Classifying Swahili News Articles using Probabilistic and Deep Learning Approaches

Samantha Ball
1603701
School of Computer Science and Applied Mathematics
University of the Witwatersrand

***Abstract*—Text classification is a classical Natural Language Processing (NLP) task which has seen rapid development over recent years. However, a vast proportion of text classification research has focused on the development of models for English language text. Across the field of NLP, relatively fewer studies have focused on the application of recent techniques to African languages. Therefore, this investigation focuses on text classification of news articles written in the Swahili language.**

In particular, this study explores a range of techniques from simpler, more classical text classification algorithms to complex, state-of-the-art Transformer networks. Accordingly, the Naïve Bayes, linear Support Vector Classifier, and BERT models are studied. The approaches are compared and contrasted in terms of classification performance, resource usage and speed. Additionally, specific techniques are employed to optimize each model and correct for the class imbalance present in the dataset.

The results indicate that the multilingual BERT model obtains the optimal classification performance with 90% test accuracy. However, this is at the cost of significant training time and computational resources in comparison to classical techniques. Notably, key optimizations such as the one cycle learning rate schedule, class imbalance adjustments and grid search are shown to have crucial effect on the training and performance of the models, leading to significantly improved performance.

***Index Terms*—transformer, bert, news classification, swahili**

I. Introduction

Text classification aims to automatically categorise text documents according to their content. Text classification has been applied in a wide range of applications including email filtering, tagging of customer queries, and fake news detection [1]. Approaches to text classification have evolved from simpler techniques such as the probabilistic Naïve Bayes model to deep networks such as the Transformer model. However, despite the rapid development in techniques, most studies have focused on the text classification in the context of the English language.

In an effort to stimulate research into Natural Language Processing techniques for African languages, the Artificial Intelligence for Development in Africa program (AI4D Africa) launched the African Language Dataset challenge to curate and annotate multiple African language datasets for use in NLP tasks [2]. The project aims to help close the technological gap between Africa and the world, focusing on languages such as Yorùbá, Luganda, Chichewa, and Swahili [2].

Stemming from this initiative, this investigation seeks to apply text classification techniques to the problem of Swahili news classification. Initially, classical techniques such as Naïve Bayes and Support Vector Machines will be implemented in order to determine the efficacy of these techniques on the Swahili dataset. Subsequently, the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) [3] network will be explored.

The speed, efficiency and simplicity of classical methods will be evaluated against the potential improvements in accuracy offered by more recent Transformer model. Furthermore, specific optimization techniques will be investigated for each algorithm to ensure optimal performance and improved learning.

The paper is structured as follows. In Section II, relevant background regarding the history and state-of-the-art of text classification algorithms is provided along with related work focusing on NLP for the Swahili language. In Section III, an overview of the project design is provided, detailing the motivation for the chosen dataset and machine learning models. Section IV, V and VI detail the data pre-processing and exploration phase, model implementation and preliminary results respectively. Thereafter, Section VII describes the optimization techniques applied to each model. Lastly, Sections VIII and VIIII present an analysis of the final results and summarize our central findings.

II. Related Work

In this section, relevant background to the investigation will be presented including current approaches to the text classification problem and recent studies investigating NLP tasks in the Swahili context.

A. Text Classification

Several approaches have been developed for the task of text classification stemming from the early development of the popular Naïve Bayes classifier. The Naïve Bayes classifier is computationally inexpensive with low memory requirements and has been widely used for spam filtering and sentiment analysis [4]. Subsequently, machine learning algorithms such as k-nearest neighbours and Support Vector Machines (SVMs) have been applied to the text classification problem. Tree and graph-based classifiers have also been explored, such as in the case of random forests and conditional random fields (CRFs) [5]. Furthermore, ensemble approaches such as bagging and boosting have shown potential in the text classification domain [6].

Most crucially, deep networks have demonstrated great ability to perform a wide variety of NLP tasks, with earlier networks such as recurrent neural networks (RNNs), long-short-term memory networks (LSTMs) and gated recurrent units (GRUs) paving the way for the development of the pivotal Transformer architecture [7]. Transformer-based methods first pre-train the network to extract semantic knowledge which then forms the base for a variety of NLP tasks, rather than one specific task. Transformer models such as the popular Bidirectional Encoder Representations from Transformers (BERT) [3] are then fine-tuned on the desired task. Improvements upon the BERT model include XLNet which incorporates auto-regressive pre-training, achieving the state-of-the-art on several NLP benchmarks [8].

B. Word Embeddings

In addition to the active development of text classification models, significant research has focused on the effective representation of text as numerical inputs to the respective models. The *Bag of Words* (BoW) approach is a simple representation relying on the frequency of each word in the text while the *n-gram* model extends this to capture the ordering between words [9]. A further popular approach known as *term-frequency-inverse-document-frequency* (TF-IDF) measures the relevance of a word in a document, taking into account the entire collection of documents. Specifically, TF-IDF measures the frequency of a word in a document divided by the number of documents containing the word, therefore penalising words that appear often across the corpus [10]. The more recent *word2vec* employs a shallow neural network to learn word embeddings from local context such that similar words are closely located in vector space [11]. Lastly, the *GloVe* (Global Vectors for Word Representation) model is an unsupervised learning algorithm utilizing statistical information from the co-occurence counts matrix [12]. In contrast, transformer models such as BERT generate their own word embeddings dynamically based on the given context and only require that documents are tokenized before input.

