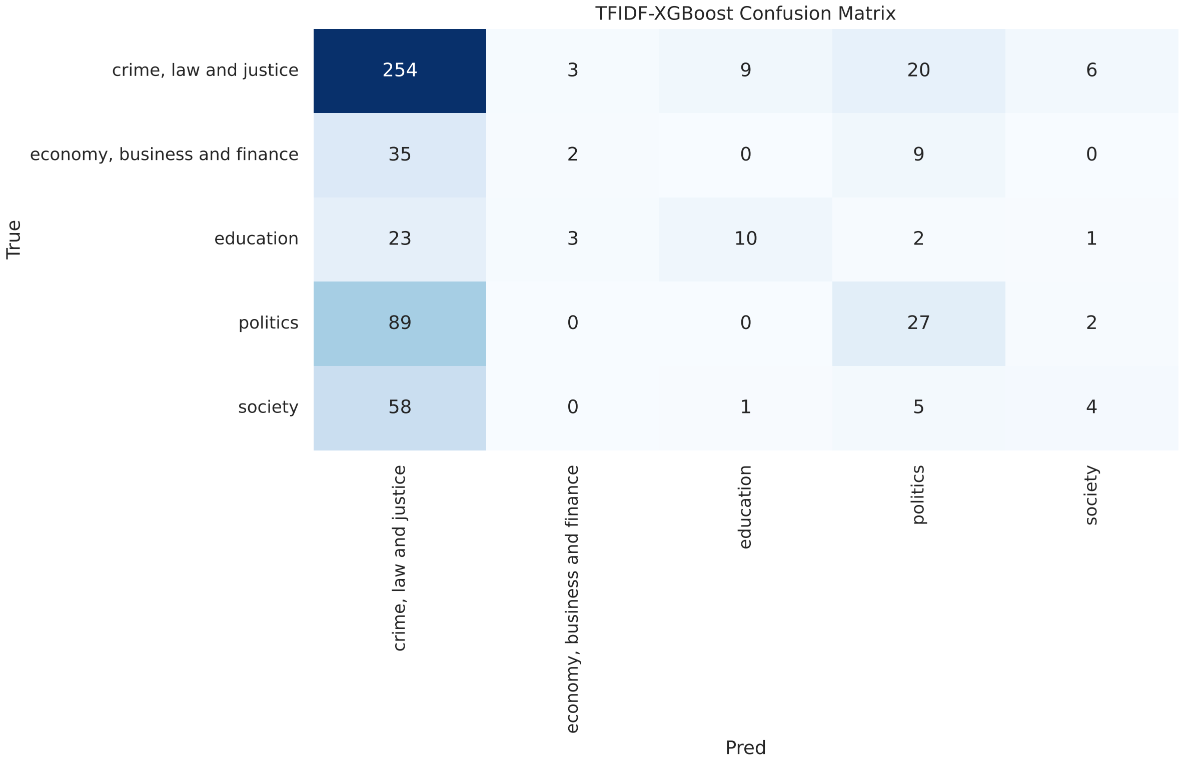


Figure 5.2: TFIDF-XGBoost Confusion Matrix for isiZulu Articles Original Dataset

5.1.2 Original Data-isiZulu Titles Model Training

The isiZulu Titles dataset contains the titles from the isiZulu Articles dataset, therefore, they have the same class imbalance and the isiZulu Titles are shorter as compared to the full articles in terms of text size, so the amount of texts also differ, as the title dataset contains short text.

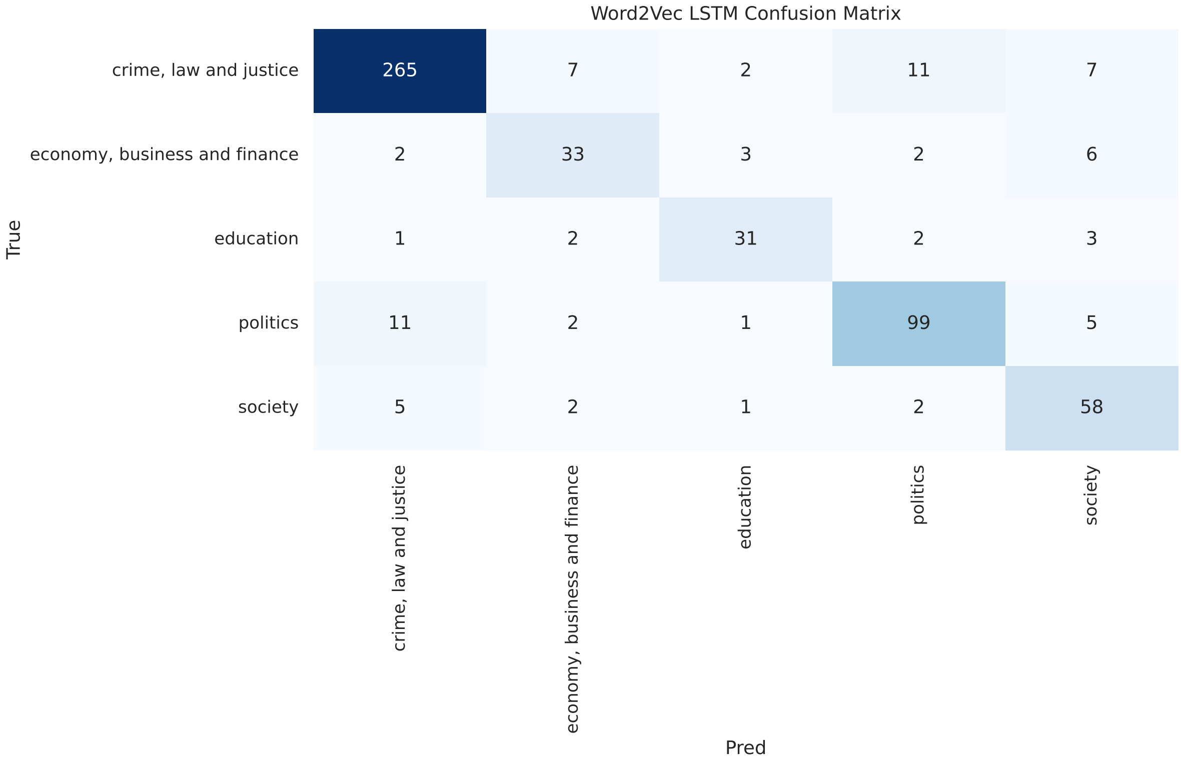
The models were trained and table 5.2 visualises the obtained models performance, all the models performed poorly scoring the accuracy below 60%, except for LSTM model. Word2vec preprocessing and LSTM model performed better than all the models, obtaining the accuracy of 71.75% and f1-score of 72.01% as shown in the Table 5.2. Moreover, from the confusion matrix in Figure 5.3, it was observed that the classification model managed to correctly classify most of the title documents, however, the class imbalance can still be observed from the amount of titles that are correctly classified as

© University of Pretoria

crime,Law and Justice as compared to other class categories, the other class categories have few observations than class category *crime,law and justice*. Therefore, this explains the poor performance of the other machine learning algorithms, that is,class imbalance had negative impact on the performance of classical models, with Naive Bayes(Count Vectorizer) being the least performing model with the lowest precision of 17.6%. Models trained using Count and TFIDF vectorizers produced f1-score less than 30% , Logistic regression confusion matrix in Figure 5.4 below shows that majority of the news titles were incorrectly classified into the major class category *crime, law and justice*.

Table 5.2: isiZulu Titles Original Dataset Model Performance

Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	17.6	20.62	15.33	51.69	(12.36,18.31)
Count Vectorize	Logistic Regression	18.36	21.83	17.38	52.76	(14.25,20.51)
Count Vectorize	XGBoost	20.91	21.23	17.03	51.51	(13.92,20.13)
TF-IDF 1-grams	Naive Bayes	19.89	20.89	15.57	52.4	(12.57,18.56)
TF-IDF 1-grams	Logistic Regression	20.47	21.9	17.58	52.93	(14.44,20.73)
TF-IDF 1-grams	XGBoost	18.07	20.79	16.37	51.34	(13.31,19.43)
Word2vec	Naive Bayes	27.83	25.58	22.75	57.2	(19.29,26.22)
Word2vec	Logistic Regression	41.85	38.65	39.18	57.72	(35.14,43.21)
Word2vec	XGBoost	40.63	31.17	31.03	57.73	(27.21,34.85)
Word2vec	LSTM	**72.96**	**71.75**	**72.01**	**71.75**	(68.3,75.72)



	crime, law and justice	economy, business and finance	education	politics	society
crime, law and justice	265	7	2	11	7
economy, business and finance	2	33	3	2	6
education	1	2	31	2	3
politics	11	2	1	99	5
society	5	2	1	2	58

Figure 5.3: Word2vec-LSTM Matrix for isiZulu Titles Original Dataset

© University of Pretoria

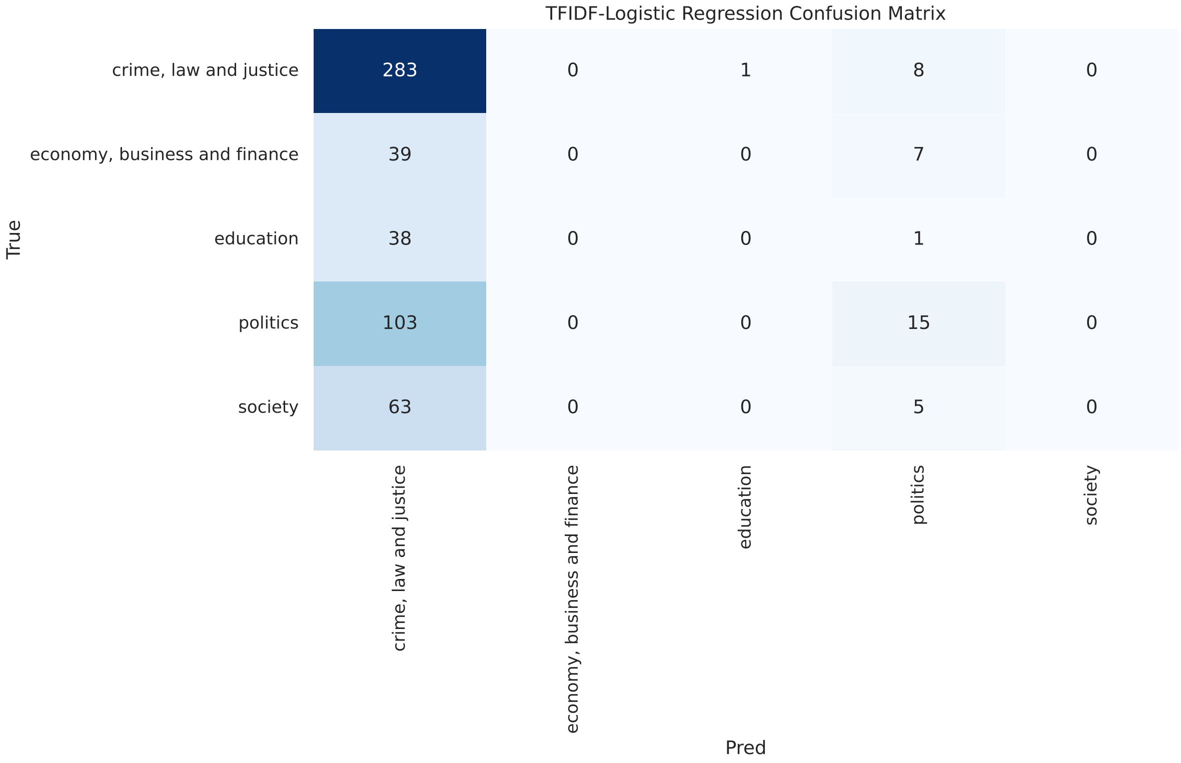


Figure 5.4: TFIDF-Logistic Regression Matrix for isiZulu Titles Original Dataset

5.1.3 Original Data-siSwati Titles Model Training

siSwati original dataset was fed into the models to observe the performance of each model on the original dataset that suffers from class imbalance. The results in the Table 5.3 shows that the models did not perform well except for LSTM (Word2vec) that scored the accuracy of 80.88% and f1-score of 81.06%. This confirms that LSTM works better than classical models on an imbalanced class dataset, since it has performed better than the other models in all three different datasets. The confusion matrix in the Figure 5.5 below confirms that LSTM model correctly classified majority of the siSwati news titles. Confusion matrix of the XGBoost model trained using TFIDF vectorizer shown on Figure 5.6 depicts the classification biasness as it was observed that most of the news titles were classified into the major class category *human interest*.

Table 5.3: siSwati Titles Original Dataset Model Performance

Preprocessing	Model	Precision(%)	Recall(%)	F1-score(%)	Accuracy(%)	Confidence Interval(f1 score)
Count Vectorize	XGBoost	25.75	25.52	24.23	41.54	(14.05,34.42)
Count Vectorize	Naive Bayes	25.37	30	25.39	53.19	(15.04,35.73)
Count Vectorize	Logistic Regression	25.93	30.1	26.34	48.79	(15.87,36.81)
TF-IDF 1-grams	Naive Bayes	13.61	22	15.61	48.68	(6.98,24.23)
TF-IDF 1-grams	Logistic Regression	17.77	24	18.81	50.33	(9.52,28.1)
TF-IDF 1-grams	XGBoost	25.16	29.33	25.5	47.58	(15.14,35.86)
Word2vec	Naive Bayes	31.77	34.76	31.57	59.01	(20.52,42.61)
Word2vec	Logistic Regression	29.59	32	28.09	57.58	(17.4,38.77)
Word2vec	XGBoost	28.77	31.43	27.96	54.84	(17.29,38.62)
Word2vec	LSTM	**87.53**	**80.88**	**81.06**	**80.88**	(71.75,90.37)

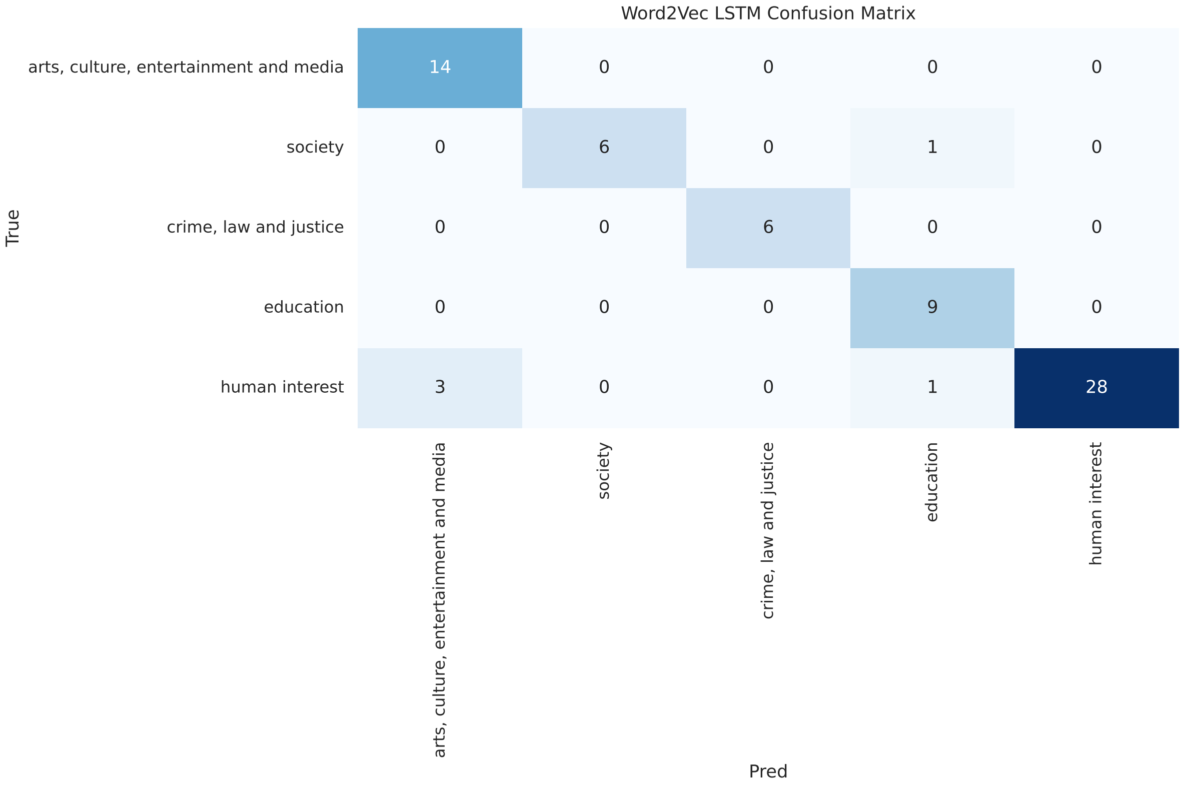
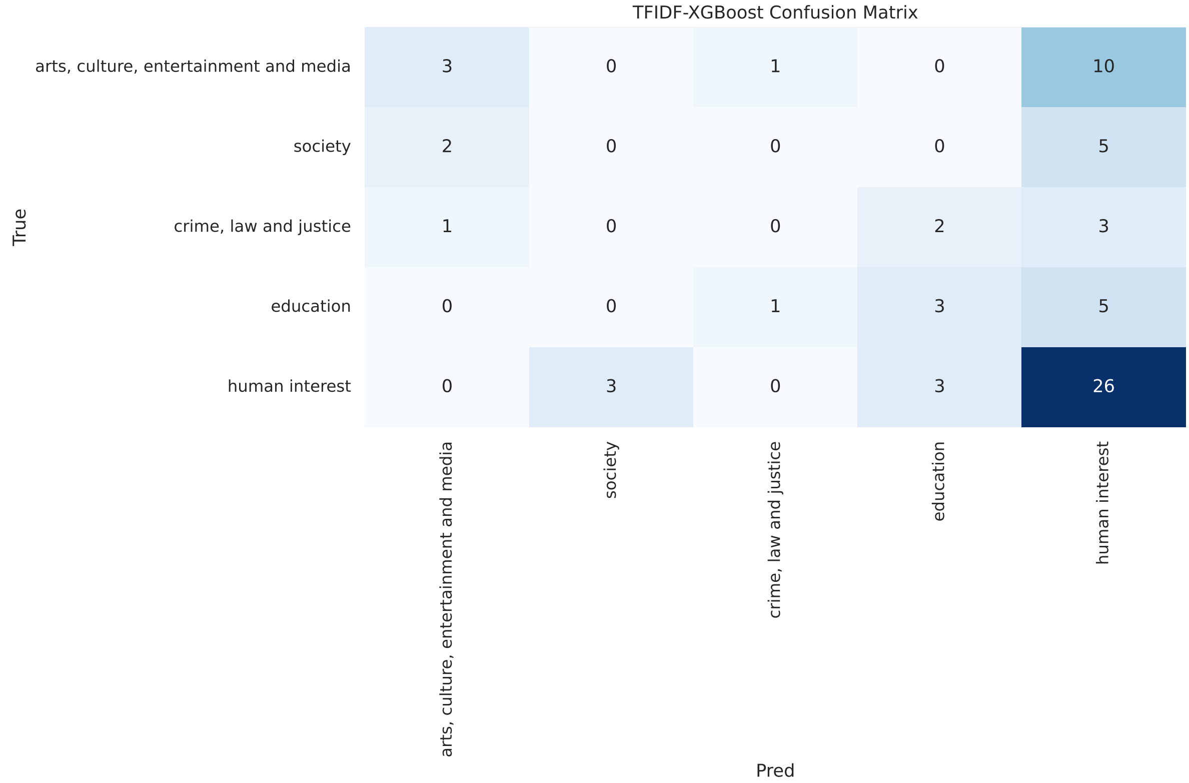


