JCT at SemEval-2023 Tasks 12A and 12B: Sentiment Analysis for Tweets Written in Low-resource African Languages using Various Machine Learning and Deep Learning Methods, Resampling, and HyperParameter Tuning

Ron Keinan, Yaakov HaCohen-Kerner

Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
21 Havaad Haleumi St., P.O.B. 16031, 9116001 Jerusalem, Israel
ronke21@gmail.com, kerner@jct.ac.il

Abstract

In this paper, we describe our submissions to the SemEval-2023 contest. We tackled subtask 12 - "AfriSenti-SemEval: Sentiment Analysis for Low-resource African Languages using Twitter Dataset". We developed different models for 12 African languages and a 13th model for a multilingual dataset built from these 12 languages. We applied a wide variety of word and char n-grams based on their tf-idf values, 4 classical machine learning methods, 2 deep learning methods, and 3 oversampling methods. We used 12 sentiment lexicons and applied extensive hyperparameter tuning.

1 Introduction

Sentiment analysis (SA), which is the computational determination of the sentiment expressed in a text, has become an increasingly important task in the field of Natural Language Processing (NLP). With the rapid growth of social media platforms, SA has been used to monitor and understand public opinion, classify sentiment, and even mitigate the spread of offensive language.

Despite the challenges of SA, such as the variability and subjectivity of sentiment expressions, researchers have been developing and improving SA systems to accurately determine the sentiment expressed in a text.

SA is a difficult task for computers and humans. To perform it in High-quality, this task requires understanding the context of the situation, the relevant culture, and in some cases the specific issue or people involved in this situation (Maynard and Greenwood, 2014).

The noisy nature of social media texts in general, and on tweeter in particular, makes the analysis task even harder. Therefore, detecting sentiment using supervised machine learning (ML), deep learning (DL) methods and natural language processing (NLP) tools is an interesting and challenging task.

SA in low-resource African languages is an even more challenging task due to the limited availability of annotated data, the variability of sentiment expressions, and the lack of NLP tools and resources

In this paper, we describe our research and participation in (1) subtask 12-A, for SA in tweets written in 12 languages: Hausa, Yoruba, Amharic, Swahili, Darija, Algerian, Nigerian Pidgin, Igbo, Xitsonga, Twi, Kinyarwandan, and Mozambique Portuguese, and (2) in subtask 12-B for SA in a multilingual tweet corpus, with tweets from all the 12 African languages mentioned above. The full description of task 12 in general and the subtasks, in particular, is given in Muhammad et al. (2023A), and the dataset is described in Muhammad et al. (2023B).

The structure of the rest of the paper is as follows. Section 2 introduces a background concerning SA in general and particularly in low-resource African languages, text classification (TC) with imbalanced classes, sentiment lexicons, and hyperparameter tuning. Section 3 describes subtasks 12-A, 12-B, and their datasets. In Sections 4 and 5, we present the submitted models and their experimental results. Section 6 summarizes and suggests ideas for future research. The appendix presents (1) the details of the training, dev, and test sets, (2) our competition best-submitted results, and (3) the sentiment lexicons that we used.

2 Background

2.1 Sentiment Analysis

Automatic SA aims to automatically identify, extract, and analyze subjective information in natural language text, to determine the author's

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 365–378

viewpoint on a particular subject. Many SA studies address marketing tasks, such as extracting opinions from customer reviews (Kiritchenko et al., 2014). At the same time, there is an increasing interest in the SA of the social web. SA enables to revealing of people's opinions about specific topics and to perform analysis to plan future actions. A wide variety of studies have been performed concerning SA of posts in various social forums such as blogs, Facebook, and Twitter. Tsytsarau and Palpanas (2012) reviewed the development of SA and opinion mining in recent years. They supplied an overview of the most popular sentiment extraction algorithms and an increasing drive towards more sophisticated algorithms.

In sentiment classification, two main techniques have been proposed: ML and Dictionary methods. Our research employed both techniques for comparison. The ML approach is composed of two general steps: (1) learn the model from a training corpus, and (2) classify a test corpus based on the trained model (Pang et al. 2002; Jeonghee et al. 2003). Various ML methods have been applied for sentiment classification. For instance, Pang and Lee (2005) applied three ML methods: Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM), and combined SVM and regression (SVR) modes, with metric labeling. Moraes et al. (2013) empirically compared SVM and ANN for document-level sentiment classification. HaCohen-Kerner et al. (2019A) applied four ML methods: Bayes Networks, SimpleLogistic, SMO, and Random Forest (RF).

2.2 Sentiment Analysis in Low Resources African Languages

SA in low-resource African languages is challenging for several factors. One of the major challenges is the limited amount of annotated data available for these languages. In SA, annotated data refers to text data that has been labeled with the sentiment expressed in the text, such as positive, negative, or neutral. This data is used to train ML algorithms for SA. However, the scarcity of annotated data for low-resource languages makes it difficult to train high-quality SA systems. This is because ML algorithms require a large amount of data to learn patterns and make accurate predictions. As a result, SA systems trained on a limited amount of data for low-resource African languages tend to have lower performance and are less accurate.