C. Transformer Architecture

The Transformer model is based on the concept of sequence-to-sequence models together with attention mechanisms. Sequence-to-sequence models consist of an encoder to map the input vector to a higher-dimensional space together with a decoder which generates an output sequence depending on the given task. The attention mechanism allows the transformer to focus on the parts of the input vector that are most important. The overall structure of the transformer architecture is depicted in Fig. 1, indicating the encoder and decoder structure together with the attention modules.

Critically, the transformer architecture utilizes the attention mechanisms in lieu of the traditional recurrent structures and employs positional encoding to retain word ordering [7]. Hence, the transformer model convert an input sequence into both a word vector embedding and a positional encoding vector. Due to the replacement of the recurrent layers with multi-head attention, the transformer network can be trained much faster than previous architectures by exploiting the parallelism offered by GPUs [7].

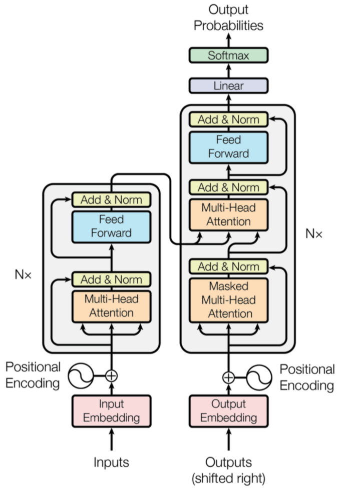


Fig. 1: Transformer architecture [7].

D. The Swahili Language

While the majority of current Natural Language Processing (NLP) research has focused on the development of models trained on English text, recent initiatives have emphasized the need for the development of multilingual and low-resource language models. The Artificial Intelligence for Development in African (AI4D Africa) has launched one of these initiatives to promote research into NLP techniques and datasets for African languages [2]. The project spans multiple languages and applications such as machine translation, text-to-speech and document classification [2].

Despite its prevalence as the most spoken language in Africa, with 150 million speakers [13], only a few previous studies have focused on the application of Natural Language Processing to the Swahili language. Wanjawa et al. [14] explore question answering in Swahili using semantic networks while Shivachi et al. [15] use a convolutional LSTM for Swahili word representation and part-of-speech tagging. Notably, Kastanos et al. [1] employ graph convolutional networks to the problem of Swahili new classification.

III. Project Design

In order to provide a comprehensive investigation into the application of current text classification techniques to the task of Swahili news classification, a variety of algorithms will be explored from a performance and efficiency perspective. This section details the project formulation and motivates the choice of dataset and machine learning models.

A. Dataset Selection

The Swahili News Classification dataset [16] collected as part of the AI4D African Language Dataset challenge is utilised for the investigation. The dataset provides a variety of news articles spanning six categories including *afya, burudani, kitaifa, kimataifa, michezo* and *uchumi* respectively. The dataset provides a suitable benchmark for investigation as it includes a variety of texts from the categories of health news, entertainment news, local, international, sports and finance news respectively. Additionally, the dataset is of high data quality as it was collected as part of the over-arching dataset development program.

Crucially, the dataset allows for comparison with previous results as both the Zindi Swahili News Classification challenge and work by Kastanos et al. [1] were benchmarked using the dataset, thus providing a baseline for comparison. In order to augment the dataset with additional data needed for text pre-processing, supplementary Swahili data from Mendeley data [17] will be utilised for the identification of stopwords.

B. Model Choice

Three contrasting models are chosen for investigation ranging from simple probabilistic approaches to state-of-the-art deep learning models.

B.1 Naïve Bayes

The first classic, probabilistic approach is the Naïve Bayes classifier. Naïve Bayes classifiers are a set of probabilistic classifiers based on Bayes' Theorem. Naïve Bayes classifiers rely on the assumption of conditional independence between every pair of input features given the target label [18]. The assumption of independence among the input features is considered to be 'naïve'.

Bayes' Theorem for a set of input features $x_1, ..., x_n$ and target variable y is given by,

$$P(y|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|y)P(y)}{P(x_1, ..., x_n)} \tag{1}$$

which, due to the independence assumption, is simplified to

$$P(y|x_1, ..., x_n) = \frac{\prod_{i=1}^{n} P(x_i|y)P(y)}{P(x_1, ..., x_n)} \tag{2}$$

Since $P(x_1, ..., x_n)$ is a normalizing constant, the classification rule can be written as,

$$\hat{y} = \arg\max_{y} \prod_{i=1}^{n} P(x_i|y)P(y) \tag{3}$$

Therefore, due to its simplicity, Naïve Bayes offers a very fast and simple baseline and a good comparator for more complex techniques. A further advantage is the relatively small amount of data needed to estimate the required parameters [19]. Naïve Bayes has been well known to work well for multi-class document classification problems and will be implemented as a baseline algorithm for the given problem.