Figure 5.5: Word2vec-LSTM Matrix for siSwati Titles Original Dataset

© University of Pretoria



True \ Pred	arts, culture, entertainment and media	society	crime, law and justice	education	human interest
arts, culture, entertainment and media	3	0	1	0	10
society	2	0	0	0	5
crime, law and justice	1	0	0	2	3
education	0	0	1	3	5
human interest	0	3	0	3	26

Figure 5.6: TFIDF-XGBoost Matrix for siSwati Titles Original Dataset

5.2 Augmentated Datasets

Contextual Data Augmentation was applied on the datasets to increase the data size and balance the class categories.The results for each language are shown and discussed below.

5.2.1 Augmentation-isiZulu Articles Model Training

The isiZulu Articles dataset was augmented to increase the data size and balance the class categories, then fed into the models to perform topic classification. The models were trained and Table 5.4 shows the performance of the models and it was observed that Data Augmentation improved the models performance as compared to the results obtained from the isiZulu Articles original dataset.

© University of Pretoria

Table 5.4: isiZulu Articles Augmented Dataset Model Performance

Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	71.65	68.55	68.42	68.89	(65.87,70.97)
Count Vectorize	Logistic Regression	83.35	83.92	83.09	83.23	(81.04,85.15)
Count Vectorize	XGBoost	74.28	73.85	73.68	73.51	(71.26,76.09)
TF-IDF 1-grams	Naive Bayes	75.71	73.77	73.6	73.98	(71.18,76.02)
TF-IDF 1-grams	Logistic Regression	79.65	79.91	79.2	79.39	(76.97,81.42)
TF-IDF 1-grams	XGBoost	80.44	80.44	79.92	80.02	(77.72,82.11)
Word2vec	Naive Bayes	72.37	71.79	71.79	71.31	(69.32,74.26)
Word2vec	Logistic Regression	91.6	91.9	91.3	91.3	(89.75,92.84)
Word2vec	XGBoost	95.54	**95.73**	**95.21**	**95.14**	(94.04,96.39)
Word2vec	LSTM	**96.08**	94.45	94.45	94.45	(93.2,95.71)

The models performed well, more especially with the Word2vec vectorizer. The combination of XGBoost model and Word2vec outperformed all the models with the accuracy of 95.14% and f1-score of 95.21%. LSTM model came second in terms of performance, scoring the accuracy of 94.45% and f1-score of 94.45%. On the other hand, Naive Bayes model and Count Vectorizer combination produced the worst results as compared to other models and Vectorizer combination, scoring the accuracy of 68.89% and f1-score of 68.42% as shown in Table 5.4. However, Naive Bayes model performed better when the TFIDF vectorizer was utilised. Therefore, we need to deep-dive to unpack the blackbox models and understand the words that were considered important by the model when making predictions.

The LIME model was used to explain the blackbox classification models, however, only the best performing model was explained and in this case, it is XGBoost(Word2vec). One article was selected and used to explain the model hidden prediction process. The model predicted and true classes for the selected article(document 3) are shown below

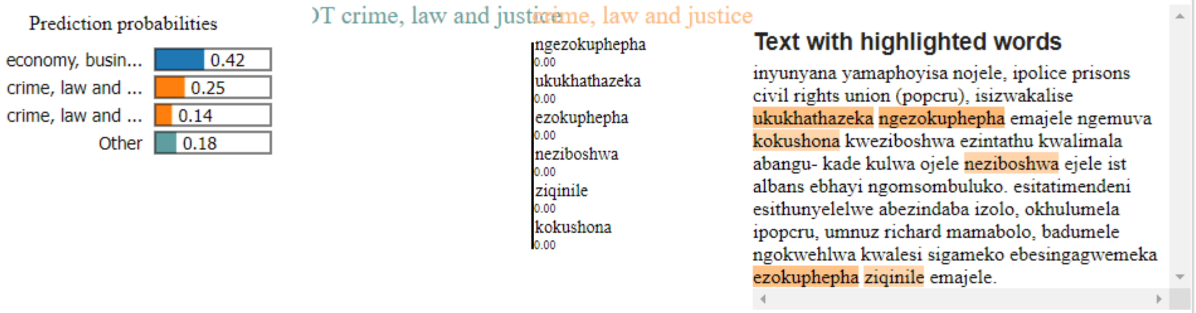
Article(document: 3): *'INYUNYANA yamaphoyisa nojele, iPolice and Prisons Civil Rights Union (Popcru), isizwakalise ukukhathazeka ngezokuphepha emajele ngemuva kokushona kweziboshwa ezintathu kwalimala abangu-25 kade kulwa ojele neziboshwa ejele iSt Albans eBhayi ngoMsombuluko. Esitatimendeni esithunyelelwe abezindaba izolo, okhulumela iPopcru, uMnuz Richard Mamabolo, uthe badumele ngokwehlwa kwalesi sigameko ebesingagwemeka ukuba ezokuphepha ziqinile emajele.'*

- **document id:** 3
- **True Class:** *crime, law and justice*

© University of Pretoria

- **Predicted class:** *economy, business and finance*

The model incorrectly classified document 3 as the class category *economy, business and finance* got the higher prediction probability, however, LIME model produced the words that were considered for prediction of the correct class category *crime, law and justice*, the XGBoost model used the words such as *’ukhathazeka’(worried), ’ezokuphepha’(security), ’kokushon’(death), ’neziboshwa’(prisoners), e.t.c* which are valid words to be associated with class category *’crime, law and justice’* as shown in Figure 5.7, furthermore, there are other words that the model could have considered in the documents that are closely related to the class category *’crime, law and justice’* such as *’ejele’/ ’emajele’/’ojele’/’nojele’(prison) and ’ipolice’.* However, XGBoost(Word2vec) correctly classified majority of the documents as shown in confusion matrix Figure 5.8. On the other hand, the least performing model incorrectly classified a large number of documents and obtaining the precision of 71.65% as shown in Figure 5.10 and Table 5.4. Models trained using TFIDF vectorizer performed fairly good as they scored more than 70% f1-score as shown in Table 5.4. Confusion matrix Figure 5.9 shows that XGBoost model trained using TFIDF managed to correctly classify majority of the news articles, however, most *politics* articles were incorrectly classified as *crime, law and justice* and *society* , whereas most *crime, law and justice* were classified as *society* and *politics*, therefore, we can conclude that the model was unable to differentiate *politics*, *crime, law and justice* and *society.*



Text with highlighted words

inyunyana yamaphoyisa nojele, ipolice prisons civil rights union (popcru), isizwakalise ukukhathazeka ngezokuphepha emajele ngemuva kokushona kweziboshwa ezintathu kwalimala abangu- kade kulwa ojele neziboshwa ejele ist albans ebhayi ngomsombuluko. esitatimendeni esithunyelelwe abezindaba izolo, okhulumela ipopcru, umnuz richard mamabolo, badumele ngokwehlwa kwalesi sigameko ebesingagwemeka ezokuphepha ziqinile emajele.

Figure 5.7: Data Augmentation and XGBoost- Lime Model Explanation for isiZulu Articles

© University of Pretoria

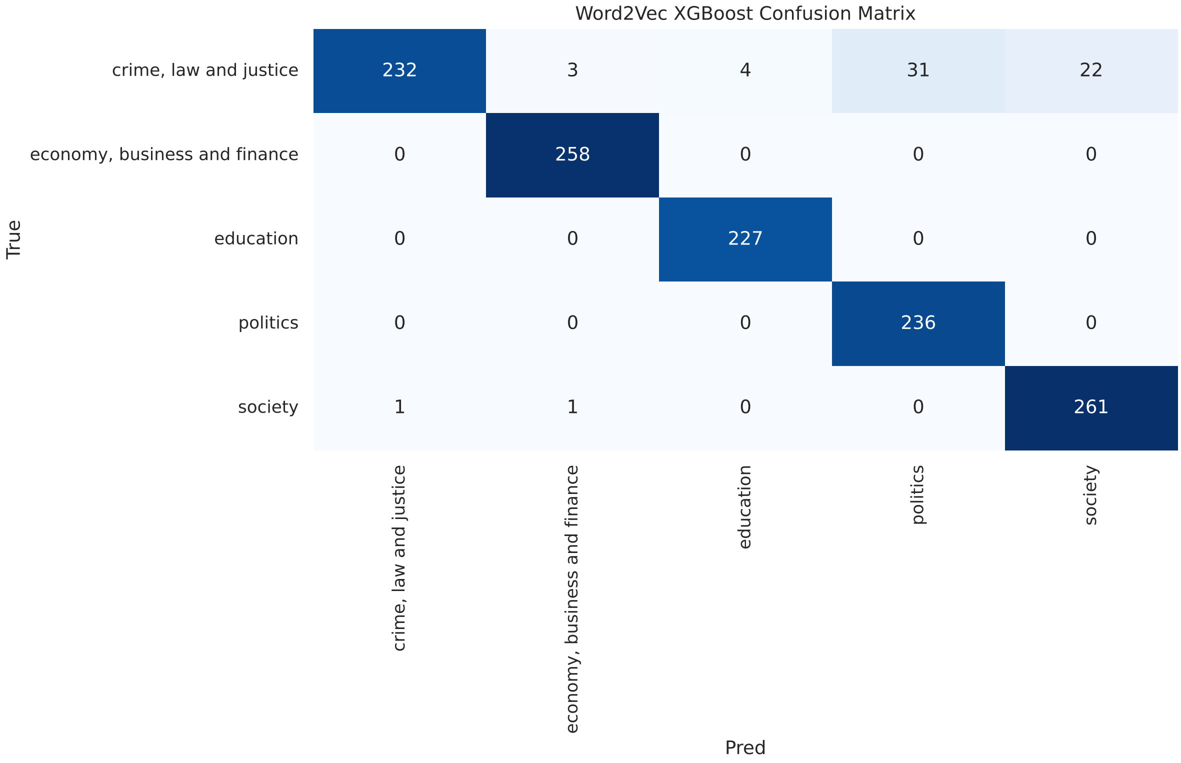


Figure 5.8: Word2vec-XGBoost Confusion Matrix for isiZulu Articles Augmented Dataset

© University of Pretoria

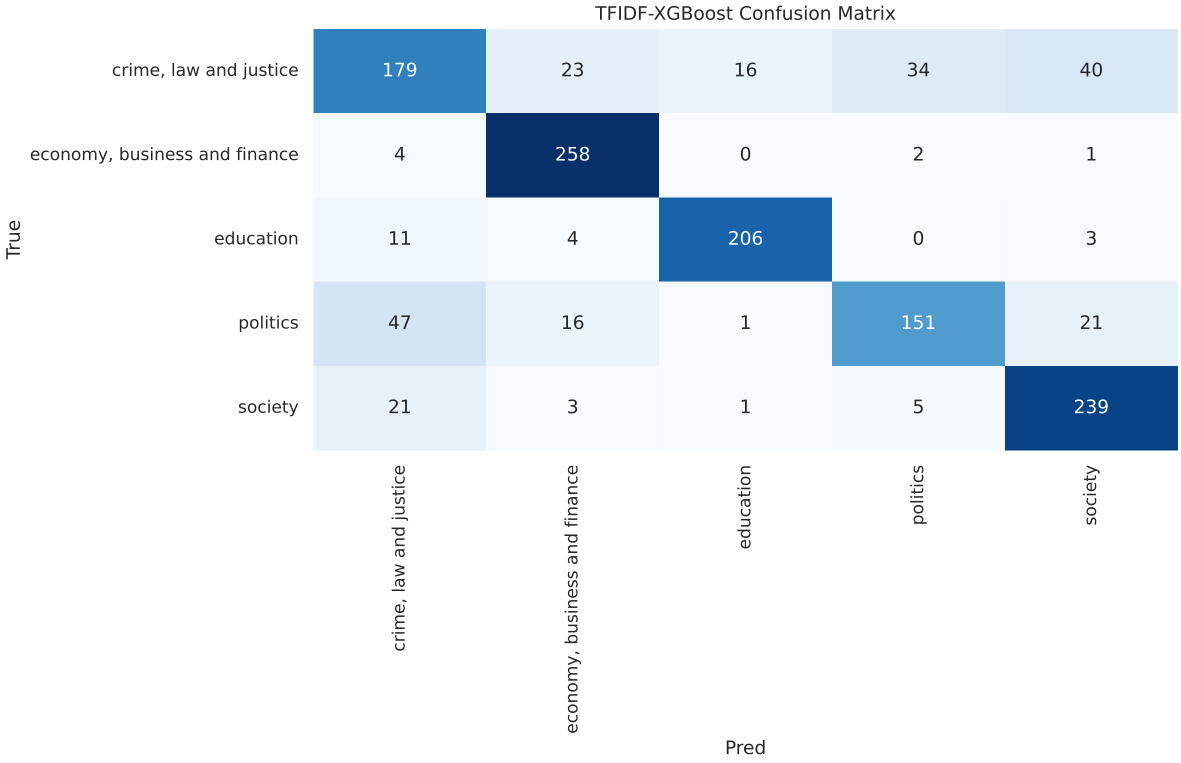


Figure 5.9: TFIDF-XGBoost Confusion Matrix for isiZulu Articles Augmented Dataset