Another challenge is the variability of sentiment expressions in low-resource African languages. Unlike English, for example, many African languages have a rich expression of emotions, making it difficult to accurately determine the sentiment. Additionally, cultural factors play a significant role in determining the sentiment expressed in a text, adding to the complexity of the task.

Finally, the low availability of NLP tools and resources for low-resource African languages, such as text-to-speech and machine translation systems, makes it difficult to pre-process text data and prepare it for SA.

Muhammad et al. (2022) conducted extensive research to build a wide database of 4 resource-poor African languages and provided original and significant resources that we used such as the stopwords database as well as sentiment dictionaries in Nigerian languages. Earlier research (Yimam et al., 2020) recognized the challenges of performing SA in Amharic, a low-resource language. The authors built a SA dataset in Amharic, annotated it, and tested different types of word embeddings and ML models to classify the sentiment of each tweet.

Kelechi et al. (2021) attempted to train a multilingual language model on only low-resource African languages. Their model, named AfriBERTa, covers 11 African languages, including the first language model for 4 of these languages. Evaluations on TC spanning 10 languages show that their model outperforms mBERT[1] and XLM-Roberta[2] and it is very competitive overall.

Dossou et al. (2022) presented AfroLM, a multilingual language model trained from scratch on 23 African languages using a self-active learning framework. According to their research, AfroLM outperforms many multilingual pre-trained language models (AfriBERTa, XLM-Roberta-base, mBERT) on various NLP downstream tasks (NER, TC, and SA).

2.3 Text Preprocessing

Text preprocessing is crucial in NLP fields such as ML, SA, text mining, and TC. In both general and social text documents, noise such as typos, emojis, slang, HTML tags, spelling mistakes, and repetitive letters often appear. Improperly

[1] https://github.com/google-research/bert/blob/master/multilingual.md

[2] https://huggingface.co/docs/transformers/model_doc/xlm-roberta

preprocessed text can result in incorrect analysis outcomes. In many cases, the application of preprocessing methods is considered effective for TC tasks. For instance, HaCohen-Kerner et al. (2008) demonstrated that using word unigrams including stop words leads to improved results compared to the results obtained using word unigrams excluding stop words.

HaCohen-Kerner et al. (2019B, 2020) investigated the impact of all possible combinations of six preprocessing methods (punctuation mark removal, reduction of repeated characters, spelling correction, HTML tag removal, converting uppercase letters into lowercase letters, and stopword removal) on TC in three datasets. The main conclusion recommended is always to perform an extensive and systematic variety of preprocessing methods, combined with many ML methods to improve the accuracy of TC.

As part of the competition, we made a comparison to find the pre-processing process that improves the results of the models the most, as detailed at the end of Chapter 3.

2.4 Text Classification with Imbalanced Classes

Classification of texts with imbalanced classes is a problem where there are too few examples of the minority class to effectively learn a good predictive model. The main idea to solve this problem is to change the dataset to reach a more balanced distribution. Two popular sampling methods enable such a change: oversampling and undersampling. Oversampling means duplicating examples in the minority class. Undersampling means deleting examples in the majority class. The two basic versions of these methods do over/under-sampling randomly.

Another frequent method is to generate synthetic samples which means randomly sampling the attributes from instances in the minority class. The most popular algorithm that supports the generation of synthetic samples is called SMOTE, Synthetic Minority Oversampling Technique (Chawla, 2002). This method is an oversampling method that instead of creating copies, creates synthetic samples from the minor class. This method selects at least 2 similar instances and perturbs an instance one attribute at a time by a random number within the difference to the similar instances. More solutions to TC with imbalanced classes are referred to in the following articles (Chawla et al., 2002; He and Ma, 2013; Krawczyk, 2016; Brownlee, 2020, and Shaikh et al., 2021).

Since most of the datasets were not balanced, we compared the different balancing processes to find the ideal balance in each language, as detailed in Table 4.

2.5 Sentiment Lexicons

A sentiment lexicon is a list of positive and negative words and phrases. Each word or phrase has a positive or negative score reflecting its sentiment polarity. For example, words like "love", "wonderful", and "delightful" might have a strong positive sentiment score, while words like "hate", "disgusting", and "terrible" might have a strong negative sentiment score (Pang & Lee, 2008). The coverage and the quality of a sentiment lexicon are and may contribute to the success of various tasks like opinion mining and SA (Liu, 2012; Feldman, 2013, Yang et al., 2020).

Since SA involves determining the emotional tone or attitude expressed in a piece of text, and sentiment lexicons provide a pre-defined set of words and their corresponding sentiments, lexicons can be used as features for TC models.

Sentiment lexicons can be created through various methods, including manual annotation, crowdsourcing, and NLP techniques. Each word in a sentiment lexicon is assigned a sentiment score, which represents its strength of association with a particular sentiment.

Sentiment lexicons are widely used in TC because they provide a quick and efficient way to extract relevant features from text data (Chiong et al., 2021). In SA, these lexicons can be used to identify the overall sentiment of a text document by calculating the sum of the sentiment scores of the document's words.

The issue of negation is a crucial aspect of sentiment analysis that needs to be taken into account when using sentiment lexicons. Negation refers to the use of words that change the meaning of a sentence to its opposite, such as "not," "no," and "never." Negation can completely flip the sentiment of a sentence and affect the accuracy of sentiment analysis. Some sentiment lexicons include special notation for negation, and advanced natural language processing techniques may also be used to better account for negation and other linguistic nuances in a text.