B.2 Support Vector Machines

Support Vector Machines (SVMs) aim to find the optimal set of hyperplanes to separate the feature space into classes. A good choice of hyperplane will result in a large distance between the hyperplane and the nearest data point belonging to a specific class, known as a margin [20]. Therefore, in the context of classification, Support Vector Machines aim to solve the primal problem defined by,

$$\begin{gathered} \min_{w,b,\gamma} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \gamma_i \\ \textit{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \gamma_i \end{gathered} \tag{4}$$

which represents the maximization of the margin by minimizing the squared norm of the weights ω, and introduces a penalty for misclassified samples, as controlled by the parameter C [20]. To allow for imperfectly separable data, the γ term is introduced, representing the allowable tolerance that a misclassified sample may be from the correct margin boundary in each dimension.

Support Vector Machines are widely regarded as one of the best classical text classification algorithms. While slower than Naïve Bayes, Support Vector Machines obtain higher accuracy and are still much faster and simpler than deep learning models. In addition, Support Vector Machines offer the advantage of remaining effective in high dimensional spaces [20]. Support Vector Machines represents a geometric machine learning approach to the problem of text classification and therefore provide a contrast to both the more probabilistic Naïve Bayes classifier and the more complex deep learning models.

B.3 Transformer Models

Several different language models have been developed from the pioneering Transformer model. Notably, the Bidirectional Encoder Representations from Transformers (BERT) provides a pre-trained network for fine-tuning on a range of language tasks [3]. Variants of BERT include the smaller and more efficient DistilBERT which allows for faster training and inference while retaining 95% of the original performance [21]. On the other hand, the RoBERTa variant [22] offers improved performance at the cost of slower speeds and a larger model. Furthermore, the ALBERT model reduces memory consumption by decreasing the number of necessary parameters and thus speeding up training time [23]. The differences between the BERT variants are summarised in Table I.

In addition to the architecture variants, available BERT models include versions pre-trained on different textual data. For this investigation, the BERT model trained on a multilingual dataset will be utilised to provide the most appropriate pretraining for our Swahili task. The multilingual BERT

TABLE I: Comparison of BERT variants

Technique	Advantages	Disadvantages
BERT	High accuracy	Slow training and inference
DistilBERT	Smaller and more efficient	Less accurate
RoBERTa	Improved performance	Slow and large
ALBERT	Lower memory requirements	No multilingual variant

model is trained on 104 languages using Masked Language Modelling (MLM) [3]. As there are relatively few multilingual transformer models currently available, the base BERT model offering the advantage of being pre-trained on multiple languages was chosen over the other transformer variants.

The pretrained model provides a significant benefit due to the limited size of the training dataset which would render training the network from scratch an intractable alternative.

C. Evaluation Metrics

To provide a thorough analysis of the classification performance of each of the models, several metrics will be reported, including test accuracy, precision, recall and the F1-score. Each of the chosen metrics provides different insight into the performance of the models.

In addition, the confusion matrix will be plotted to indicate the nature of the misclassifications, and the training time will be recorded in order to evaluate the efficiency and practicality of the respective models.

C.1. Accuracy

Accuracy provides a measure of how many samples are correctly predicted and is calculated as follows,

$$Accuracy = \frac{(TP + TN)}{(TN + FP + FN + TP)} \tag{5}$$

where TP, TN, FP and FN represent the number of true positives, true negatives, false positives and false negatives respectively.

The test accuracy provides an appropriate measure of generalization as it reports the classification performance on unseen data. However the test accuracy can provide a skewed metric in the case of imbalanced datasets as good performance on the majority class will translate to high accuracy.

C.2. Precision

Precision can be interpreted as the confidence that a certain positive classification is correct. Precision is the rate at which samples classified as positive are correctly classified.

$$Precision = \frac{TP}{(TP + FP)} \tag{6}$$

In the multi-class classification context, precision calculates how many instances classified as a certain news category actually belong to that news category.

C.3. Recall

Recall, also known as the True Positive Rate (TPR) measures the rate at which all samples labelled as positive are identified.

$$Recall = \frac{TP}{(TP + FN)} \tag{7}$$

In the case of multi-class classification, this represents how many instances of a certain class are identified relative to all the samples in that class.

C.4. F1-Score

The F1-score is the harmonic mean of the precision and recall and therefore provides a combined metric balancing these two objectives.

$$F1\ Score = \frac{(2 * precision * recall)}{(precision + recall)} \tag{8}$$

The F1-score provides an insightful metric in the face of class imbalance as it incorporates both precision and recall and is not skewed by the majority class.

IV. Data Preparation

In order to prepare the dataset for classification, pre-processing and data exploration were performed to understand the structure and characteristics of the dataset.

A. Data Pre-processing

Several key pre-processing steps were performed to transform the data into the correct form for the text classification models, as summarized in the algorithm below.