© University of Pretoria

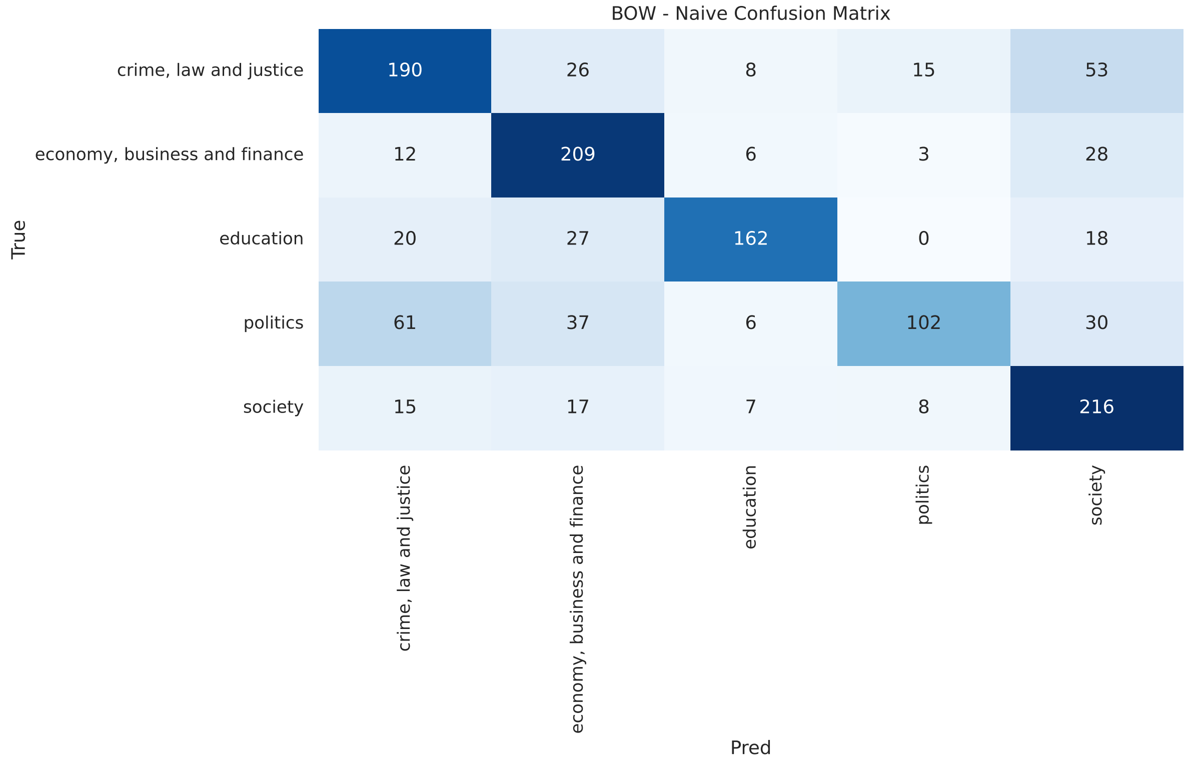


Figure 5.10: CountVectorizer-Naive Bayes Confusion Matrix for isiZulu Articles Augmented Dataset

5.2.2 Augmentation-isiZulu Titles Model Training

The isiZulu Titles augmented dataset was also utilised to train the models, perform topic classification, and the resulting models performance is shown in the Table 5.5 below. It was observed from the Table 5.5 that the models trained with Word2vec preprocessing performed better than the models trained with Count and TFIDF Vectorizer preprocessing by far. Moreover, Logistic Regrression, XGBoost and LSTM models(both trained with Word2vec) performed very well with the slight difference of f1 score/accuracy amongst themselves. However, Logistic Regression obtained the highest score and became the best performing model with 85.69% accuracy and 86.42% f1-score, followed by LSTM model with 84.96% accuracy and 85.83% f1-score. The combination of XGBoost model and Count Vectorizer performed poorly with the 33.27% accuracy and 24.47% f1-score as shown in the Table 5.5.

Table 5.5: isiZulu Titles Augmented Dataset Model Performance

Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	58.93	32.91	31.62	37.83	(28.86,34.37)
Count Vectorize	Logistic Regression	60.79	34.54	34.05	39.2	(31.24,36.85)
Count Vectorize	XGBoost	51.12	28.22	24.47	33.27	(21.92,27.01)
TF-IDF 1-grams	Naive Bayes	59.45	33.25	32.3	38.1	(29.54,35.07)
TF-IDF 1-grams	Logistic Regression	59.41	34.87	34.42	39.47	(31.6,37.23)
TF-IDF 1-grams	XGBoost	53.33	28.85	25.41	33.82	(22.83,27.98)
Word2vec	Naive Bayes	67.92	57.97	59.3	60.89	(56.39,62.21)
Word2vec	Logistic Regression	**86.35**	**87.65**	**86.42**	**85.69**	(84.39,88.45)
Word2vec	XGBoost	86.2	85.99	85.83	84.96	(83.77,87.89)
Word2vec	LSTM	85.32	85.16	84.37	85.16	(82.22,86.52)

Interrogating the model's prediction decision process, using the selected document 3(however,this is a title document) to perform the experiment. The model incorrectly categorized the document into 'politics' instead of 'crime, law and justice', hence incorrect prediction.

Title(Document 3): *'popcru ikhathazekile ngezokuphepha emajele'*

- **document id:** 3
- **True Class:** *crime, law and justice*
- **Predicted class:** *politics*

Lime model explained the title and it was noticed that the Logistic Regression(Word2vec) model is using only a word *popcru* to predict class category *crime, law and justice* and words like *ngezokuphepha* and *'emajele'* to predict other class category such as 'politics' as shown in Figure 5.11, hence the model was not able to collect enough information to be able to classify this document correctly, however, the model did well on the other documents, Logistic Regression model classified majority of the news titles correctly and that is supported by the confusion matrix in Figure 5.12 below and the precision of 86.35%. Logistic Regression model trained using TFIDF vectorizer under-performed in this category, scoring f1-score of 34.42% , the confusion matrix in Figure 5.13 revealed that the classification was biased to class category *crime, law and justice.* Moreover, the other confusion matrix in Figure 5.14 of the XGBoost model trained using Count

© University of Pretoria

Vectorizer shows how bad the least performing model is, the classification was also biased to one class category *crime, law and justice*.

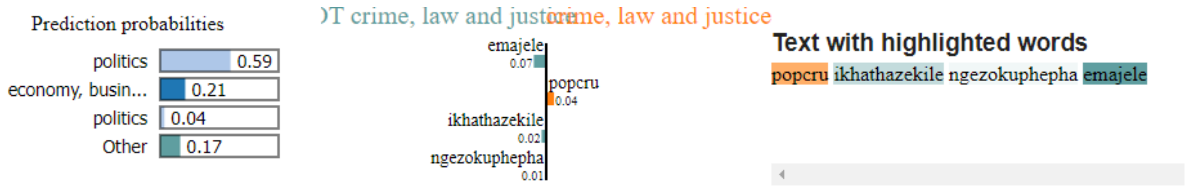
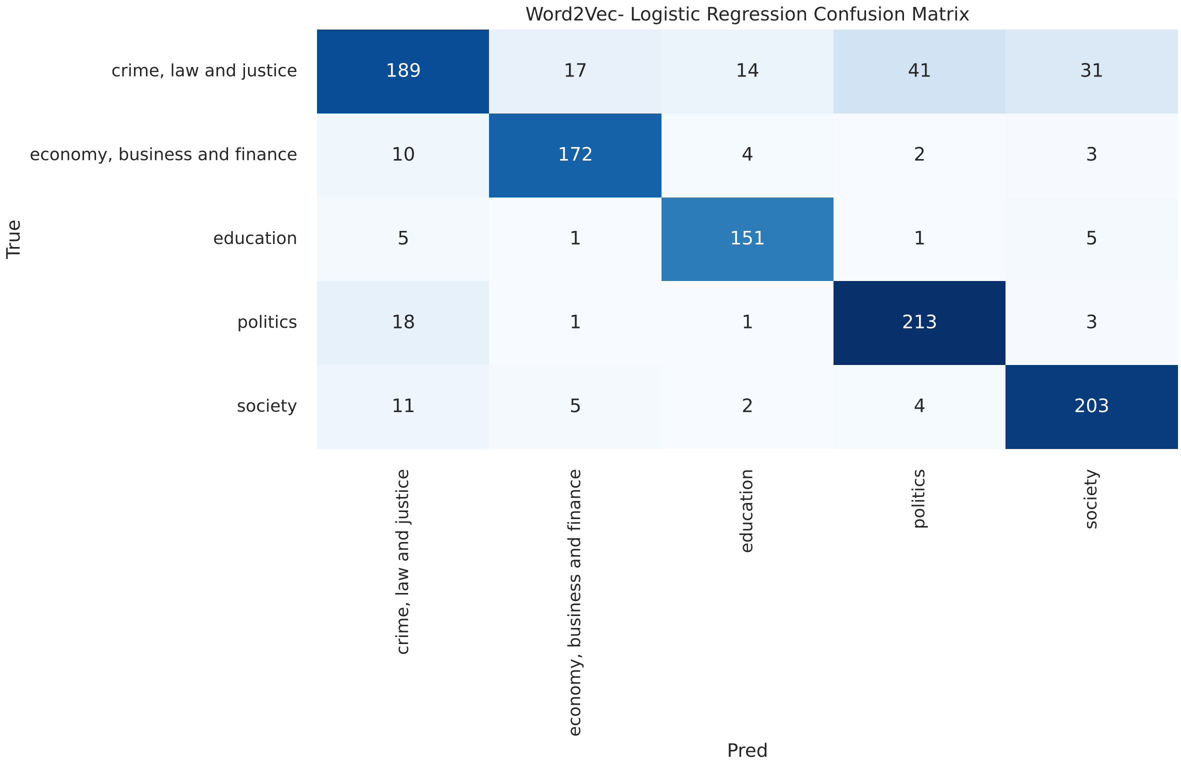


Figure 5.11: Data Augmentation and Logistic Regression-Lime Model Explanation for isiZulu Titles



	crime, law and justice	economy, business and finance	education	politics	society
crime, law and justice	189	17	14	41	31
economy, business and finance	10	172	4	2	3
education	5	1	151	1	5
politics	18	1	1	213	3
society	11	5	2	4	203

Figure 5.12: Word2vec-Logistic Regression Confusion Matrix for isiZulu Titles Augmented Dataset

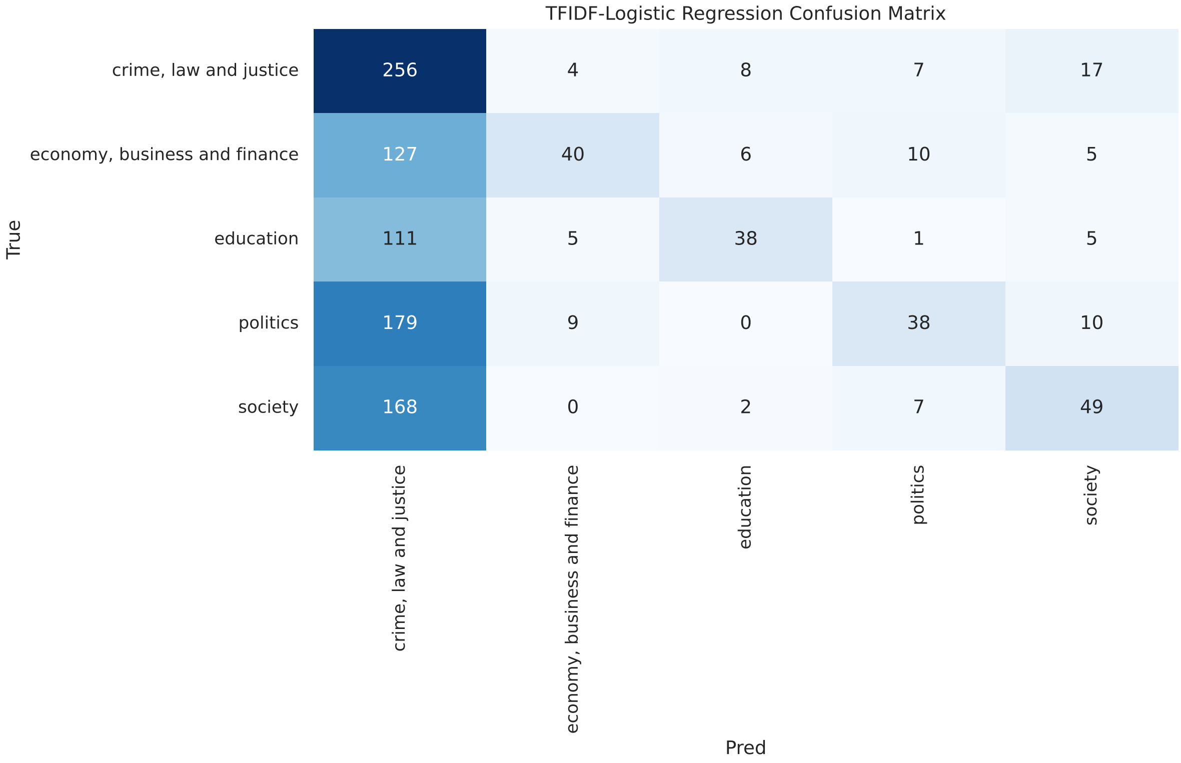


Figure 5.13: TFIDF-Logistic Regression Confusion Matrix for isiZulu Titles Augmented Dataset

© University of Pretoria

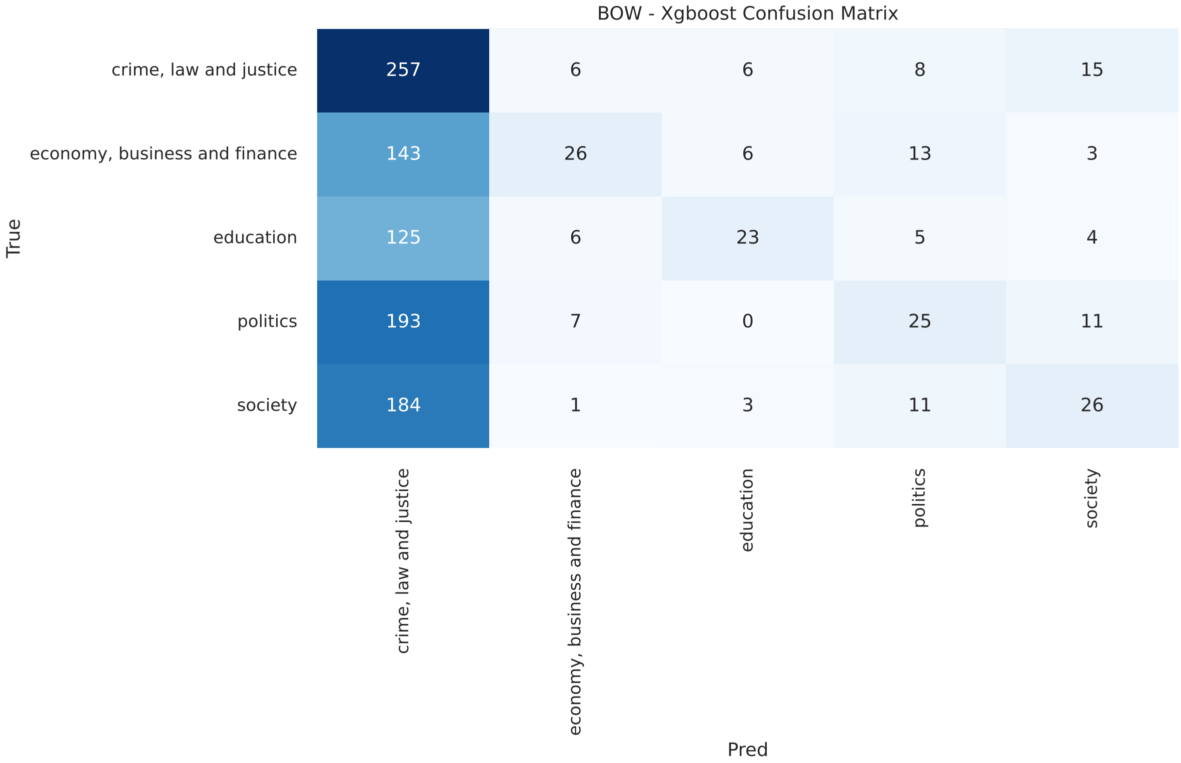


Figure 5.14: CountVectorizer-XGBoost Confusion Matrix for isiZulu Titles Augmented Dataset

5.2.3 Augmentation-siSwati Titles Model Training

The siSwati dataset was augmented and fed into the model to perform document classification and the performance results for each model was recorded on the Table 5.6. The models trained using Word2vec outperformed the models trained using Count Vectorizer and TFIDF vectorizers. Moreover, TFIDF vectorizer performed better than Count Vectorizer.

LSTM model outperformed all the other models obtaining 92.41% accuracy and 93.15% f1-score, and the least performing model was found to be Naive bayes(Count Vectorizer) with 68.79% accuracy and 69.35% f1 score as shown in Table 5.6.