The VADER (Valence Aware Dictionary and Sentiment Reasoner) lexicon (Hutto & Gilbert, 2014) is a popular tool for SA systems. The VADER lexicon is designed specifically for SA

in social media and works well for texts with informal language, sarcasm, and emoticons. The VADER lexicon is a dictionary of words and phrases along with their sentiment scores, which range from negative (-1) to positive (+1).

As part of the competition, we combined sentiment dictionaries for all languages. The results of the combinations and their contribution are detailed in Table 4. The structure and source of each dictionary are detailed in Appendix E.

2.6 HyperParameter Tuning

Hyperparameter tuning is a process in ML that involves selecting the best set of hyperparameters for a given model. The goal of hyperparameter tuning is to find a set of hyperparameters that result in a model that generalizes well to unseen data (Bardenet et al., 2013).

Hyperparameters are parameters that are set before training an ML model, and they can have a significant impact on the performance of the model. Common examples of hyperparameters include the learning rate in a neural network, the number of trees in an RF, and the regularization coefficient in a linear regression model.

One of the most common approaches to hyperparameter tuning is grid search (Bergstra & Benjio, 2012). Grid search is a brute force method that involves exhaustively testing all combinations of hyperparameters within a specified range.

As part of our research for the competition, we adjusted parameters as detailed in Chapter 4.

2.7 Task and Datasets Description

The AfriSenti-SemEval Shared Task 12 is based on a collection of Twitter datasets in 14 African languages for sentiment classification (Muhammad et al., 2023A). The dataset involves tweets labeled with three sentiment classes (positive, negative, and neutral). Each tweet is annotated by three annotators (Muhammad et al., 2023B).

Task A is a monolingual SA task. Given training data in a target language, the mission is to determine the polarity of a tweet in the target language (positive, negative, or neutral). This sub-task has 12 tracks of different languages: Hausa, Yoruba, Igbo, Nigerian Pidgin from Nigeria, Amharic, from Ethiopia, Swahili from Kenya and Tanzania, Algerian Arabic dialect from Algeria, Kinyarwanda from Rwanda, Twi from Ghana, Mozambique Portuguese and Xitsonga from Mozambique, and Moroccan Arabic/Darija from Morocco.

Task B is a multilingual sentiment classification task. Given combined training data from Task-A (Tracks 1 to 12), determine the polarity of a tweet in the target language.

Appendices A-C presents sentiment details about the training, dev, and test sets for all the languages. The analysis of the details presented in Appendices A and B show that the training, dev, and test sets for most of the languages are highly imbalanced, with ratios like 47:35:18 or 30:10:60 for positive: negative: neutral tweets. We made a few efforts to balance the dataset in our experiments using the sampling methods that we have mentioned above.

3 System Overview

We created feature matrices from the datasets, while the features were word unigrams and char n-grams in ranges of 3-10 from the training dataset with the highest TF-IDF values.

We enriched the feature matrices with more words from the open-source sentiment lexicons that we found for all the languages. For Xitsonga, we did not find a lexicon, so we translated the Vader lexicon with Google Translate.

We applied 4 supervised ML methods on the training datasets' feature matrices: RF, Support Vector Classifier (SVC), Logistic regression (LR), and Multinomial Naive Bayes (MNB). We also applied 2 supervised DL methods: Multi-layer Perceptron (MLP) and Neural Network (NN) with BERT embeddings.

RF is an ensemble learning algorithm that is used for classification and regression problems. (Breiman, 2001). Ensemble methods use multiple learning algorithms to obtain improved predictive performance compared to what can be obtained from any of the constituent learning algorithms. RF combines multiple decision trees to form a forest of trees, and the final prediction is made by taking a majority vote of the trees. RF combines Breiman's “bagging” (Bootstrap aggregating) idea in Breiman (1996) and a random selection of features introduced by Ho (1995) to construct a forest of decision trees

SVC is a variant of the SVM ML method (Cortes and Vapnik, 1995) implemented in SkLearn. SVC uses LibSVM (Chang & Lin, 2011), which is a fast implementation of the SVM method. SVM is a supervised learning algorithm that is used for classification and regression analysis. It works by finding the hyperplane that

maximally separates the data into classes. SVM is known for its good generalization ability and its ability to handle non-linearly separable data using the kernel trick.

LR (Cox, 1958; Hosmer et al., 2013) is a supervised learning algorithm that is used for binary and multiclass classification problems. It models the relationship between the dependent variable and the independent variables using a logistic function. It is known also as maximum entropy regression (MaxEnt), logit regression, and the log-linear classifier.

Multinomial Naive Bayes (MNB) is a statistical machine learning algorithm based on the Bayes theorem (Kim et al., 2006). MNB assumes that features are conditionally independent given the target class, estimates the probabilities of each class and the probabilities of each feature given the class, and uses these probabilities to make predictions.