Algorithm 1 Text Pre-processing

```
Read in stopword list
for sample in news articles do
    Convert to lowercase
    Remove ’,.”()-!? punctuation symbols
    Remove \n, \t and \xao characters
    Tokenize into individual words
    Remove stopwords that match those in stopword list
end for
```

The main data pre-processing steps are detailed as follows,

- The text was first converted to lowercase, followed by the removal of punctuation marks and whitespace characters to simplify the data.
- Tokenizing was then applied to split the text document in separate words for later processing such as creating the vocabulary and calculating word frequencies.

- Lastly, in order to effectively remove stopwords, the dataset was augmented with additional data from Mendeley data [17] which included a list of common Swahili stopwords to be removed from the news article data.

Stopwords represent common words that add no meaning and can therefore be removed without changing the overall content of the sentence. Overall, data cleaning is vital to remove irrelevant noise from the dataset, allowing the NLP models to focus on the text meaning.

B. Data Exploration

In order to understand the characteristics of the dataset and ensure data quality, data exploration was performed. Several key metrics were calculated to characterize the target variable and input features respectively.

B.1. Class Distribution

The dataset contains 23268 samples mapping the news article content to one of the six target classes. In order to inspect the characteristics of the target variable, the target variable distribution was plotted as shown in Fig. 4 below.

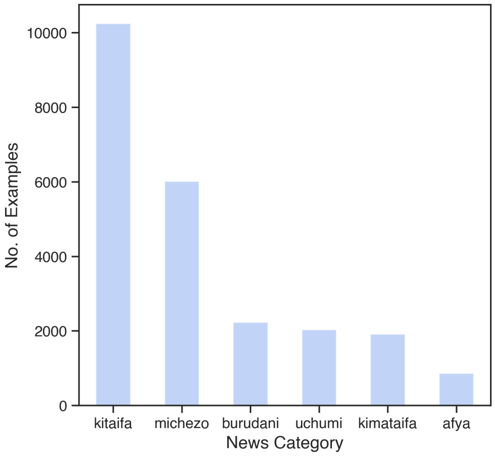
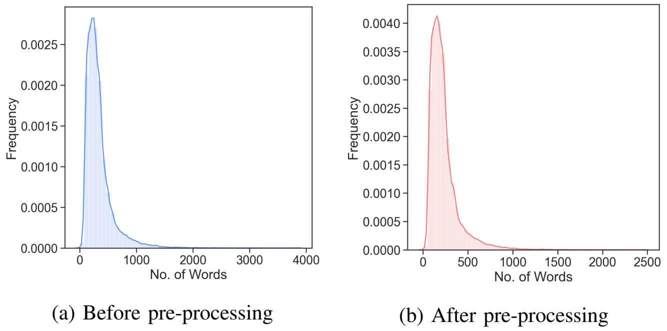


Fig. 2: Distribution of news article classes in the dataset.

From Figure 4 it can be observed that the news articles are classified into six categories including *kitaifa, michezo, burundani, uchumi, kimataifa* and *afya* respectively. However, since the number of samples in each category varies greatly, the dataset is imbalanced. This is clearly demonstrated as the local news category (*kitaifa*) makes up 44% of the samples, while health news (*afya*) makes up only 4%. Class imbalance can have a significant effect on classification performance.

B.2. Feature Characteristics

To explore the attributes of the input features, or news articles, a variety of metrics were calculated. Firstly the median number of words per sample was calculated to inspect the length of each article, and was found to be 274. However, the sample lengths were reduced following pre-processing, with a final median sample length of 188. This is indicated by the sample length distributions in Fig. 3.



(a) Before pre-processing

(b) After pre-processing

Fig. 3: Distribution of sample lengths before and after text pre-processing.

The sample lengths were then further divided into the median sample length per class, as shown in Fig. 4. It can be noted that the entertainment news category (*burudani*) contains significantly shorter articles than the other categories, both before and after data cleaning.

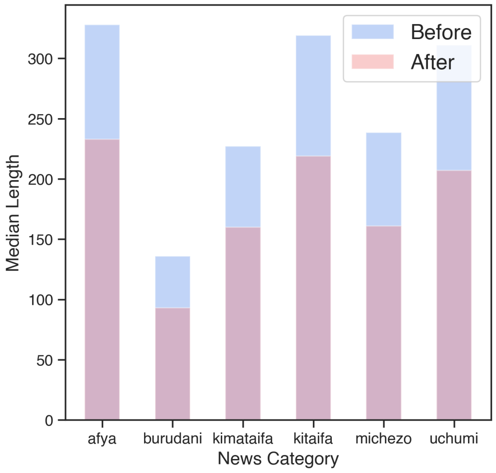


Fig. 4: Median sample length per class before and after text pre-processing.

V. Model Implementation

The following section details the implementation of each model together with selection of appropriate hyperparameters.