© University of Pretoria

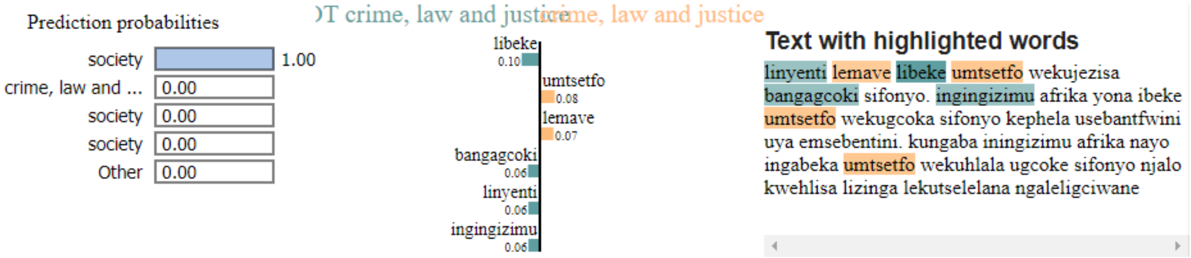
Table 5.6: siSwati Titles Augmented Dataset Model Performance

Preprocessing	Model	Precision(%)	Recall(%)	F1-score(%)	Accuracy(%)	Confidence Interval(f1 score)
Count Vectorize	Naive Bayes	71.98	69.52	69.35	68.79	(61.82,76.88)
Count Vectorize	Logistic Regression	78.78	74.8	74.74	74.31	(67.65,81.84)
Count Vectorize	XGBoost	81.99	74.7	74.47	74.33	(67.35,81.59)
TF-IDF 1-grams	Naive Bayes	75.67	73.03	72.85	72.24	(65.58,80.11)
TF-IDF 1-grams	Logistic Regression	78.93	75.5	75.57	75	(68.55,82.59)
TF-IDF 1-grams	XGBoost	81.1	74.13	73.09	73.62	(65.84,80.33)
Word2vec	Naive Bayes	84.26	83.41	82.52	82.66	(76.32,88.73)
Word2vec	Logistic Regression	91.17	89.9	87.83	88.89	(82.49,93.17)
Word2vec	XGBoost	91.57	91.33	89.8	90.22	(84.86,94.74)
Word2vec	LSTM	**94.88**	**92.41**	**93.15**	**92.41**	(89.02,97.27)

LSTM model was investigated to understand the decision process that the model used to perform the classification/prediction on the news titles. LSTM classified most the document correctly as depicted on the confusion matrix Figure 5.16 below and looking at the type of words that LSTM model managed to identify as important for the prediction of *crime, law and justice* class category , it is clear that the model was able to learn from the data since the word *'umtsetfo'(LAW)* is at least associated with the correct class category, although the document was not classified correctly as the other class category obtained the highest prediction category as shown in Figure 5.15. However, the confusion matrix in Figure 5.16 supports the observation that the LSTM model performed well and managed to correctly classify majority of the documents. The confusion matrix in Figure 5.16 shows that all the documents were classified correctly, however, the obtained f1-score and accuracy are not 100% since they were averaged from 5-fold cross validation iterations, therefore, other iterations produced 100% and other did not. Logistic Regression model trained using TFIDF vectorizer performed fairly good, obtaining f1-score of 75.57% and the confusion matrix in Figure 5.17 shows that the model predicted most of *human interest* titles as *art, culture, entertainment and media.* On the other hand, Naive Bayes model(Count Vectorizer) continues to under-perform, the confusion matrix Figure 5.18 shows the poor classification of the model as it can be seen that majority of documents are incorrectly classified.

Title:*'Linyenti lemave libeke umtsetfo wekujezisa bantfu uma bangagcoki sifonyo. Ingingizimu Afrika yona ibeke umtsetfo wekugcoka sifonyo kephela uma usebantfwini noma uya emsebentini. Kungaba njani uma Iningizimu Afrika nayo ingabeka umtsetfo* *wekuhlala ugcoke sifonyo njalo kwehlisa lizinga lekutselelana ngaleligciwane'*

- **document id:** 3
- **True Class:** *crime, law and justice*
- **Predicted class:** *Society*



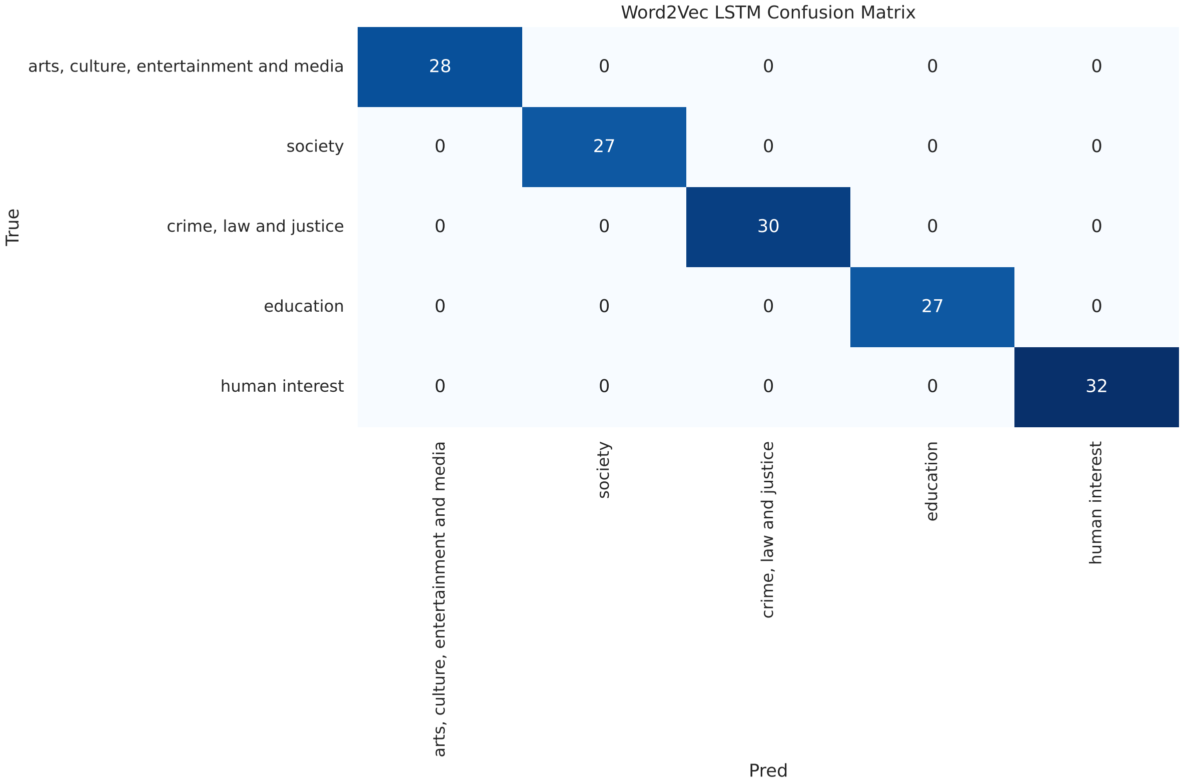
Prediction probabilities

society	1.00
crime, law and ...	0.00
society	0.00
society	0.00
Other	0.00

Text with highlighted words

linyenti lemave libeke umtsetfo wekujezisa bangagcoki sifonyo. ingingizimu afrika yona ibeke umtsetfo wekugcoka sifonyo kephela usebantfwini uya emsebentini. kungaba iningizimu afrika nayo ingabeka umtsetfo wekuhlala ugcoke sifonyo njalo kwehlisa lizinga lekutselelana ngaleligciwane

Figure 5.15: Data Augmentation and LSTM-LIME Model Explanation for siSwati Titles



Word2Vec LSTM Confusion Matrix

True \ Pred	arts, culture, entertainment and media	society	crime, law and justice	education	human interest
arts, culture, entertainment and media	28	0	0	0	0
society	0	27	0	0	0
crime, law and justice	0	0	30	0	0
education	0	0	0	27	0
human interest	0	0	0	0	32

Figure 5.16: Word2vec-LSTM Confusion Matrix for siSwati Titles Augmented Dataset

© University of Pretoria

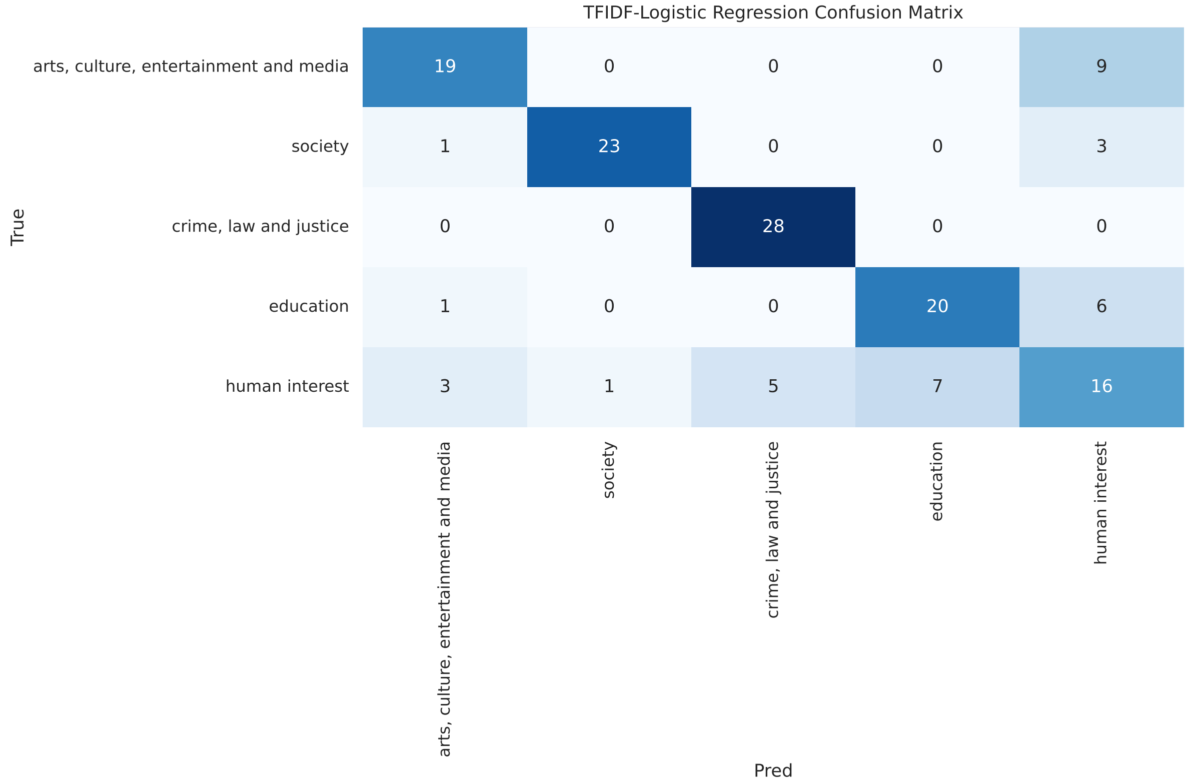
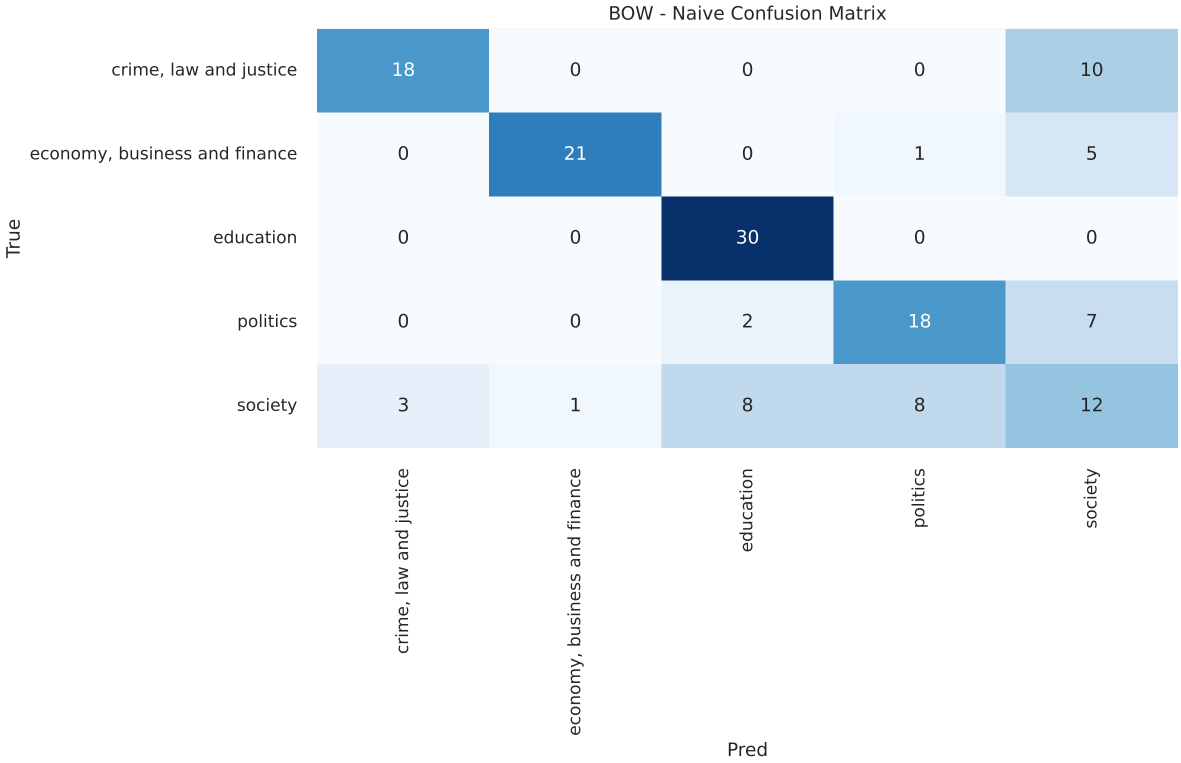


Figure 5.17: TFIDF-Logistic Regression Confusion Matrix for siSwati Titles Augmented Dataset



	crime, law and justice	economy, business and finance	education	politics	society
crime, law and justice	18	0	0	0	10
economy, business and finance	0	21	0	1	5
education	0	0	30	0	0
politics	0	0	2	18	7
society	3	1	8	8	12

Figure 5.18: CountVectorizer-Naive Bayes Confusion Matrix for siSwati Titles Augmented Dataset

5.3 SMOTE Datasets

The other approach that has been used to balance the data class categories is SMOTE. Each of the three datasets were fed into SMOTE model to increase and balance the dataset, thereafter, the four classification models were trained and the results are shown below.

5.3.1 isiZulu Articles Model Training

The models trained on isiZulu Articles dataset after applying SMOTE technique,performed pretty well more especially for Word2vec and TFIDF vectorizer whereas Count Vectorizer continues to under perform.

© University of Pretoria

The combination of XGBoost model and Word2vec with the accuracy of 93.56% and f1-score of 93.35% performed better than all the other models, with Logistic Regression model and Word2vec taking the second position with 92.12% accuracy and 91.88% f1-score. On the other hand, the least performing model was found to be Naive bayes model(Count Vectorizer) with 39.04% accuracy and 39.63% f1-score, followed by logistic regression model (Count Vectorizer) with 51.16% accuracy and 50.08% f1 score as shown in the below table 5.7.

Table 5.7: isiZulu Articles SMOTE Dataset Model Performance

Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	56.37	39.06	36.63	39.04	(34.16,39.11)
Count Vectorize	Logistic Regression	55.67	51.19	50.08	51.16	(47.52,52.65)
Count Vectorize	XGBoost	82.31	76.34	75.99	76.37	(73.8,78.18)
TF-IDF 1-grams	Naive Bayes	78.93	77.81	76.83	77.81	(74.67,79.0)
TF-IDF 1-grams	Logistic Regression	82.2	82.38	81.68	82.4	(79.7,83.66)
TF-IDF 1-grams	XGBoost	81.7	79.17	79.51	79.18	(77.44,81.58)
Word2vec	Naive Bayes	74.44	74.25	74.12	74.25	(71.87,76.37)
Word2vec	Logistic Regression	92.43	92.11	91.88	92.12	(90.48,93.28)
Word2vec	XGBoost	**93.75**	**93.55**	**93.35**	**93.56**	(92.08,94.63)

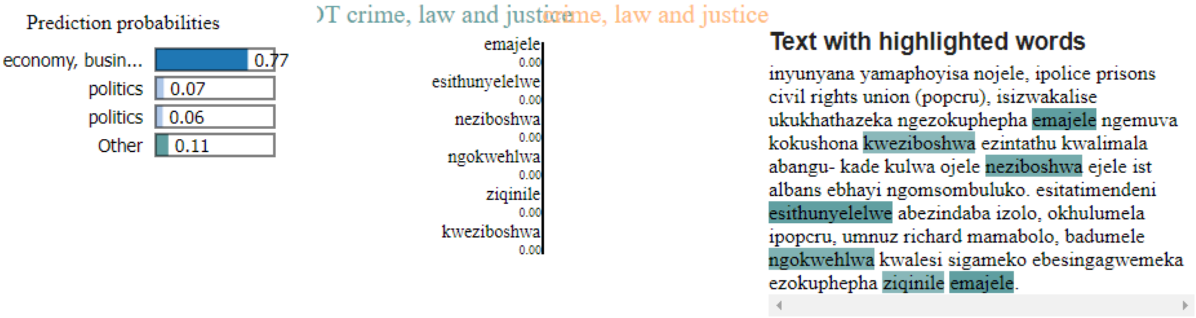
LIME model was used to understand the prediction decision-making process of the model and document 3 was used to perform the experiment. The best performing model XGBoost model(Word2vec) failed to correctly classify the article(document 3) as the document was classified as *politics* whereas the true class is *crime, law and justice* as it can be seen below.