Multilayer Perceptron (MLP) is a type of artificial neural network (ANN) that is used for a variety of tasks, including classification and regression. An MLP consists of an input layer, one or more hidden layers, and an output layer. The input layer receives the inputs to the network, which are then processed and transformed by the hidden layers. The output layer produces the final output of the network (Hassan et al. 2016).

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model that was trained on a massive corpus of text data, allowing it to learn rich representations of the relationships between words and their meaning (Devlin et al., 2018). These representations can be fine-tuned for specific NLP tasks, e.g., TC, by tokenizing the text and converting it to numerical representations using pre-trained tokenizers. These representations are fed into the pre-trained BERT model to obtain contextualized representations of the input text (Chi et al., 2019). These representations can be thought of as a fixed-length vector, which is then passed through a fully connected neural network (NN) for classification. One key advantage of using BERT for TC is that it can handle contextual information effectively.

These ML methods were applied using the following tools and information sources:

- The Python 3.8 programming language. (Van Rossum & Drake, 2009).
- Sklearn – a Python library for ML methods. (Buitinck et al., 2013).
- Numpy – a Python library that provides fast algebraic calculous processing (Harris et al., 2020).
- Pandas – a Python library for data analysis. It provides data structures for efficiently storing large datasets and tools for working with them (McKinney, 2010).
- Imblearn – a Python library for balancing imbalanced datasets in machine learning with oversampling or undersampling (Lemaitre et al., 2017).
- Tensorflow – an open-source Python library for building ML-DL models (Abadi et al., 2015).
- Transformers – a Python library for natural language processing. It provides pre-trained models based on transformer architecture that can be fine-tuned for specific use cases (Wolf et al., 2020). Hugging Face is a leading AI research lab of NLP. It provides a platform for researchers, developers, and data scientists to access and use the latest NLP models and tools (Huggingface API, 2023).

We applied three preprocessing methods: (1) Punctuation removal, (2) Change to lowercase and remove HTML tags and @user hashtags, and (3) Stop words removal.

We applied three resampling methods to balance the data: (1) Random oversampling, (2) Random undersampling, and (3) SMOTE oversampling.

4 Experimental Setup

Our way of working was based on the train and dev datasets only. The goal was to train different models on the train dataset and select the best models according to the F1 score (according to the competition requirement) on the dev dataset.

For classical ML models, we applied the following process. In the first step, for each language, we created a TF-IDF table for all the n-grams in the language and tried to identify how many grams should be selected as model features. We chose the word unigrams and [3,4,5,6,7,8,9,10] char n-grams.

We ran 4 classical classifiers - LR, MNB, RF, and SVM, and the MLP classifier, with varying numbers of features - 500, 1000, 2000, 3000, 4000, and 5000 and saw which number of features achieves the highest score for each model. At this stage, we did not tune the parameters of the classifiers but worked with their default values as defined in Python. In large language databases, as well as in language datasets where we saw that all scores are concentrated in 5,000 features, we ran additional amounts of features and reached a total of 15,000 features. We also performed various

comparisons on the 'min_df' parameter that determines the minimum number of documents in which the selected features appear, to make sure that they are indeed meaningful. We tested ranges of between 2-8 minimum number of documents and found an ideal number for each language and model.

In the second step, to find the optimal number of features for each of the 4 models, we tested the model again and this time compared resampling methods. We ran each model on the original dataset, on the dataset with random undersampling, and on the dataset with oversampling - once with random and once with SMOTE. We compared the results of all methods and created a list of the preferred resampling for each type of model in each language.

In the third step, we applied different pre-processing methods to the data - each method separately as well as the combinations of the methods and examined the level of improvement in each model and the effect of each pre-processing method.

In the fourth step, to each outstanding model, we added features that came from the sentiment lexicon in the same language and key positive and negative words in the language, to see if they can affect the classification. We checked for each model whether the sentiment dictionary improves or damages the results.

In the fifth step, for each outstanding model including all the additions from earlier: resampling, sentiment lexicon features, etc. we performed Hyperparameter tuning to choose ideal parameters for each model and in each language to achieve a maximum F1 score in prediction.

Next, we tried to train a more complex deep learning-based model for each language. We chose the BERT infrastructure and with it, we assigned vectors to each tweet in the train and dev dataset. We built a simple neural network for classification from the Bert model and performed fine-tuning to adapt the language model to the content of our dataset and to learn from the classifications of the train dataset. And then we set the model to classify the dev dataset based on what it learned. Also at this stage, we performed several different pre-processing methods on the information.

For each language, we performed different runs of the BERT model with various epochs and batch sizes to find an ideal classifier model. We also used several BERT infrastructures for each language. All languages were trained by African models - 'castorini/afriberta_large' (Kelechi et al. 2021), and 'bonadossou/afrolm_active_learning' (Dossou et al. 2022), and a multilingual model trained on sentiment classification for Twitter tweets - "Twitter/twhin-bert-base"(Zhang, 2022). Also for each language, we trained a unique BERT model that was trained on the language itself that we found in huggingface.

In the final step of these 2 studies, we selected 5 outstanding models for each language, which achieved the highest results on the dev dataset. With the help of these models, we predicted the classification for the test dataset.