A. Naïve Bayes

Naïve Bayes can be applied to the problem of text classification as follows, where the probability of a given target class c given text t is found to be,

$$P(c|t) = \frac{P(c)\prod_{\omega \in t} P(\omega|c)^{n_{\omega t}}}{P(t)} \tag{9}$$

where $P(c)$ is the prior probability of class c, $P(t)$ is the normalizing constant, $P(\omega|c)$ is the probability of observing the word ω given the class c and $n_{\omega t}$ is the frequency of word ω in text t [18].

The available variations of the Naïve Bayes classifier differ depending on the underlying distribution that is assumed for $P(\omega|c)$. Commonly used distributions include Gaussian, Bernoulli and multinomial naïve Bayes models. The multinomial distribution was chosen for the text classification problem as it is suitable for discrete counts and therefore the modelling of word counts.

A.1 Word Embeddings

In order to transform the textual data into a suitable numerical format, the training data was first cleaned and tokenized as described in Section IV. The cleaned samples were then used to create a vocabulary of all the words in the corpus, followed by a *Document Term Matrix* to store the frequency of each word in each respective news article. Lastly, in order to account for varying document lengths, *term-frequency* was introduced by calculating the number of times a certain word appears in a document.

B. Support Vector Machine (SVM)

Support Vector Machines may be implemented with a variety of kernel functions, including linear, polynomial, radial basis function (rbf) or custom kernels. For the Swahili news classification task, the SVM was implemented using a linear kernel, as represented in the following equation,

$$\min_{w,b} = \frac{1}{2}w^T w + C\sum_{i=1}^{n} max(0, y_i(w^T\phi(x_i)+b)) \qquad (10)$$

where ϕ is the identity function representing a linear kernel. The same data preprocessing steps were followed as for the Naïve Bayes classifier, transforming the text into word embeddings through construction of a Document Term Matrix and the use of *term-frequency*.

B.1 Hyperparameters

The SVM was trained with stochastic gradient descent, therefore requiring careful selection of the hyperparameters. The *l2* loss was chosen as the penalty term while the α term controlling the strength of the regularization was initialized to 1×10^{-3}. The SVM trained for 5 epochs initially. All hyperparameters were then further optimized during model optimization in Section VII.

C. Transformer Models

In order to apply the transformer model to the task of Swahili news classification, the selected multilingual BERT model was fine-tuned on the Swahili dataset. This allowed for the use of transfer learning by utilising the previous knowledge gained during pre-training of the BERT model for the current task.

C.1 NLP Libraries

Several deep learning frameworks were utilised to train and evaluate the BERT network. A pre-trained BERT model from the *huggingface* Transformers library [24] was utilised together with the *ktrain* library [25] in order to load the model, perform data pre-processing into the expected format, and train the model on the Swahili news dataset.

Specifically, the *bert-base-multilingual-cased* model [3] was selected from the available models in the *huggingface* package. The *ktrain* library provides a wrapper functionality to load, train and validate the model on the given data. The dataset was split into 70% training data, 15% validation data and 15% test data.

C.2 Hyperparameters

Several hyperparameters required careful tuning to ensure successful training on the Swahili dataset.

a) Learning rate schedule

A learning rate schedule was employed in favour of a fixed learning rate as cyclical learning rates have been shown to simultaneously improve performance while requiring fewer iterations [26]. In particular, the cyclical triangular learning rate schedule was implemented as detailed in [26], shown below in Fig. 5.

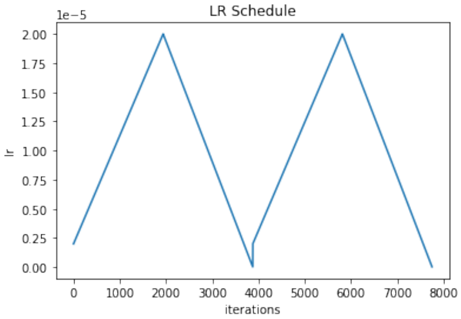


Fig. 5: Triangular learning rate schedule as proposed in [26]

In order to select an appropriate maximum learning rate, learning rates between 2×10^{-5} and 5×10^{-5} were considered as this range is known to work well for BERT-based models. However, through experimentation, it was found that a maximum learning rate of 5×10^{-5} was too large and caused the loss to diverge, as depicted in Fig. 6. Therefore a maximum learning rate of 2×10^{-5} was chosen to ensure convergence.

b) Batch size

Selecting an appropriate batch size was essential to ensure that memory limitations were taken into account and thus to

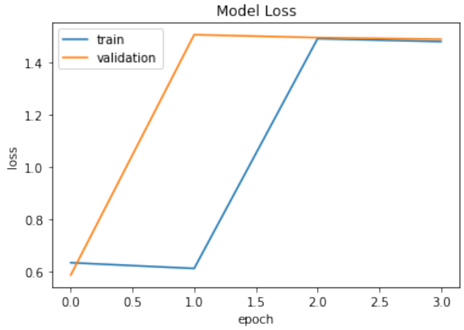


Fig. 6: Training loss diverges if maximum learning rate too high.

prevent resource exhaustion on the Google Colab platform. A batch size of 4 was found to allow for training without exceeding memory limits.

c) Epochs

The number of epochs was initially restricted to 2 epochs to ensure a feasible training time. However, the number of epochs is further explored during model optimization.