- **document id:** 3
- **True Class:** *crime, law and justice*
- **Predicted class:** *economy, business and finance*

The explanation provided by LIME model regarding the classification performed above is shown below in Figure 5.19. It was observed that the model did not associate words such as *'emajele'*, and *'neziboshwa'* with *crime, law and justice* class category, meaning that the model failed to extract meaningful words from the input text, however, majority of the documents were classified correctly as shown on the confusion matrix in

© University of Pretoria

Figure 5.20 below. Logistic Regression model(TFIDF Vectorizer) managed to correctly classify majority of the news articles, obtaining 81.68% f1-score and the confusion matrix in Figure 5.21 shows that the model performed fairly good on the other class categories, however, model was unable to separate *crime, law and justice* from *politics* and *society.* On the other hand, the least performing model Naive Bayes Model(Count Vectorizer) incorrectly classified majority of the documents, as shown by the confusion matrix Figure 5.22 and explained by the precision score of 56.37% (shown in table 5.7), the model was biased towards class category *crime, law and justice* as majority of class were incorrectly classified into this class.



Text with highlighted words

inyunyana yamaphoyisa nojele, ipolice prisons civil rights union (popcru), isizwakalise ukukhathazeka ngezokuphepha emajele ngemuva kokushona kweziboshwa ezintathu kwalimala abangu- kade kulwa ojele neziboshwa ejele ist albans ebhayi ngomsombuluko. esitatimendeni esithunyelelwe abezindaba izolo, okhulumela ipopcru, umnuz richard mamabolo, badumele ngokwehlwa kwalesi sigameko ebesingagwemeka ezokuphepha ziqinile emajele.

Figure 5.19: SMOTE and XGBoost-LIME Explanation for isiZulu Articles

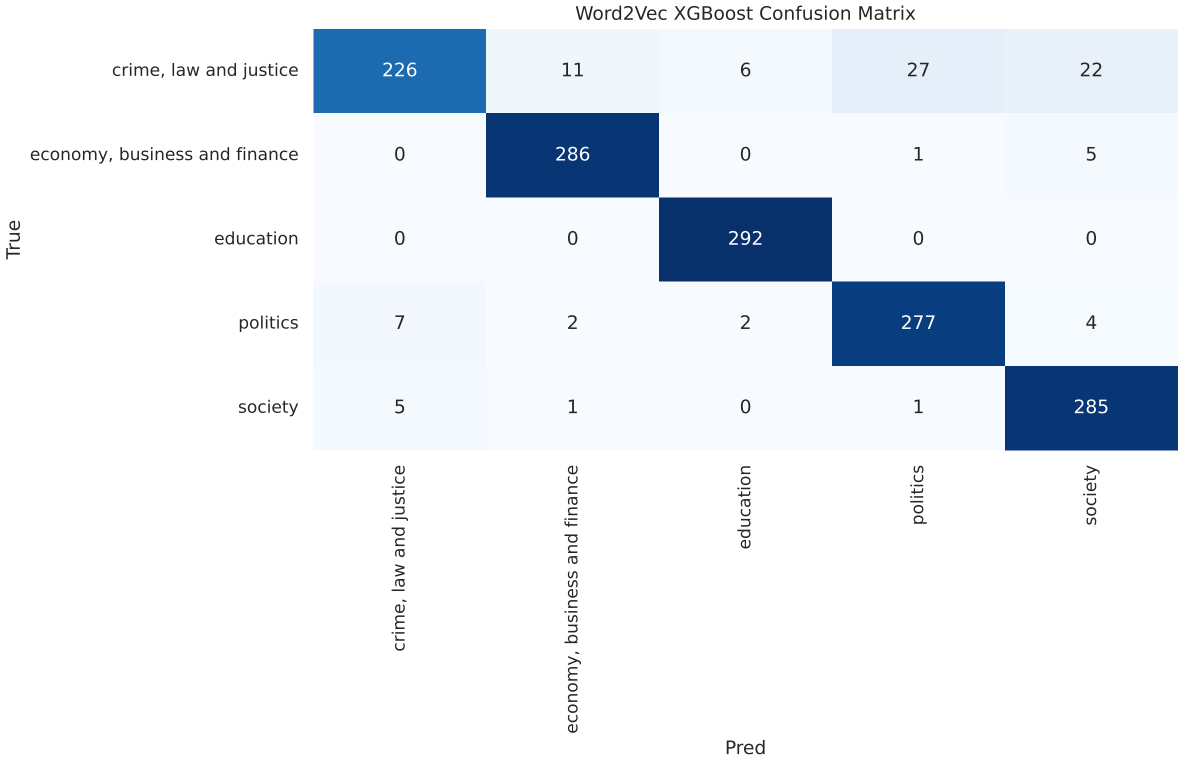
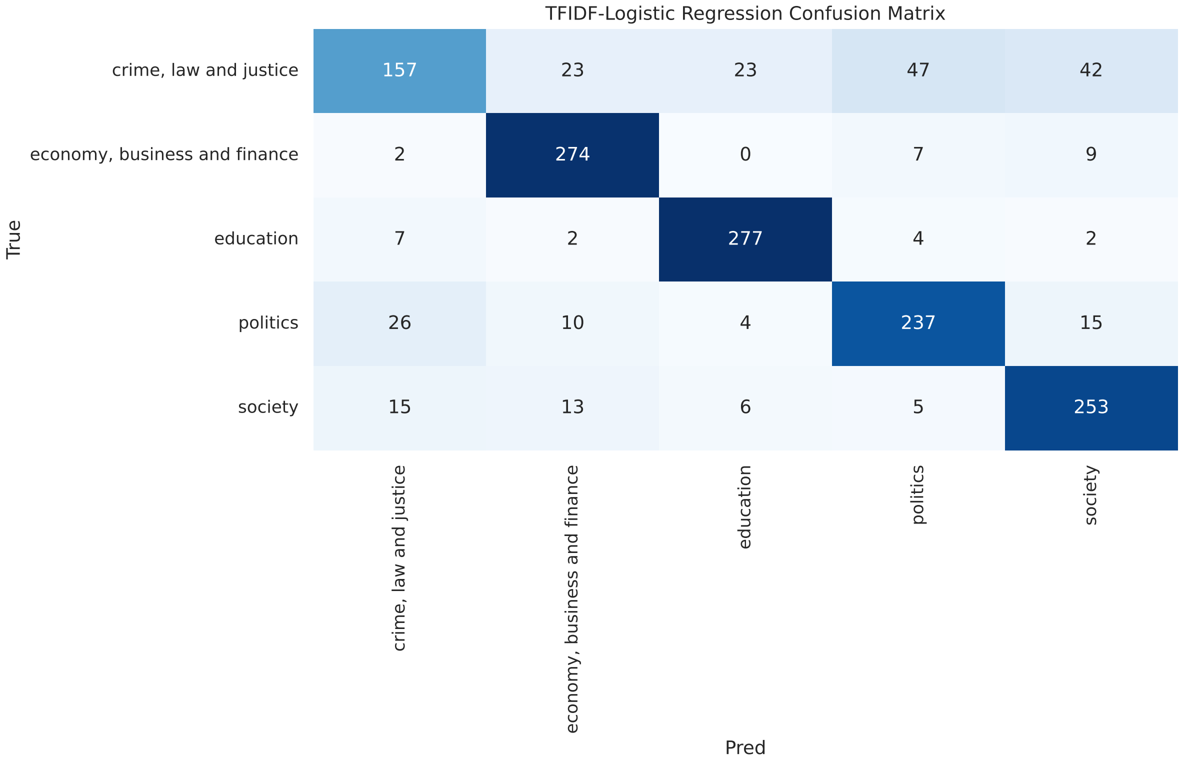


Figure 5.20: Word2vec-XGBoost Confusion Matrix for isiZulu Articles SMOTE Dataset

© University of Pretoria



	crime, law and justice	economy, business and finance	education	politics	society
crime, law and justice	157	23	23	47	42
economy, business and finance	2	274	0	7	9
education	7	2	277	4	2
politics	26	10	4	237	15
society	15	13	6	5	253

Figure 5.21: TFIDF-Logistic Regression Confusion Matrix for isiZulu Articles SMOTE Dataset

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

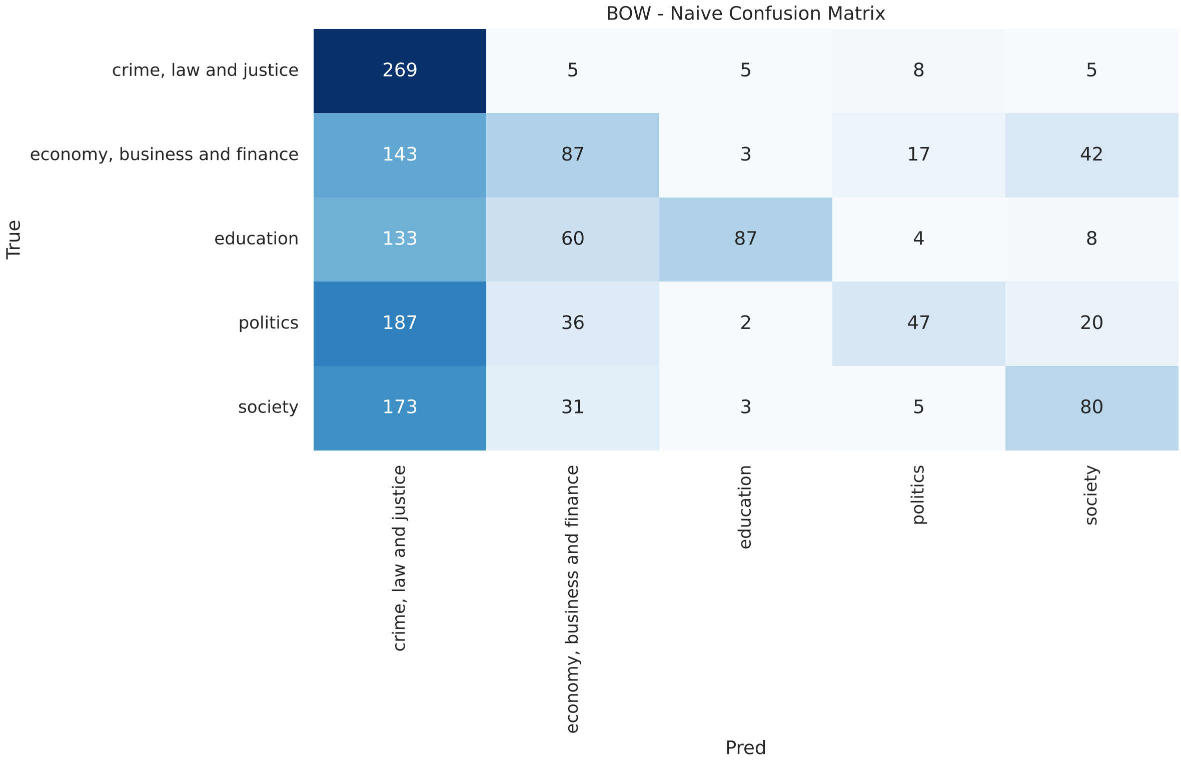


Figure 5.22: CountVectorizer-Naive Bayes Confusion Matrix for isiZulu Articles SMOTE Dataset

5.3.2 isiZulu Titles Model Training

The isiZulu Titles dataset was fed into the model post applying SMOTE for class categories balancing purpose, therefore, the performance results for the models is shown on Table 5.8 below. The models trained from Count and TFIDF vectorizers scored f1 score and accuracy below 40% whereas Word2vec scored over 70% on both accuracy and f1 score, hence it is clear that the models are performing well when Word2vec preprocessing is used than Count and TFIDF vectorizers.

The best model was found to be XGBoost (Word2vec) with 91.58% accuracy and 91.26% f1 score, and the least performing model to be Naive bayes model(Count Vectorizer) with 23.22% accuracy and 15.91% f1 score as shown in Table 5.8.

Table 5.8: isiZulu Titles SMOTE Dataset Model Performance

Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	36.92	23.33	15.91	23.22	(14.03,17.78)
Count Vectorize	Logistic Regression	46.08	25.9	18.23	25.89	(16.25,20.21)
Count Vectorize	XGBoost	65.14	38.34	37.52	38.36	(35.03,40.0)
TF-IDF 1-grams	Naive Bayes	64.37	37.69	37.38	37.6	(34.9,39.86)
TF-IDF 1-grams	Logistic Regression	65.23	39.72	39.71	39.73	(37.2,42.22)
TF-IDF 1-grams	XGBoost	65.6	38.2	37.56	38.22	(35.08,40.05)
Word2vec	Naive Bayes	74.49	74.02	73.85	74.04	(71.6,76.11)
Word2vec	Logistic Regression	91.56	91.08	90.63	91.1	(89.13,92.12)
Word2vec	XGBoost	91.96	91.56	91.26	91.58	(89.81,92.71)
Word2vec	LSTM	73.53	72.82	72.75	72.82	(69.08,76.43)

The best performing model XGBoost (Word2vec) incorrectly classified document 3 as shown below.

- **document id:** 3
- **True Class:** *crime, law and justice*
- **Predicted class:** *economy, business and finance*

LIME model revealed the words that led to that incorrect prediction. firstly, the text is shorter and it contains only one word that carries information related to *crime, law and justice* which is *emajele* as shown in Figure 5.23, hence the other words carried more weight towards the incorrect class, however, the model classified majority of the documents correctly as shown in the confusion matrix Figure 5.24. Logistic regression(TFIDF Vectorizer) performed poorly and confusion matrix Figure 5.25 shows that the classification was bias to class category *crime, law and justice.* Naive Bayes model(Count Vectorizer) continues to be the least performing model with its classification performance demonstrated on the confusion matrix in Figure 5.26, shows the model's inability to learn from the datasets, probably due to the model assumption that states that the effect of each predictor variable is independent of the other predictor variables for a particular class [86], however, it is worth a further investigation.

© University of Pretoria

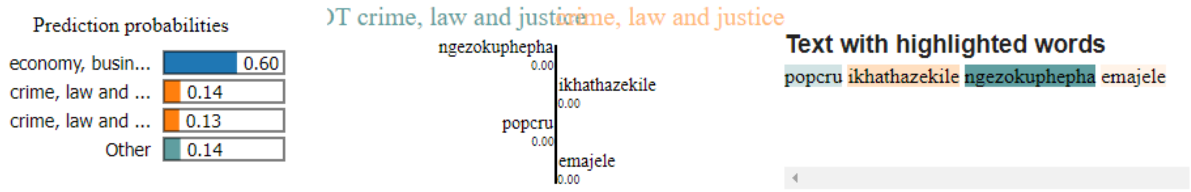
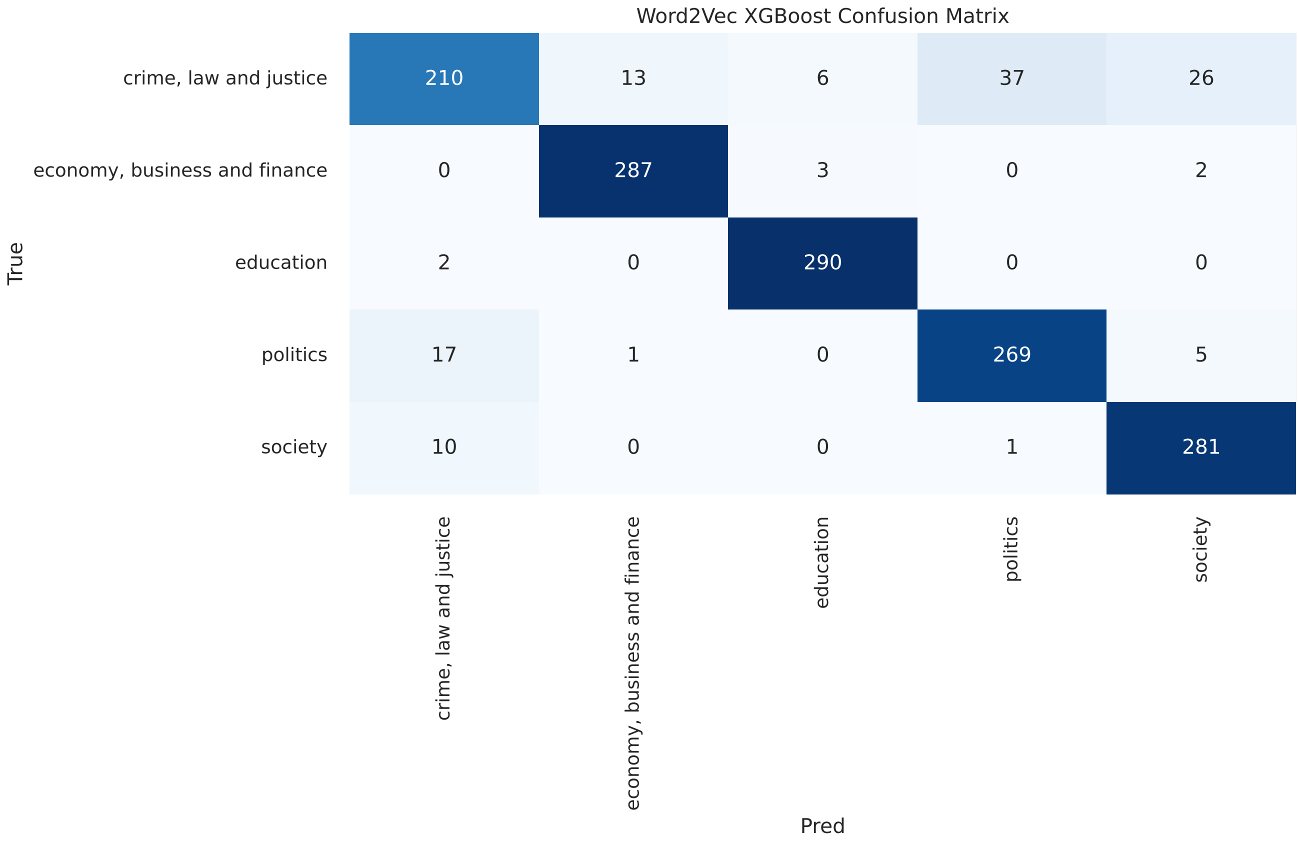