5 Experimental Results

Tables 4 and 5 present the F1-scores over the of our models for tasks A and B. Table 4 for the best ML classifiers in each language, presents the classifier, feature parameters like n-gram analyzer, max number of features, minimal document (tweet) frequency, method of resampling, and usage of sentiment lexicon as additional features. Table 5 shows the best BERT-based NN, including the trained model name from huggingface, number of iterations (epochs), batch size, and model final validation accuracy and loss.

It is important to note that the results listed in Tables 4-5 are the optimal final results we achieved in each language. Within the deadline of the competition, in some languages, we submitted models that are less good than what you see here. The results submitted can be seen in Appendix D.

Analyzing the above results lead to several interesting conclusions can be identified concerning ML classifiers (from Table 4):

- Using char n-grams is usually much better than word n-grams. The range of n-grams varies from language to language.
- In general, features with a min df of 2 or 3, i.e., expressions that appeared in at least 2 or 3 tweets in the training set were very good. In rare cases, we saw higher numbers, such as 7 or 8, and in most languages usually caused overfitting.
- In large datasets (e.g., Hausa or multilingual), the outstanding models had a greater number of features.
- Although all 5 classifiers we selected appeared at least once, the most frequent and powerful classifiers are SVM and RF.
- In general, oversampling improved the models and obtained better results than without resampling or with undersampling. Within oversampling, the SMOTE method is generally better than the random method.

Language	Analyzer	n-gram range	Min df	Max features	Classifier	Resampling	Lexicon	Dev f1	Test f1
Algerian	char	3	7	8000	RF	over-Rand	yes	0.624	0.593
	char	3	2	5000	MLP	over-SMOTE	no	0.615	0.588
	char	5	8	5000	SVM	over-SMOTE	no	0.643	0.586
Amharic	char	5	2	8000	SVM	over-SMOTE	yes	0.53	0.36
	char	6	2	10000	RF	over-Rand	no	0.53	0.323
	char	4	2	4000	MNB	under-Rand	no	0.524	0.24
Hausa	char	4	3	15000	SVM	without	no	0.779	0.757
	char	4,6	3	20000	SVM	without	no	0.779	0.757
	char	5	6	10000	SVM	over-Rand	no	0.772	0.756
Igbo	char	5	2	5000	SVM	over-SMOTE	no	0.77	0.791
	word	1	6	5496	SVM	over-Rand	no	0.785	0.781
	char	4,5	2	12000	RF	over-SMOTE	no	0.778	0.78
Kinyarwandan	char	6	2	8000	SVM	over-SMOTE	yes	0.621	0.596
	char	6	3	8000	LR	over-Rand	yes	0.621	0.588
	char	6,8	2	10000	LR	over-Rand	no	0.582	0.585
Moroccan	char	5	3	12000	SVM	over-SMOTE	no	0.97	0.554
	char	4,10	2	5000	MLP	over-SMOTE	no	0.967	0.545
	char	4	2	5000	MLP	without	no	0.969	0.544
Mozambique Portuguese	char	4	2	3000	LR	under-Rand	no	0.604	0.638
	char	7	2	5000	RF	over-Rand	no	0.631	0.63
	char	7	2	5000	RF	over-Rand	no	0.595	0.628
Nigerian pidgin	char	5	3	12000	SVM	over-SMOTE	yes	0.738	0.613
	char	5	4	12000	SVM	over-SMOTE	no	0.72	0.594
	char	5	2	3000	MLP	under-Rand	no	0.494	0.594
Swahili	char	7	2	5000	RF	over-SMOTE	yes	0.518	0.561
	word	1	2	8000	SVM	SMOTE	yes	0.527	0.56
	char	3	2	5000	SVM	over-SMOTE	no	0.528	0.558
Twi	char	7	2	8000	RF	over-Rand	yes	0.667	0.642
	char	6	2	8000	SVM	over-SMOTE	no	0.643	0.642
	char	10	2	8000	SVM	over-SMOTE	yes	0.654	0.641
Xitsonga	char	4,5	3	8000	MLP	over-SMOTE	no	0.572	0.573
	char	4	8	3000	SVM	SMOTE	yes	0.615	0.561
	char	5	3	6000	SVM	over-Rand	no	0.619	0.56
Yoruba	char	3,5	3	10000	SVM	over-SMOTE	no	1	0.75
	word	1	3	8709	SVM	over-SMOTE	no	1	0.741
	char	4	3	5000	SVM	over-SMOTE	yes	0.986	0.726
TASKB - Multilingual	word	1	3	40000	SVM	without	no	0.941	0.705
	word	1	3	15000	SVM	without	no	0.918	0.699
	word	1	3	30000	LR	without	no	0.802	0.686

Table 4: F1 scores for best ML models for tasks A&B.

- In almost every language, there is a model combining features with a sentiment lexicon that improved the results, but in other models in the language, it hurt them.