D. *Model Training*

The BERT model and subsequent variants were trained using an NVIDIA GPU in the Google Colab environment to ensure feasible training times.

D.1 Checkpoints and Visualisation

In order to provide a comprehensive training pipeline, a checkpoint and visualisation system was integrated into the training process. To guard against errors in the training process, the trained model was saved as a checkpoint, allowing the model to be reloaded and evaluated in the case of a connection error.

Additionally, in order to identify overfitting, the training and validation loss were plotted over the duration of training to ensure correct learning.

D.2 Dataset Size

The effects of dataset size on training and performance were investigated, as summarized in Table II.

TABLE II: Effects of Training Dataset Size

Algorithm	Dataset Size	Accuracy	F1-score	Time (hrs)
bert-multilingual	7756	0.886	0.81	1.22
bert-multilingual	15512	0.896	0.82	2.29

It can be noted that training with half the training dataset size (7756 samples) obtains very similar performance to training with the whole training set. This is a very interesting result given that this allows the model to train in approximately half the time. However, the full dataset was utilised in the final results in order to provide a fair comparison with the other approaches.

VI. Preliminary Results

The classification results obtained on the test set for each text classification model are summarised in Table III. It can be seen that the deep learning approach outperforms the classical methods on almost all metrics. However, the BERT model requires a significantly longer training time while the training time for both Naïve Bayes and the Support Vector Machine is almost negligible.

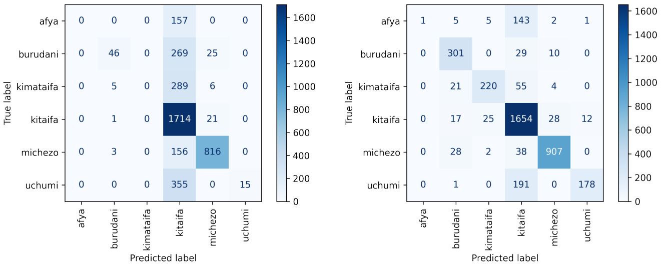
It should be noted that accuracy can provide a skewed representation of model performance since there is an over-representation of the *kitaifa* class and thus if the model performs well on this class, a high accuracy will be reported even if performance on the alternative classes remains poor. Thus, given the imbalanced nature of our dataset, we use the F1-score as our primary metric as it provides a balance of precision and recall. Moreover, the macro average for the precision, recall and F1-score is reported in favour of the weighted average in order to place equal importance on each news category and provide a fair representation of model performance.

A significant discrepancy in F1-score between the classical and transformer-based approaches is observed with poor recall obtained by both classical methods. This indicates that the multilingual BERT model provides the best balance of precision and recall. Furthermore, although the SVM outperforms the BERT model with respect to precision, the recall of the SVM is much lower, thus indicating that although most of the samples classified as a particular news category are correct, there are still many other instances of the news category that are not identified by the SVM.

A. *Confusion Matrix*

In order to further explore the classification performance, the confusion matrix for each model is plotted.

A.1 Classical Methods



(a) Naïve Bayes

(b) SVM

Fig. 7: Confusion matrix indicating classification performance for the Naïve Bayes and Support Vector Machine approaches respectively.

TABLE III: Classification Performance Metrics before Optimization

Algorithm	Test Accuracy	Precision	Recall	F1-Score	Training Time (hrs)
Classical					
Naïve Bayes	0.67	0.56	0.33	0.32	0.01
Support Vector Machine	0.84	**0.89**	0.66	0.68	0.02
Transformers					
bert-base-multilingual	**0.90**	0.85	**0.80**	**0.82**	2.29

From the confusion matrix given by the Naïve Bayes classifier in Fig. 7a, it is clearly seen that most samples are predicted to be in the majority class *kitaifa* and that the classifier is heavily influenced by the class imbalance present in the training dataset. This results in the poorly represented classes such as the health news (*afya*) class and international news (*kimataifa*) class, which have the smallest number of samples, obtaining no correct classifications. This leads to a very poor recall score for the Naïve Bayes classifier overall.

In contrast, the SVM indicates much better performance as larger numbers of correct classifications are indicated along the diagonal, with less dominance of the majority *kitaifa* class. However, poor results are still obtained for the least represented *afya* class as only one sample is correctly classified. Overall, the SVM approach far outperforms the Naïve Bayes classifier across all metrics.

A.2 Deep Learning Methods

The confusion matrix indicating classification performance for the multilingual BERT model is shown in Fig. 8 below.

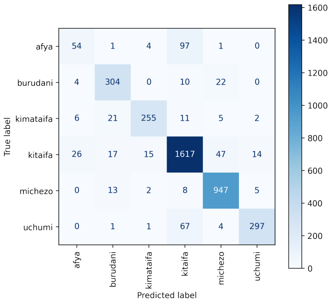


Fig. 8: Confusion matrix indicating classification performance for the multilingual BERT model.