Figure 5.23: SMOTE and XGBoost-LIME Explanation for isiZulu Titles


Word2Vec XGBoost Confusion Matrix

True \ Pred	crime, law and justice	economy, business and finance	education	politics	society
crime, law and justice	210	13	6	37	26
economy, business and finance	0	287	3	0	2
education	2	0	290	0	0
politics	17	1	0	269	5
society	10	0	0	1	281

Figure 5.24: Word2vec-XGBoost Confusion Matrix for isiZulu Titles SMOTE Dataset

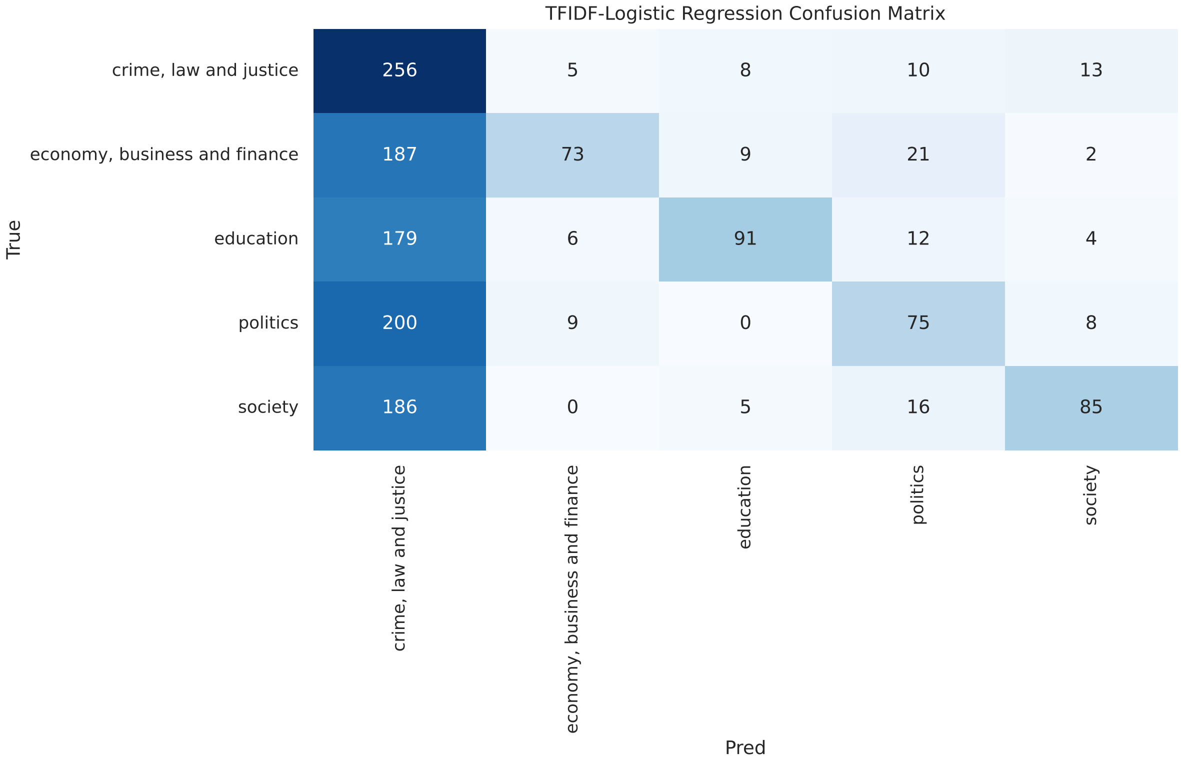


Figure 5.25: TFIDF-Logistic Regression Confusion Matrix for isiZulu Titles SMOTE Dataset

© University of Pretoria

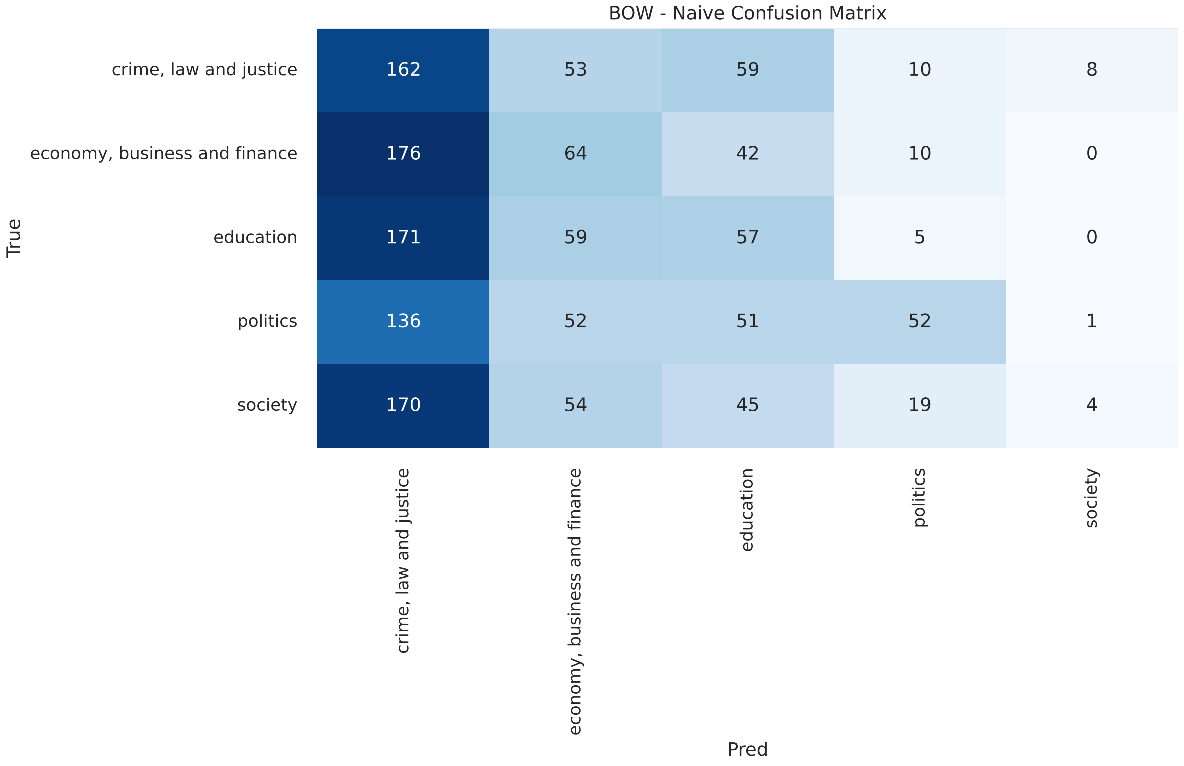


Figure 5.26: CountVectorizer-Naive Bayes Confusion Matrix for isiZulu Titles SMOTE Dataset

5.3.3 siSwati Titles Model Training

SMOTE was used to balance the class categories for siSwati Titles dataset then the models were trained, and therefore, the models performance results were recorded on Table 5.9 below. It was observed that Logistic Regression, XGBoost and Naive Bayes models trained from Word2vec produced good results obtaining over 80% on accuracy and f1 score, including Logistic Regression trained from TFIDF preprocessing.

The best performing model was found to be XGBoost model(Word2vec) with 88.75% accuracy and 87.46% f1-score, followed by Logistic Regression model(Word2vec) with 88.12% accuracy and 86.20% f1-score. On the other hand, the least performing model was found to be Naive Bayes(Count vectorizer) with 40% accuracy and 37.91% f1-score as shown on table 5.9

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Table 5.9: siSwati Titles SMOTE Dataset Model Performance

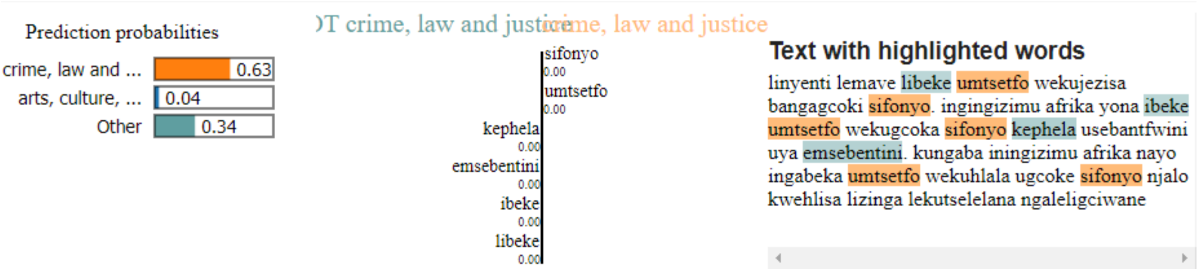
Preprocessing	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
Count Vectorize	Naive Bayes	60.63	40.67	37.91	40	(30.4,45.43)
Count Vectorize	Logistic Regression	65.03	44.19	42.91	44.38	(35.24,50.58)
Count Vectorize	XGBoost	81.3	74.38	73.65	74.38	(66.83,80.48)
TF-IDF 1-grams	Naive Bayes	80.71	79.14	74.32	78.75	(67.55,81.09)
TF-IDF 1-grams	Logistic Regression	82.25	82.95	80.42	83.12	(74.27,86.57)
TF-IDF 1-grams	XGBoost	85.47	77.05	76.6	76.88	(70.04,83.16)
Word2vec	Naive Bayes	85.86	83.71	82.5	83.75	(76.62,88.39)
Word2vec	Logistic Regression	**90.35**	88.1	86.2	88.12	(80.86,91.55)
Word2vec	XGBoost	89.88	**88.76**	**87.46**	**88.75**	(82.33,92.59)

The analysis on the decision making criteria that was used by the model to make a prediction was done using LIME model. After applying LIME model on document 3, it was found that the best performing model XGBoost model (Word2vec) correctly classified the document, as shown below, the predicted class is *crime, law and justice* and true class is *crime, law and justice*

- **document id:** 3
- **True Class:** *crime, law and justice*
- **Predicted class:** *crime, law and justice*

The words that the model used to make the above correct prediction are *’Sifonyo’* and *’umtsetfo’* as shown in Figure 5.27. It is clear that the model is good and was able to learn from the data, regardless of data size limitation. Furthermore, with the existing data size limitation, the model managed to correctly classify majority of the model as shown in confusion matrix in Figure 5.28. Logistic Regression model(TFIDF) performed fairly good obtaining 80.42% f1-score and confusion matrix in Figure 5.29 shows that the model correctly classified majority of the news titles, however, struggled to differentiate *art, culture, entertainment and media* from *human interest*. The confusion matrix in Figure 5.30 of the least performing model Naive Bayes (Count Vectorizer) confirms the poor performance of the model.

© University of Pretoria



Prediction probabilities

crime, law and ... 0.63
arts, culture, ... 0.04
Other 0.34

)T crime, law and justicecrime, law and justice

sifonyo 0.00
umtsetfo 0.00
kephela 0.00
emsebentini 0.00
ibeke 0.00
libeke 0.00

Text with highlighted words

linyenti lemave libeke umtsetfo wekujezisa bangagcoki sifonyo. ingingizimu afrika yona ibeke umtsetfo wekugcoka sifonyo kephela usebantfwini uya emsebentini. kungaba iningizimu afrika nayo ingabeka umtsetfo wekuhlala ugcoke sifonyo njalo kwehlisa lizinga lekutselelana ngaleligciwane

Figure 5.27: SMOTE and XGBoost-LIME Explanation for siSwati Titles

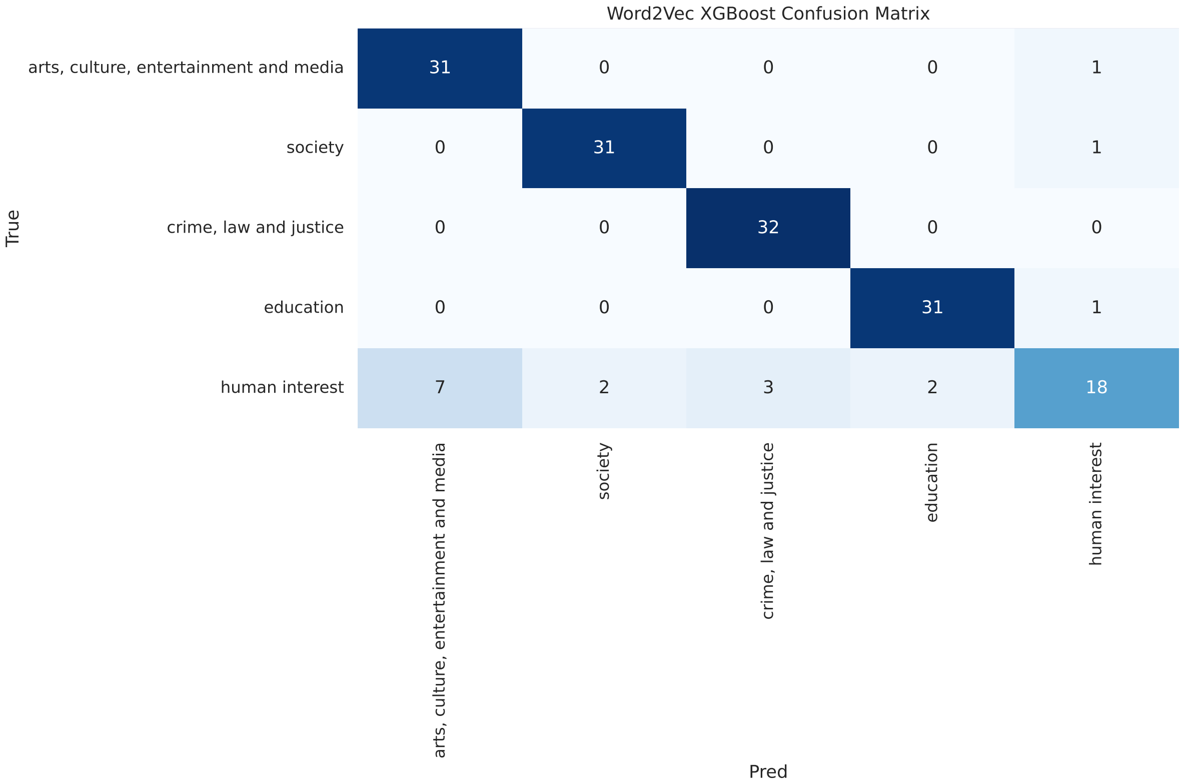


Figure 5.28: Word2vec-XGBoost Confusion Matrix for siSwati Titles SMOTE Dataset

© University of Pretoria

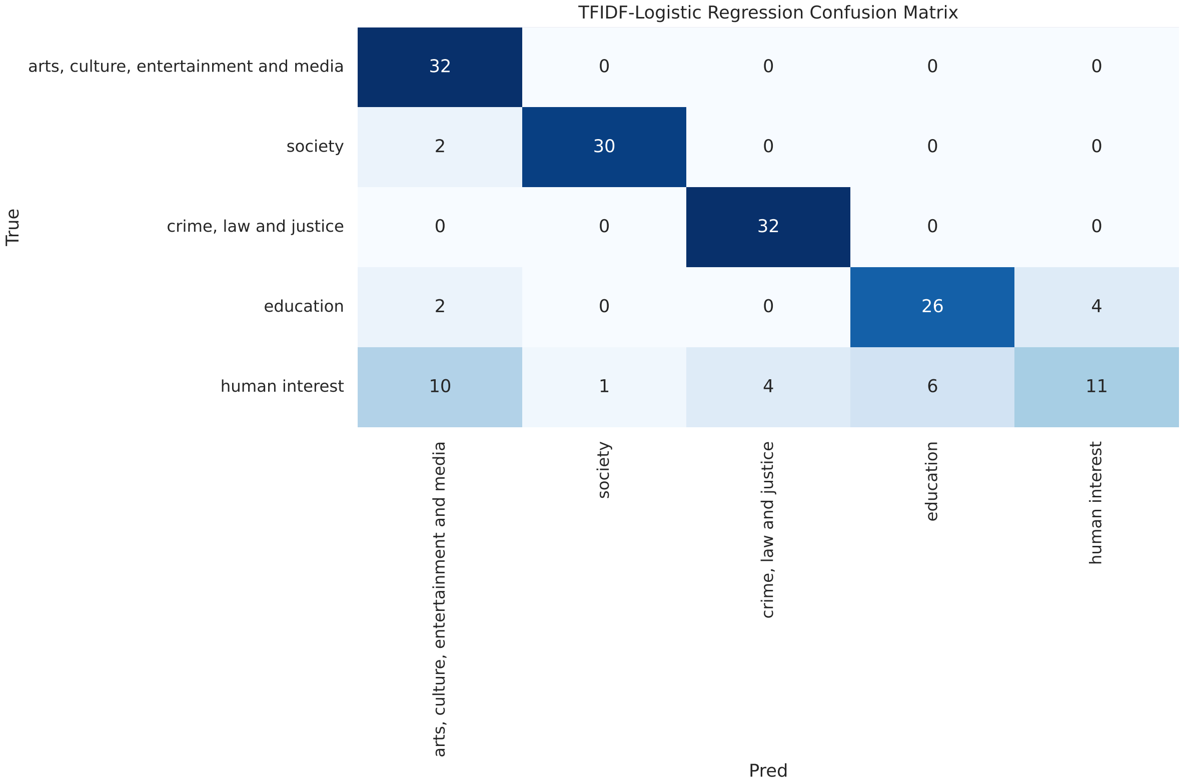


Figure 5.29: TFIDF-Logistic Regression Confusion Matrix for siSwati Titles SMOTE Dataset