It is also possible to identify several interesting conclusions concerning the models that trained vectors using BERT and classification using a simple neural network (according to Table 5):

- In most cases training the same BERT model for 5 epochs was more effective than training for 10 epochs which faced overfitting.
- In general, the batch size was better with a value of 64 compared to 32 although a small value represents more thorough training.
- Because we used each language in three BERT multilingual models, it can be concluded that in the absolute majority, the models trained on African languages (AfriBERTa, AfroLM) were better than the general multilingual model that was trained on tweets. Between the 2 African models, the older AfriBERTa model was better than AfroLM in all languages except Amharic.
- Concerning BERT models trained on multilingual datasets and then finetuned on

Language	Model	Epoch num	Batch size	Val acc	Test f1
Algerian	Twitter/twhin-bert-base	5	64	0.366	0.246
	castorini/afriberta_large'	10	64	0.384	0.243
	bonadossou/afrolm_active_learning	5	64	0.373	0.236
Amharic	bonadossou/afrolm_active_learning	5	64	0.387	0.538
	castorini/afriberta_large	5	64	0.406	0.16
	castorini/afriberta_large	10	64	0.808	0.796
Hausa	castorini/afriberta_large	5	64	0.803	0.791
	Davlan/bert-base-multilingual-cased-finetuned-hausa	5	64	0.782	0.785
	bonadossou/afrolm_active_learning	10	64	0.761	0.76
Igbo	castorini/afriberta_large	5	64	0.877	0.785
	castorini/afriberta_large	10	64	0.888	0.771
	bonadossou/afrolm_active_learning	5	64	0.851	0.761
Kinyarwandan	Davlan/bert-base-multilingual-cased-finetuned-kinyarwanda	5	64	0.579	0.613
	castorini/afriberta_large	5	64	0.651	0.579
	bonadossou/afrolm_active_learning	5	64	0.595	0.514
Moroccan	bonadossou/afrolm_active_learning	5	64	0.495	0.439
	Twitter/twhin-bert-base	5	64	0.487	0.397
	SI2M-Lab/DarijaBERT	5	32	0.505	0.358
Mozambique Portuguese	neuralmind/bert-base-portuguese-cased	5	32	0.846	0.645
	castorini/afriberta_large	5	64	0.8	0.571
	Twitter/twhin-bert-base	5	64	0.788	0.557
Nigerian pidgin	castorini/afriberta_large	10	32	0.912	0.636
	Twitter/twhin-bert-base	5	64	0	0.625
	castorini/afriberta_large	5	64	0.908	0.624
Swahili	castorini/afriberta_large	10	64	0.876	0.606
	Davlan/bert-base-multilingual-cased-finetuned-swahili	5	32	0.857	0.599
	castorini/afriberta_large	5	64	0.832	0.594
Twi	castorini/afriberta_large	5	64	0.798	0.618
	bonadossou/afrolm_active_learning	5	64	0.769	0.604
	Twitter/twhin-bert-base	5	64	0.796	0.604
Xitsonga	castorini/afriberta_large	5	64	0.767	0.513
	bonadossou/afrolm_active_learning	5	64	0.638	0.478
	Twitter/twhin-bert-base	5	64	0.664	0.427
Yoruba	castorini/afriberta_large	10	64	0.805	0.7
	Davlan/bert-base-multilingual-cased-finetuned-yoruba	5	32	0.801	0.692
	castorini/afriberta_large	5	64	0.77	0.666
TaskB – Multilingual	castorini/afriberta_large	5	64	0.643	0
	Twitter/twhin-bert-base	5	64	0.641	0
	bonadossou/afrolm_active_learning	5	64	0.636	0

Table 5: F1 scores for best BERT based models for tasks A&B.

specific languages, it was often less good than the general AfriBERTa model.

- Despite the great reputation of DL models, it seems that in African languages the achievements are close to classical ML models and sometimes slightly less good.

6 Conclusions and Future Research

In this paper, we describe our submissions to subtasks 12-A and 12-B of SemEval-2023: SA for low-resource African languages using Twitter datasets.

We applied word and char n-grams based on their tf-idf values, 4 supervised machine learning methods, 2 deep learning methods, 3 oversampling methods, 12 sentiment lexicons, and extensive parameter tuning. By various classification methods, we built a system for assessing the sentiment of tweets in low-resource African languages – in monolingual/multilingual datasets.

Comparing machine learning models, it seemed that the best model in most languages was SVM. In principle, it would appear that processes such as resampling to balance the reservoirs as well as adjusting parameters helped the models and that a pre-processing process did not help the accuracy of the models. Also, the integration of features from sentiment dictionaries helped the accuracy of the models.

Comparing deep learning models, it seems that BERT models trained on African languages multilingually gave a better result for most languages also compared to BERT models trained on a specific language.

There are various ideas for future research regarding the nature of Twitter messages:

(1) Using skip character n-grams because they can serve as generalized n-grams that allow us to overcome problems such as noise and sparse data (HaCohen-Kerner et al., 2017), which are common to Twitter messages.

(2) Many Twitter messages contain acronyms. Acronym disambiguation might enable better classification (HaCohen-Kerner et al., 2010A).

(3) Trying to enrich our training dataset and tune more parameters and longer training in the DL methods we checked (BERT based) because deep learning becomes better with more data to train and more time (without overfitting).

(4) Another idea that may improve the results is to use additional feature sets such as stylistic feature sets (HaCohen-Kerner et al., 2010B).

(5) Error analysis must be performed in-depth and repetitive patterns of errors, consistently incorrect classifications, etc. must be identified, to allow for the correction and improvement of the models.