It can be observed that the BERT model performs far better for the poorly represented *afya* class in comparison to the two classical techniques, with 54 correct classifications. Nevertheless, the majority of *afya* samples are still predicted to be in the majority *kitaifa* class, therefore requiring model improvement to provide viable classification performance.

VII. Model Optimization

Following the preliminary results, several model optimization techniques were employed to improve both model performance and training.

A. Naïve Bayes

Since the Naïve Bayes model contains very finite parameters, parameter tuning is not an effective means of optimization. Instead, the problem of class imbalance is addressed by integrating information regarding the complement of each target class.

A.1 Complement Naïve Bayes

Complement Naïve Bayes [27] is an adaptation of the multinomial Naïve Bayes classifier which aims to overcome the problems associated with imbalanced datasets. This is achieved by including information regarding the complement of each class. Specifically, the model parameters are estimated by training with data from all the classes excluding the current class c [27]. Accordingly, the documents assigned to the class c are those who poorly match the complement parameter estimates. The classification rule is therefore,

$$\hat{c} = \arg\min_{c} \sum_{i} t_i \omega_{ci} \tag{11}$$

where t_i represents document i. The Complement Naïve Bayes approach is effective against class imbalance as it allows for a more even amount of training data per class [27].

B. Support Vector Machines

Support Vector Machines involve a number of hyperparameters that can be tuned for optimal performance. One hyperparameter, known as the *class weight*, is particularly pertinent for our imbalanced problem and is explored as a key optimization below.

B.1 Class Weights

The *class weight* parameter allows more importance to be placed on classes that are under-represented in the dataset. Weights are calculated to be inversely proportional to class frequency according to,

$$\omega_i = \frac{no.\ samples}{no.\ classes * no.\ samples_i} \tag{12}$$

This allows more weight to be placed on the classes with fewer training samples, therefore counteracting class imbalance.

B.2 Grid Search

In addition to correcting for class imbalance, the remaining hyperparameters requiring tuning were optimized through grid search. Grid search is an exhaustive search of specified parameters in order to find the optimal combination. Key findings from the grid search process included SVM parameters such as the use of 50 epochs in training and $\alpha = 1 \times 10^{-4}$.

In addition, several optimizations of the word embedding stages were highlighted such as using *inverse document frequency* in addition to *term frequency* and expanding the n-gram range from (1, 1) to (1, 2) during the creation of the Document Term Matrix. An n-gram range of (1, 2) expands the n-gram construction to include bigrams as well as unigrams.

C. Transformer Models

Several key optimizations were applied to the BERT model to improve both the classification performance and training speed. Specifically, improved learning rate schedules were explored along with adjustments for class imbalance.

C.1 Improved Learning Rate Schedule

In order to improve the training speed of the multilingual BERT model, the one-cycle learning rate schedule was utilised in place of the original triangular learning rate schedule. The one-cycle schedule [28] replaces the repetitive increase and decrease structure present in the triangular policy with a single cycle starting from the minimum learning rate to the maximum and then back to below the minimum, as depicted in Fig. 9.

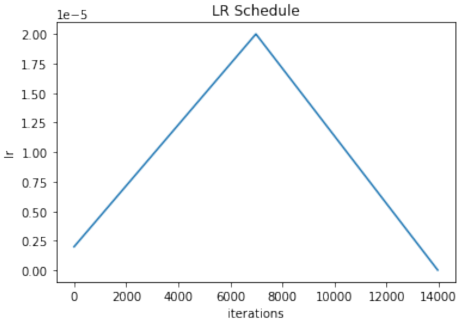


Fig. 9: One cycle learning rate schedule as proposed in [28]

The one-cycle policy allows for higher learning rates and faster convergence, known as super-convergence [28].

C.2 Number of Epochs

The improved one cycle learning rate schedule offers the vital optimization of faster training, therefore allowing for more epochs in the same training time. Thus, the one cycle schedule was combined with double the epochs in order to further optimize the BERT network.

C.3 Adjusting for Class Imbalance

Similarly to the SVM, class weights can be set to emphasize under-represented classes during model training. Using the *class weight* parameter in *ktrain*, the class weights were set manually. Initially, the class weights were chosen to be the ratio of the number of samples of each class to the number of samples in the majority class, as per traditional use. However, while this resulted in a great increase in recall, precision and accuracy were severely degraded. Thus, in order to provide a balance between the improved recall and decreased accuracy and precision, the class weights were then modified to half their original value.

VIII. Results and Analysis

The final results detailing the effects of the various optimizations on model performance are summarized in Table IV. The original results before optimization are contrasted with each investigated technique. It should be noted that optimizations are cumulative. For instance, grid search SVM also makes use of class weights, while the BERT model with class weights also makes use of the one cycle learning policy.

A. Classification Performance

Once again, since all news categories are of equal importance and we require the models to perform optimally on all categories, the macro F1-score is utilised as the primary metric.

A.1 Classical Methods

It can be observed that the Support Vector Machine (SVM) together with the optimizations of class weight and hyperparameter tuning attains the highest accuracy and F1-score with respect to the classical algorithms. While the addition of the class weight ensured significantly improved recall from 0.66 to 0.81 by countering the effects of class imbalance, hyperparameter tuning through grid search allowed for a 2% increase in accuracy.