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

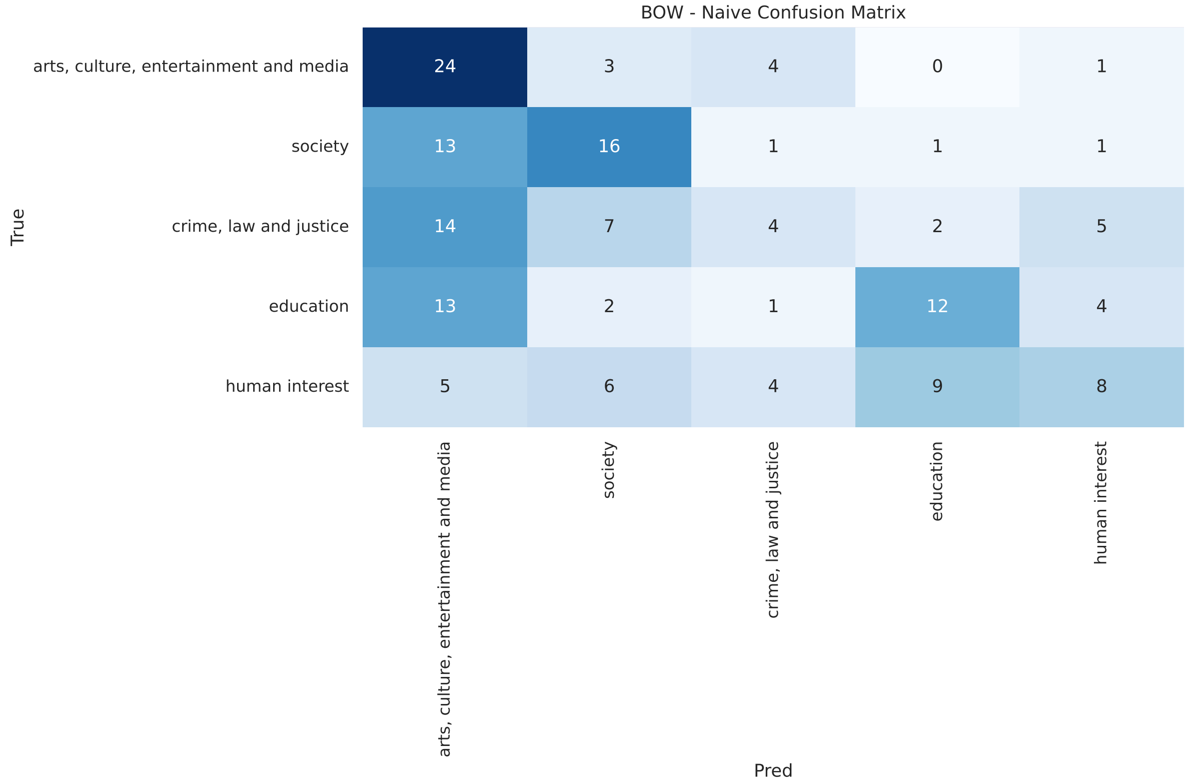


Figure 5.30: CountVectorizer-Naive Bayes Confusion Matrix for siSwati Titles SMOTE Dataset

5.4 Discussion Summary

The above section demonstrated the outcomes of the machine learning classification models together with the performance of each model on the isiZulu Articles, isiZulu Titles and siSwati Titles datasets (oversampled and augmented) coupled with each of the three word embedding, namely, Count vectorizer, TFIDF vectorizer and Word2vec. The findings from the experiment shows that for all the three datasets, the models trained with Word2vec performed way better in all instances as compared to models trained from Count and TFIDF vectorizers. Moreover, the models trained from TFIDF vectorizer outperformed models trained from Count vectorizer.

The models trained before the oversampling techniques were applied performed very bad except for LSTM model, LSTM model managed to obtain over 70% f1 score and accuracy on imbalanced dataset, while other classical models were struggling. However, the class category imbalance problem was then mitigated through applying the sampling techniques, namely, Contextual Data Augmentation and SMOTE then train the same models again. It was found that the models performance improved drastically, meaning that the oversampling techniques had positive impact on the models performance.

XGBoost(Word2vec) model outperformed all the models in many instances, except for two instances, that is, Augmented isiZulu Titles and siSwati Titles where Logistic regression(Word2vec) and LSTM(Word2vec) took the lead respectively as shown in table 5.10. Although XGBoost was outperformed in those two instances, the model still showed the ability to learn from the data, performed better and scored over 80% f1-score(which is slightly different from the best models). In the case of SMOTE, XGBoost outperformed all the models in all instances. Logistic Regression(Word2vec) outperformed all the models on isiZulu Titles augmented dataset whereas XGBoost(Count Vectorizer) was the least performing model on the same dataset, this is the only instance that Naive Bayes model was not the least performing model. Furthermore, LSTM model performed well on siSwati augmented dataset and isiZulu original imbalanced datasets(both isiZulu Articles and Titles) and lastly, Naive Bayes model produced poor results in most instances.

The classification models performed well on isiZulu Articles dataset possibly because the dataset size is large and contains long-texts as compared to isiZulu Titles and siSwati Titles datasets. isiZulu Titles dataset is large but contains very short texts, whereas siSwati Titles dataset is small, and the texts are short(but not as short as isiZulu Titles). However, it was observed that in the case of Contextual Data Augmentation technique, the best model performed better on siSwati Titles augmented dataset than on isiZulu Titles augmented dataset, whereas in the case of SMOTE, the model perfomance on siSwati Titles outperformed isiZulu Titles as shown on Table 5.10 in terms of scoring high f1-score. In conclusion, Contextual Data Augmentation performed better on large-size dataset containing short-text and SMOTE did well on small-size dataset containing short-text. In Summary, the best performing models for isiZulu Articles and siSwati Titles were obtained from augmented datasets and only for isiZulu Titles the best model was obtained from SMOTE dataset.

The highest accuracy and f1 score obtained from isiZulu Titles augmented dataset best performing model is 85.69% and 86.42% and for siSwati Titles is 92.41% and 93.14% respectively, whereas the best performing models trained using SMOTE dataset produced the highest accuracy and f1 score of 91.58% and 91.26% for isiZulu Titles and 88.75% and 87.46% for siSwati Titles as documented on Table 5.10, it was observed that SMOTE techniques scored the highest f1-score on isiZulu Titles dataset(very short text dataset) and Contextual Data Augmentation scored the highest f1-score on siSwati Titles(short text and small size dataset). This is possibly due to the difference between the two oversampling techniques, that is, SMOTE synthesizes the original observations to create slightly different new observations, with a possibility of sample-overlapping [87], which in turn makes it possible for the exact observations to be on both training and test sets during cross-validation split and those observations are obvious to predict since the model has seen them during training. This gives isiZulu Titles dataset an advantage since it is large and most of the replicated observations will be overlaping on both training and test sets, leading to many obvious predictions. Therefore, SMOTE did well on large size dataset containing short texts(isiZulu Titles) dataset possibly not because the models were predicting accurately on their own, however, could be that the models have seen most of the observations during training(easy to memorise). On the other hand, Contextual Data Augmentation creates new different(altered) observations from the original observations, then it is impossible for the exact observations to be on both training and test sets, hence provides fair model learning and prediction. However, study need to be carried out to provide more clarity on the above claim.

LIME model assisted with providing the interpretation of the prediction's decision process, hence, we were able to deep dive and see the model prediction's decision making process, and it was observed that best performing models were at some point struggling to identify important words from a text in order to make correct prediction although they managed to correctly classify majority of the documents.

The learning and explanation from LIME model made more sense in terms of the choice of influential words, the influential words derived from the models trained from SMOTE and Contextual Augmentation differ, for instance, XGBoost performed well on both isiZulu Articles SMOTE and Data Augmentation datasets, however, the Lime explanation for the two instances differ in terms of the choice of influential words, and in comparison, the model trained on Contextual Augmentation dataset provided more reasonable influential words.

The Pipeline followed in this study was summarised and presented in Figure 5.31 below, the Figure shows the choice that produced the best results under different circumstances for the three different datasets. It was observed that the datasets used resembled three different qualities, that is, large size and long-text (isiZulu Articles), large size and short text(isiZulu Titles), and small size and short text(siSwati), these varieties produced different outcomes from the models under the same circumstance and can be generalised as follows:

- If the data size is large and contains long-text then Contextual Data Augmentation is recommended over SMOTE, and LSTM is likely to perform better.
- If the data size is large and contains short-text then SMOTE is recommended over Contextual Data Augmentation, and XGBoost is likely to perform better.
- If the data size is small and contains short-text then Contextual Data Augmentation is recommended over SMOTE, and XGBoost is likely to perform better

The Above generalisation is limited to Word2vec word embedding since it is the one that produced outstanding results from all the datasets as compared to TFIDF and Count vectorizers. It remains a task to further investigate the poor performance from TFIDF and Count Vectorizers, possibly the parameter change in classification could lead to good results.

Table 5.10: Top Performing Classification Models

Best Model based on Sampling technique								
Dataset	**Sampling**	**Word embbeding**	**Model**	**Precision(%)**	**Recall(%)**	**F1-score(%)**	**Accuracy(%)**	**Confidence Interval(f1 score)**
isiZulu Articles	Augmented	Word2vec	XGBoost	95.54	95.73	95.21	95.14	(94.04,96.39)
isiZulu Titles	Augmented	Word2vec	Logistic Regression	86.35	87.65	86.42	85.69	(84.39,88.45)
siSwati Titles	Augmented	Word2vec	LSTM	94.88	92.41	93.15	92.41	(89.02,97.27)
isiZulu Articles	SMOTE	Word2vec	XGBoost	93.75	93.55	93.35	93.56	(92.08,94.63)
isiZulu Titles	SMOTE	Word2vec	XGBoost	91.96	91.56	91.26	91.58	(89.81,92.71)
siSwati Titles	SMOTE	Word2vec	XGBoost	89.88	88.76	87.46	88.75	(82.33,92.59)

© University of Pretoria

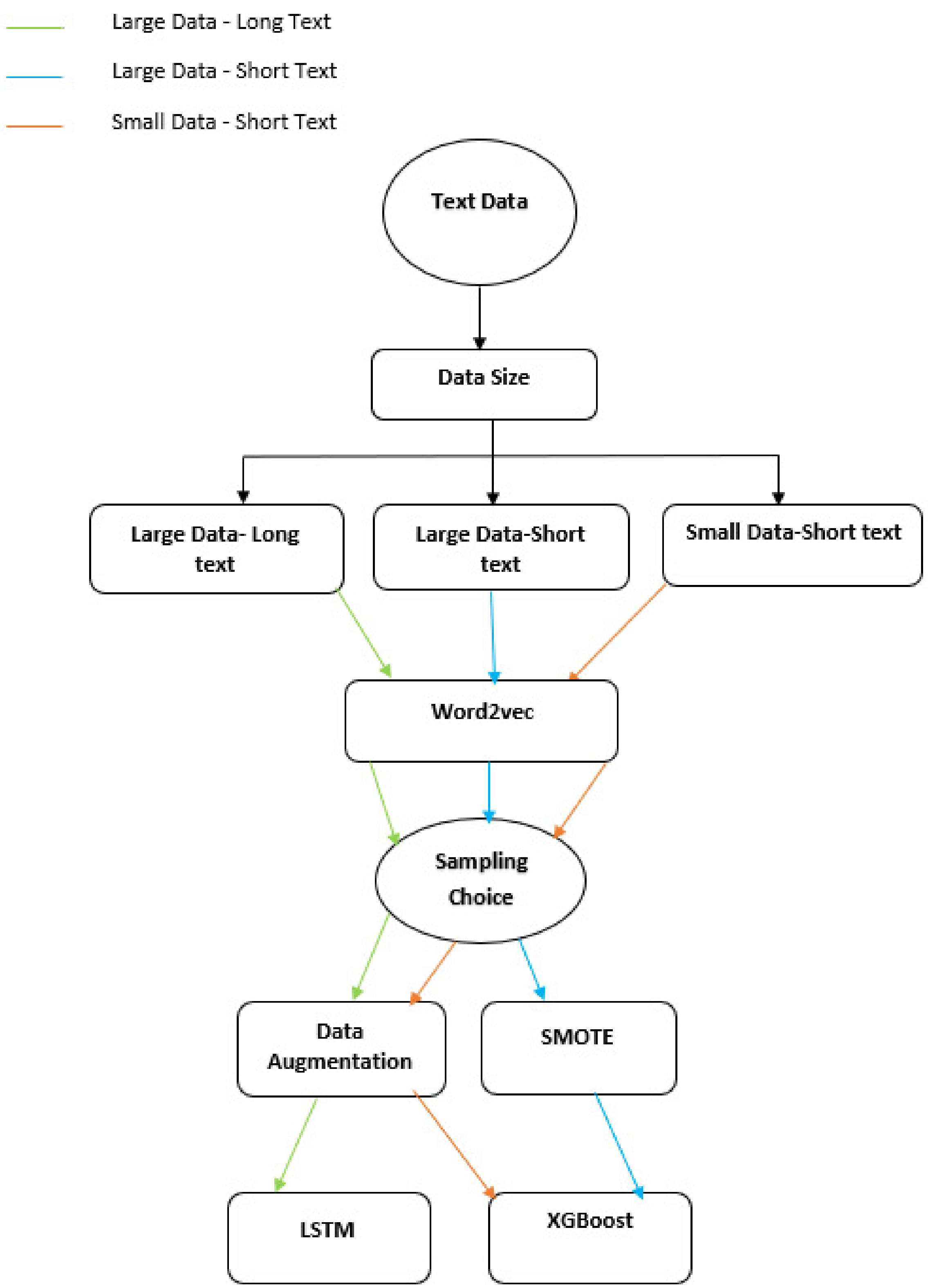


Figure 5.31: Recommended Pipeline

© University of Pretoria

The results obtained from the study conducted by [4] for Sepedi and Setswana languages showed that augmentation only improved the performance of classical models and reduced the performance of MLP Neural Network model (for TFIDF vectorizer). Moreover, XGBoost performed better for the case of augmentation for Sepedi languages and Logistic Regression for Setswana [4]. In this study, context Augmentation and SMOTE improved the performance of both classical and Neural Network (LSTM), however, this is for word2vec vectorizer. This shows that there is a great possibility that word2vec may increase the performance for MLP neural network for Setswana and Sepedi languages in the case of augmentation and outperform TFIDF vectorizer. It was also noted that XGBoost performed better in many instances for this study, which shows a potential for better performance in low resource languages.

© University of Pretoria

Chapter 6

Conclusion and future work

This work introduced the collection and annotation of isiZulu and siSwati news datasets. There is still a data shortage(more especially annotated data) of these two local languages, especially siSwati. However, this work paved a way for the other researchers who would want to use annotated data for isiZulu and/or siSwati in downstream NLP tasks.

The experimental findings from the classification models and different combinations of word embeddings with model baselines were presented. Though we were limited by the data availability, however, this provides an overview of what could be achieved with minimal datasets. The isiZulu and siSwati annotated datasets will be made available for other researchers, the pre-trained vectorizers will be open-sourced to other researchers and the classification results that may be used as benchmarks.

The collection and annotation of local language datasets remain a task for the future. Furthermore, NLP researchers need to focus more on effective ways to augment the datasets. They should be compared with SMOTE sampling, because of the imbalance in the dataset. It is beneficial to have effective ways to augment local datasets.

In addition, it is also worth investigating the poor performance of TFIDF and Count vectorizers compared to Word2vec, possible investigation areas could be the word embedding nature and the classification models hyperparameters that could improve classification performance. Another extension of this work is transfer learning from isiZulu to siSwati. The isiZulu dataset is large compared to the siSwati dataset. Therefore, we

© University of Pretoria

can leverage that and assess if transfer learning improves the classification performance for siSwati.