(6) Concerning sentiment lexicons, the effect of negation must be examined, since it can be misleading and in particular in SA tasks.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Rémi Bardenet, Mátyás Brendel, Balázs Kégl, Michèle Sebag, 2013. Collaborative hyperparameter tuning. In 30th International Conference on Machine Learning (ICML 2013) (Vol. 28, pp. 199-207). Acm Press.

James Bergstra & and Yoshua Bengio, 2012. Random search for hyper-parameter optimization. Journal of machine learning research 13.2.

Leo Breiman. 1996. Bagging predictors. Machine learning 24(2), 123-140.

Leo Breiman. 2001. Random forests. Machine learning 45(1), 5-32.

Jason Brownlee. 2020. Imbalanced classification with Python: Better metrics, balanced skewed classes, cost-sensitive learning. Machine Learning Mastery.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, & Gaël Varoquaux, 2013. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning (pp. 108–122).

Raymond Chiong, Satia Budhi Gregorious, & Dhakal Sandeep, 2021. "Combining sentiment lexicons and content-based features for depression detection." IEEE Intelligent Systems 36.6: 99-105.

Chih-Chung Chang and Chih-Jen Lin, 2011. LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2(3), 1-27.

Nitesh V. Chawla, Kevin W Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority oversampling technique. Journal of artificial intelligence research, 16, 321-357.

Sun Chi, Qiu Xipeng, Xu Yige, Huang Xuanjing, 2019. "How to Fine-Tune BERT for Text Classification?." arXiv e-prints: arXiv-1905.

Corinna Cortes and Vladimir Vapnik, 1995. Support-vector networks. Machine learning 20.3 : 273-297.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bonaventure Dossou, Atnafu Lambebo Tonja, Oreen Yousuf, Salomey Osei, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, Chris & Chinenye Emezue, 2022. AfroLM: A Self-Active Learning-based Multilingual Pretrained Language Model for 23 African Languages. arXiv preprint arXiv:2211.03263.

Ronen Feldman, 2013. Techniques and applications for sentiment analysis. Communications of the ACM, 2013, 56(4): 82-89

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz, 2008. Combined one sense disambiguation of abbreviations. In Proceedings of ACL-08: HLT, Short Papers, Association for Computational Linguistics, pages 61-64, Columbus, Ohio, Association for Computational Linguistics. URL: https://aclanthology.org/P08-2.

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2010A. HAADS: A Hebrew Aramaic abbreviation disambiguation system. Journal of the American Society for Information Science and Technology, 61(9), 1923-1932.

Yaakov HaCohen-Kerner, Hananya Beck, Elchai Yehudai, and Dror Mughaz. 2010B. Stylistic feature sets as classifiers of documents according to their historical period and ethnic origin. Applied Artificial Intelligence, 24(9), 847-862.

Yaakov HaCohen-Kerner, Ziv Ido, and Ronen Ya'akobov. 2017. Stance classification of tweets using skip char Ngrams. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 266-278). Springer, Cham.

Yaakov HaCohen-Kerner, Rakefet Dilmon, Maor Hone, Matanya Aharon Ben-Basan, 2019A. Automatic classification of complaint letters according to service provider categories. Information Processing & Management, 56(6), 102102.

Yaakov HaCohen-Kerner, Yair Yigal, and Daniel Miller. 2019B. The impact of Preprocessing on Classification of Mental Disorders, in Proc. of the 19th Industrial Conference on Data Mining, (ICDM 2019), New York.

Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation, PloS one, vol. 15, p. e0232525.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gerard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, & Travis E. Oliphant (2020). Array programming with NumPy. Nature, 585(7825), 357–362.

Ramchoun Hassan, Mohammed Amine Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil, 2016. "Multilayer Perceptron: Architecture Optimization and Training." International Journal of Interactive Multimedia and Artificial Intelligence 4, no. 1 (2016): 26+.

Haibo He and Yunqian Ma (Eds.). 2013. Imbalanced learning: foundations, algorithms, and applications.

Tin Kam Ho. 1995. Random decision forests. Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE.

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

HuggingFace API, 2023. https://huggingface.co/docs/api-inference/index Last Access: 13/Feb/2023

Clayton Hutto, and Eric Gilbert, 2014. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." Proceedings of the international AAAI conference on web and social media. Vol. 8. No. 1. 2014.

Ogueji Kelechi, Yuxin Zhu, and Jimmy Lin, 2021. "Small data? no problem! exploring the viability of pretrained multilingual language models for low-

resourced languages." Proceedings of the 1st Workshop on Multilingual Representation Learning.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim and Sung Hyon Myaeng, 2006. "Some Effective Techniques for Naive Bayes Text Classification," in IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 11, pp. 1457-1466, Nov. 2006, doi: 10.1109/TKDE.2006.180.

Svetlana Kiritchenko, Zhu Xiaodan, Mohammad Saif, 2014. Sentiment Analysis of Short Informal Text. The Journal of Artificial Intelligence Research (JAIR). 50. 10.1613/jair.4272.

Bartosz Krawczyk, 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221-232.

Guillaume Lemaitre, Fernando Nogueira, & Christos K. Aridas, 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, 18(17), 1-5.

Bing Liu, 2012. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1):1–167.Diana G. Maynard and Mark A. Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In Lrec 2014 proceedings. ELRA.

Wes McKinney, 2010. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 56 - 61).

Rodrigo Moraes, João Francisco Valiati, Wilson P. Gavião Neto, 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN, Expert Systems with Applications, Volume 40, Issue 2, 2013, Pages 621-633, ISSN 0957-4174.

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

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

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

Bo Pang, and Lillian Lee, 2008. Opinion mining and sentiment analysis. Foundations and Trends® in information retrieval 2.1–2: 1-135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pages 79–86. Association for Computational Linguistics.

Bo Pang and Lillian Lee, 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting on the Association for computational linguistics. pp. 115–124. Association for Computational Linguistics.Mohammad Saif, 2022. "Ethics sheet for automatic emotion recognition and sentiment analysis." Computational Linguistics 48.2 (2022): 239-278.

Yimam Seid Muhie, Alemayehu Hizkiel Mitiku, Ayele Abinew, and Biemann Chris. 2020. Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models. In Proceedings of the 28th International Conference on Computational Linguistics, pages 1048–1060, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sarang Shaikh, Sher Muhammad Daudpota, Ali Shariq Imran, and Zenun Kastrati, 2021. Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models. Applied Sciences, 11(2), 869.

Mikalay Tsytsarau, and Themis Palpanas, 2012. Survey on mining subjective data on the web. Data Min. Knowl. Discov. 24, 478–514 (2012).

Guido Van Rossum & Fred Drake, 2009. Python 3 Reference Manual. CreateSpace.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi,

Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, & Alexander M. Rush, 2020. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38–45). Association for Computational Linguistics.

Li Yang, Ying Li, Wang Jin, & R. Simon Sherrat, 2020. Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning. IEEE Access, 8, 23522-23530.

Zhang Xinyang, Malkov Yury, Florez Omar Park, Serim McWilliams Brian, and Jiawei Han, 2022. "TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations." ArXiv.

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu & Wayne Niblack, 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In Third IEEE international conference on data mining (pp. 427-434). IEEE.

Appendix

A Details of the Training Sets

Language	Pos.	Neg.	Neu.	Total
Hausa	4687	4573	4912	14172
Yoruba	3542	1872	3108	8522
Igbo	3084	2600	4508	10192
Nigerian Pidgin	1808	3241	72	5121
Amharic	1332	1548	3104	5984
Algerian Arab.	417	892	342	1651
Moroccan Arab.	1758	1664	2161	5583
Swahili	547	191	1072	1810
Kinyarwanda	899	1146	1257	3302
Twi	1644	1315	522	3481
Mozambican Portuguese	681	782	1600	3063
Xitsonga	384	284	136	804
TaskB-Multilingual	20783	0.32634	20108	63685

B Details of the Dev Sets

Language	Pos.	Neg.	Neu.	Total
Hausa	887	894	896	2677
Yoruba	884	443	763	2090
Igbo	560	470	811	1841
Nigerian Pidgin	447	813	21	1281
Amharic	333	388	776	1497
Algerian Arab.	105	223	86	414
Moroccan Arab.	385	360	470	1215
Swahili	137	48	268	453
Kinyarwanda	225	287	315	827
Twi	183	147	58	388
Mozambican Portuguese	171	196	400	767
Xitsonga	96	72	35	203
TaskB-Multilingual	4413	4341	4899	13653

C Details of the Test Sets

Language	Total
Hausa	5303
Yoruba	4515
Igbo	3682
Nigerian Pidgin	4154
Amharic	1999
Algerian Arab.	958
Moroccan Arab.	2961
Swahili	748
Kinyarwanda	1026
Twi	949
Mozambican Portuguese	3662
Xitsonga	254
TaskB-Multilingual	30211

D Competition Best Submitted Results

Language	Place in competition	Number of competitors
Hausa	21	35
Yoruba	22	33
Igbo	18	32
Nigerian Pidgin	28	32
Amharic	24	29
Algerian	30	30
Moroccan	30	32
Swahili	18	30
Kinyarwanda	28	34
Twi	23	31
Mozambique Portuguese	24	30
Xitsonga	9	31
Multilingual	31	33

E Sentiment Lexicons Used

Language	Number of positive words	Number of negative words	Source	Remarks
Hausa	5168	7979	Link1 + Link2	
Yoruba	6167	9154	Link1 + Link2	
Igbo	4746	7991	Link1 + Link2	
Amharic	1747	2143	Link	
Algerian	2008	2545	Link	Standart Arabic Lexicon.
Moroccan				
Swahili	548	766	Link	
Kinyarwanda	1540	1961	Link	
Twi	878	1601	Link	
Mozambique Portuguese	2066	2709	Link	Standart Portuguese Lexicon.
Xitsonga	1029	1637	Link-Vader	Translated English lexicon.
Nigerian Pidgin	3072	3988	Link-Vader	English lexicon.

- For the Xitsonga language We did not find a lexicon, so We used an English lexicon and translated with Google translate. Not all words were translated successfully.
- For the Nigerian Pidgin Language, We also did not find a lexicon, and the language does not exist in Google translate. So, We used a standart English sentiment lexicon, which was partially effective since this language contains many borrowed words from English.