6.1 Summary

This chapter summarises all the work that has been carried out and outlines how the objective of the study is met. Further explained the possible future work that can be executed to continue from this work.

© University of Pretoria

Bibliography

[1] Priya Dialani. *What is NLP and Why is it Important?* May 2020. URL: https://www.analyticsinsight.net/what-is-nlp-and-why-is-it-important/#:~:text=Natural%20language%5C%2.

[2] Bernardt Duvenhage, Mfundo Ntini, and Phala Ramonyai. "Improved text language identification for the South African languages". In: *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*. IEEE. 2017, pp. 214–218.

[3] South Africa gateway. *The 11 languages of South Africa*. Aug. 2021. URL: https://southafrica-info.com/arts-culture/11-languages-south-africa/.

[4] Vukosi Marivate et al. "Investigating an approach for low resource language dataset creation, curation and classification: Setswana and Sepedi". In: *arXiv preprint arXiv:2003.04986* (2020).

[5] Zahurul Islam, Alexander Mehler, and Rashedur Rahman. "Text readability classification of textbooks of a low-resource language". In: *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*. 2012, pp. 545–553.

[6] Sebastian Ruder, Ivan Vulić, and Anders Sogaard. "A survey of cross-lingual word embedding models". In: *Journal of Artificial Intelligence Research* 65 (2019), pp. 569–631.

[7] Sebastian Ruder. *The 4 Biggest Open Problems in NLP*. Feb. 2020. URL: https://ruder.io/4-biggest-open-problems-in-nlp/.

© University of Pretoria

[8] Pratik Joshi et al. "Unsung challenges of building and deploying language technologies for low resource language communities". In: *arXiv preprint arXiv:1912.03457* (2019).

[9] Roald Eiselen and Martin J Puttkammer. "Developing Text Resources for Ten South African Languages." In: *LREC*. 2014, pp. 3698–3703.

[10] Peter Baumann and Janet Pierrehumbert. "Using resource-rich languages to improve morphological analysis of under-resourced languages". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. 2014, pp. 3355–3359.

[11] Bogusława Whyatt and Nataša Pavlović. *Languages of low diffusion and low resources: translation research and training challenges: Special Issue Proposal ITT–15 (1), March 2021*. 2019.

[12] Svanhvít Lilja Ingólfsdóttir et al. "Nefnir: A high accuracy lemmatizer for Icelandic". In: *arXiv preprint arXiv:1907.11907* (2019).

[13] Shereen Khoja. "APT: Arabic part-of-speech tagger". In: *Proceedings of the Student Workshop at NAACL*. Citeseer. 2001, pp. 20–25.

[14] Ping Xu and Pascale Fung. "Cross-lingual language modeling for low-resource speech recognition". In: *IEEE transactions on audio, speech, and language processing* 21.6 (2013), pp. 1134–1144.

[15] Hamed Bonab, James Allan, and Ramesh Sitaraman. "Simulating CLIR translation resource scarcity using high-resource languages". In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 2019, pp. 129–136.

[16] Sonja Bosch, Laurette Pretorius, and Axel Fleisch. "Experimental bootstrapping of morphological analysers for Nguni languages". In: *Nordic Journal of African Studies* 17.2 (2008), pp. 23–23.

© University of Pretoria

[17] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. "Data quality from crowdsourcing: a study of annotation selection criteria". In: *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*. 2009, pp. 27–35.

[18] Pontus Stenetorp et al. "BRAT: a web-based tool for NLP-assisted text annotation". In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 2012, pp. 102–107.

[19] Natasha Latysheva. *Why do we use word embeddings in NLP?* Sept. 2019. URL: https://towardsdatascience.com/why-do-we-use-embeddings-in-nlp-2f20e1b632d2.

[20] Tom Kenter and Maarten De Rijke. "Short text similarity with word embeddings". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, pp. 1411–1420.

[21] Omer Levy and Yoav Goldberg. "Dependency-based word embeddings". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014, pp. 302–308.

[22] Anna Gladkova and Aleksandr Drozd. "Intrinsic evaluations of word embeddings: What can we do better?" In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. 2016, pp. 36–42.

[23] Guido Zuccon et al. "Integrating and evaluating neural word embeddings in information retrieval". In: *Proceedings of the 20th Australasian document computing symposium*. 2015, pp. 1–8.

[24] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. "Support vector machines and word2vec for text classification with semantic features". In: *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE. 2015, pp. 136–140.

[25] M Ikonomakis, Sotiris Kotsiantis, and V Tampakas. "Text classification using machine learning techniques." In: *WSEAS transactions on computers* 4.8 (2005), pp. 966–974.

© University of Pretoria

[26] Ciro Donalek. “Supervised and unsupervised learning”. In: *Astronomy Colloquia. USA*. Vol. 27. 2011.

[27] Rubungo Andre Niyongabo et al. “KINNEWS and KIRNEWS: Benchmarking cross-lingual text classification for Kinyarwanda and Kirundi”. In: *arXiv preprint arXiv:2010.12174* (2020).

[28] Joel Nothman, Hanmin Qin, and Roman Yurchak. “Stop word lists in free open-source software packages”. In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. 2018, pp. 7–12.

[29] Rajnish M Rakholia and Jatinderkumar R Saini. “Lexical classes based stop words categorization for Gujarati language”. In: *2016 2nd international conference on advances in computing, communication, & automation (ICACCA)(Fall)*. IEEE. 2016, pp. 1–5.

[30] Batta Mahesh. “Machine Learning Algorithms-A Review”. In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386.

[31] Davide Chicco and Giuseppe Jurman. “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. In: *BMC genomics* 21.1 (2020), pp. 1–13.

[32] Dell Zhang, Jun Wang, and Xiaoxue Zhao. “Estimating the uncertainty of average F1 scores”. In: *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. 2015, pp. 317–320.

[33] Mohammad Hossin and Md Nasir Sulaiman. “A review on evaluation metrics for data classification evaluations”. In: *International journal of data mining & knowledge management process* 5.2 (2015), p. 1.

[34] Terrance Liu. “Optimizing BLEU Scores for Improving Text Generation”. In: (2019).

[35] Kishore Papineni et al. “Bleu: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

© University of Pretoria

[36] Xin Tang et al. “Improving multilingual semantic textual similarity with shared sentence encoder for low-resource languages”. In: *arXiv preprint arXiv:1810.08740* (2018).

[37] Mehrnoush Shamsfard. “Challenges and Opportunities in Processing Low Resource Languages: A study on Persian”. In: ().

[38] Huu-Thanh Duong and Tram-Anh Nguyen-Thi. “A review: preprocessing techniques and data augmentation for sentiment analysis”. In: *Computational Social Networks* 8.1 (2021), pp. 1–16.

[39] Hugo Queiroz Abonizio and Sylvio Barbon Junior. “Pre-trained Data Augmentation for Text Classification”. In: *Brazilian Conference on Intelligent Systems*. Springer. 2020, pp. 551–565.

[40] Sosuke Kobayashi. “Contextual augmentation: Data augmentation by words with paradigmatic relations”. In: *arXiv preprint arXiv:1805.06201* (2018).

[41] Georgios Rizos, Konstantin Hemker, and Björn Schuller. “Augment to prevent: short-text data augmentation in deep learning for hate-speech classification”. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2019, pp. 991–1000.

[42] Alberto Fernández et al. “SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary”. In: *Journal of artificial intelligence research* 61 (2018), pp. 863–905.

[43] Vaibhav Rupapara et al. “Impact of SMOTE on imbalanced text features for toxic comments classification using RVVC model”. In: *IEEE Access* 9 (2021), pp. 78621–78634.

[44] Meng Fang and Trevor Cohn. “Model transfer for tagging low-resource languages using a bilingual dictionary”. In: *arXiv preprint arXiv:1705.00424* (2017).

[45] Barret Zoph et al. “Transfer learning for low-resource neural machine translation”. In: *arXiv preprint arXiv:1604.02201* (2016).

© University of Pretoria

[46] Toan Q Nguyen and David Chiang. “Transfer learning across low-resource, related languages for neural machine translation”. In: *arXiv preprint arXiv:1708.09803* (2017).

[47] Stuart Mesham et al. “Low-Resource Language Modelling of South African Languages”. In: *arXiv preprint arXiv:2104.00772* (2021).

[48] Evander Nyoni and Bruce A Bassett. “Low-Resource Neural Machine Translation for Southern African Languages”. In: *arXiv preprint arXiv:2104.00366* (2021).

[49] Jan Van den Broeck et al. “Data cleaning: detecting, diagnosing, and editing data abnormalities”. In: *PLoS medicine* 2.10 (2005), e267.

[50] Erhard Rahm and Hong Hai Do. “Data cleaning: Problems and current approaches”. In: *IEEE Data Eng. Bull.* 23.4 (2000), pp. 3–13.

[51] Xu Chu et al. “Data cleaning: Overview and emerging challenges”. In: *Proceedings of the 2016 international conference on management of data*. 2016, pp. 2201–2206.

[52] Hassan Saif et al. “On stopwords, filtering and data sparsity for sentiment analysis of twitter”. In: (2014).

[53] Bao Guo et al. “Improving text classification with weighted word embeddings via a multi-channel TextCNN model”. In: *Neurocomputing* 363 (2019), pp. 366–374.

[54] Qian Liu et al. “Task-oriented word embedding for text classification”. In: *Proceedings of the 27th international conference on computational linguistics*. 2018, pp. 2023–2032.

[55] Andreas C Müller and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists.* ” O’Reilly Media, Inc.”, 2016.

[56] *Understanding TF-ID: A Simple Introduction.* May 2019. URL: https://monkeylearn.com/blog/what-is-tf-idf/.

[57] Vatsal. *Word2Vec Explained.* Nov. 2021. URL: https://towardsdatascience.com/word2vec-explained-49c52b4ccb71.

[58] Tejas Menon. “Empirical Analysis of CBOW and Skip Gram NLP Models”. In: (2020).

© University of Pretoria

[59] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[60] Cheolhwan Oh, Seungmin Han, and Jongpil Jeong. "Time-series Data Augmentation based on Interpolation". In: *Procedia Computer Science* 175 (2020), pp. 64–71.

[61] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems* 28 (2015), pp. 649–657.

[62] Muhammed Kürşad Uçar et al. "The effect of training and testing process on machine learning in biomedical datasets". In: *Mathematical Problems in Engineering* 2020 (2020).

[63] Davide Anguita et al. "The 'K'in K-fold cross validation". In: *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. i6doc. com publ. 2012, pp. 441–446.

[64] Taiwo Oladipupo Ayodele. "Types of machine learning algorithms". In: *New advances in machine learning* 3 (2010), pp. 19–48.

[65] Reza Drikvandi and Olamide Lawal. "Sparse principal component analysis for natural language processing". In: *Annals of Data Science* (2020), pp. 1–17.

[66] Sasan Karamizadeh et al. "An overview of principal component analysis". In: *Journal of Signal and Information Processing* 4.3B (2013), p. 173.

[67] Patrik O Hoyer. "Non-negative matrix factorization with sparseness constraints." In: *Journal of machine learning research* 5.9 (2004).

[68] Ismail Bin Mohamad and Dauda Usman. "Standardization and its effects on K-means clustering algorithm". In: *Research Journal of Applied Sciences, Engineering and Technology* 6.17 (2013), pp. 3299–3303.

[69] David Sculley. "Web-scale k-means clustering". In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 1177–1178.

© University of Pretoria

[70] Purnima Bholowalia and Arvind Kumar. "EBK-means: A clustering technique based on elbow method and k-means in WSN". In: *International Journal of Computer Applications* 105.9 (2014).

[71] Xu Wang and Yusheng Xu. "An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 569. 5. IOP Publishing. 2019, p. 052024.

[72] Madhu Yedla, Srinivasa Rao Pathakota, and TM Srinivasa. "Enhancing K-means clustering algorithm with improved initial center". In: *International Journal of computer science and information technologies* 1.2 (2010), pp. 121–125.

[73] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013.

[74] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[75] Clare Liu. *Linear to Logistic Regression, Explained Step by Step*. URL: https://www.kdnuggets.com/2020/03/linear-logistic-regression-explained.html.

[76] Vlado Keselj. *Speech and Language Processing Daniel Jurafsky and James H. Martin (Stanford University and University of Colorado at Boulder) Pearson Prentice Hall, 2009, xxxi+ 988 pp; hardbound, ISBN 978-0-13-187321-6, $115.00*. 2009.

[77] *Introduction to Boosted Trees* . URL: https://xgboost.readthedocs.io/en/latest/tutorials/model.html.

[78] Jason Brownlee. *What is Deep Learning?* Aug. 2020. URL: https://machinelearningmastery.com/what-is-deep-learning/.

[79] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*. Vol. 1. MIT press Massachusetts, USA: 2017.

[80] Kazuya Kawakami. "Supervised sequence labelling with recurrent neural networks". In: *Ph. D. thesis* (2008).

[81] Hasim Sak, Andrew W Senior, and Françoise Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: (2014).

© University of Pretoria

[82] Guy S Handelman et al. “Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods”. In: *American Journal of Roentgenology* 212.1 (2019), pp. 38–43.

[83] Sofia Visa et al. “Confusion matrix-based feature selection.” In: *MAICS* 710 (2011), pp. 120–127.

[84] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Model-agnostic interpretability of machine learning”. In: *arXiv preprint arXiv:1606.05386* (2016).

[85] Jürgen Dieber and Sabrina Kirrane. “Why model why? Assessing the strengths and limitations of LIME”. In: *arXiv preprint arXiv:2012.00093* (2020).

[86] Kevin P Murphy et al. “Naive bayes classifiers”. In: *University of British Columbia* 18.60 (2006), pp. 1–8.

[87] Z Jiang et al. *A New Oversampling Method Based on the Classification Contribution Degree. Symmetry 2021, 13, 194.* 2021.

© University of Pretoria

Appendix A

Data Statement

Data Statement for the IsiZulu news (Articles and Headlines) and siSwati news Headlines Corpora

Dataset Name:

- IsiZulu news Articles and headlines
- siSwati news headlines

Citation: Pending

Link to dataset: Pending

Data set Developer(s): Andani Madodonga

Data statement author(s): Andani Madodonga

Collaborators: Live Languages, Prof Marivate, Dr Matthews

A. CURATION RATIONALE

Our data collection process for both isizulu and siSwati news datasets included scrapping the data from internet, from Isoleswe website(http://www.isolezwe.co.za) and SABC news LigwalagwalaFM Facebook page(https://www.facebook.com/ligwalagwalafm/) respectively and save them on CSV files. The datasets contain the isiZulu news article, isiZulu news headlines, and siSwati news headline. The datasets contained

© University of Pretoria

special characters and characters that are not ASCII encoded, however, special characters were removed and the other characters were decoded back to ASCII. In addition, isiZulu dataset contains only isiZulu texts whereas siSwati has some English words like 'Video', 'Live' etc that were removed from the dataset. The aim of these three datasets is to create a baseline classification models for the two south African low resource languages i.e isiZulu and siSwati. The Datasets were annotated, the vectorizers were built using the data from Sadilar(https://www.sadilar.org/) and Leipzig (https://wortschatz.uni-leipzig.de/), therefore, the classification models were trained and the performances were recorded for comparison.

B. LANGUAGE VARIETY

All news articles and headlines in isiZulu datasets are written in isiZulu language and all news headlines in siSwati dataset are written in siSwati language.

C. SPEAKER DEMOGRAPHIC

For both the isiZulu and siSwati datasets, we don't have the authors information since the datasets are from the online news reporters. All the datasets are composed of the local news, usually, the incident and updates about the things happening in south Africa.

D. ANNOTATOR DEMOGRAPHY

The isiZulu and siSwati datasets were annotated in 2020 by the isiZulu and siSwati linguistic experts from a private annotation company called Live Languages. Each article/headline was annotated by three linguistic experts.

E. SPEECH SITUATION

- The articles and headlines in the isiZulu news dataset were published during the year 2016 and 2020
- The headlines in the Siswati news dataset were published during the 2019 and 2020
- The intended audience are the isiZulu and Siswati news readers of all ages.

F. TEXT CHARACTERISTICS

Most of the news articles in the isiZulu corpus are from 'Crime, Law and Justice', 'Politics', and 'Society'. Most of the Siswati news headlines are from 'Society', 'arts, culture, entertainment and media' and 'human interest'

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

G. PROVENANCE APPENDIX

The isiZulu news articles, and headlines datasets were scrapped from isoleswe website (http://www.isolezwe.co.za) and the Siswati news headlines dataset was scrapped from SABC news LigwalagwalaFM Facebook page(https://www.facebook.com/ligwalagwalafm/). The datasets used to create the vectorizers for both isiZulu and Siswati were downloaded from Sadilar(https://www.sadilar.org/) and Leipzig(https://wortschatz.uni-leipzig.de/)

© University of Pretoria