The effects of adjusting for class imbalance through the addition of class weights to the SVM are clearly seen by the 10% improvement in F1-score. Similarly, the use of Complement Naïve Bayes instead of Multinomial Naïve Bayes also works to counteract the class imbalance, resulting in a 24% improvement in the F1-score. This highlights the significance of class imbalance adjustments for both classical methods.

Furthermore, a great improvement in accuracy is achieved for the Naïve Bayes algorithm by employing the complementary approach, with no associated increase in training time. In contrast, the training time for the optimized SVM includes time taken for grid search to find best hyperparameters and thus is no longer negligible.

TABLE IV: Classification Performance Metrics with and without Optimization

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Training Time (hrs)**
Classical					
Naïve Bayes	0.67	0.89	0.33	0.32	0.01
Naïve Bayes + complement	0.78	**0.92**	0.52	0.56	0.01
SVM	0.84	0.89	0.66	0.68	0.02
SVM + class weight	0.84	0.76	**0.81**	0.78	0.02
SVM + grid search	**0.86**	0.79	0.80	**0.79**	0.67
Transformers					
bert-base-multilingual	0.90	0.85	0.80	0.82	2.29
bert-multilingual + one cycle	**0.90**	**0.86**	0.83	0.84	2.48
bert-multilingual + class weight	0.86	0.79	0.88	0.82	2.32
bert-multilingual + half class weight + additional epochs	0.89	0.83	**0.88**	**0.85**	2.41

A.2 Deep Learning Methods

The most notable improvement to the multilingual BERT model is the use of the one-cycle learning rate schedule as this significantly reduced training time, allowing for additional epochs and improved learning. This is clearly indicated in the results as the addition of the one cycle policy together with four epochs rather than the original two epochs allowed for improvement across almost all classification metrics. While the accuracy score remained the same, the increased number of epochs resulted in a 3% improvement in recall and 2% improvement in macro F1-score, therefore improving classification performance overall. Moreover, this result was achieved within a very similar training time due to the one-cycle policy ensuring significantly faster training.

In contrast, the effects of the class weight are less clear than the effects of class weighting for the SVM. It is observed that while the introduction of class weights to the BERT model significantly improves recall, it simultaneously erodes precision and accuracy, leading to a decrease in performance. This indicates that further research is warranted with respect to the optimal use of class weights in this context.

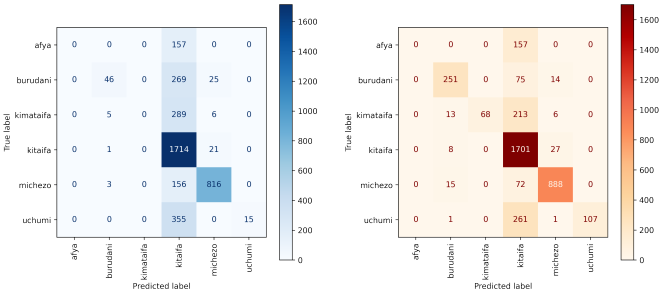
Thus, to provide a balance between the improved recall and decreased accuracy and precision, the class weights were halved to decrease their intensity. This allowed for faster training, additional epochs and improved results, with a 3% improvement over the baseline network in terms of F1-score. While the best overall accuracy was still obtained by the BERT model trained with the one cycle policy without class weighting, the combination of the one cycle policy, half class weights and four epochs attained the highest overall F1-score and therefore the best classification performance for the imbalanced problem.

B. Confusion Matrix

To provide further insight into the behaviour of each classifier, the confusion matrices before and after optimizations are plotted.

B.1 Naïve Bayes

The classification performance of the Naïve Bayes classifier before and after optimization is depicted in Fig. 10 below.



(a) Naïve Bayes

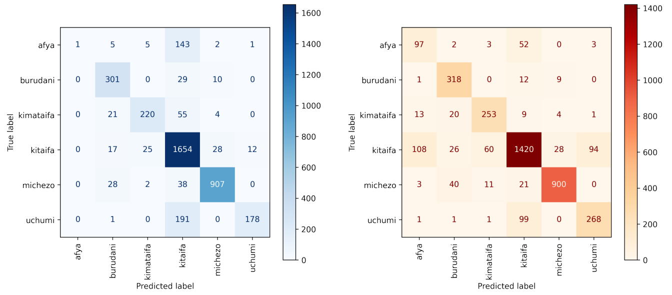
(b) Naïve Bayes + Complement

Fig. 10: Confusion matrix indicating classification performance for the unoptimized and optimized Naïve Bayes classifier.

It can be observed that accounting for class imbalance through the use of Complementary Naïve Bayes results in a great improvement in performance, with many more samples being correctly classified, as indicated by a stronger diagonal in the confusion matrix. However, the majority *kitaifa* class still dominates, especially in the case of the *afya* class where no correct classifications are made, therefore suggesting that the use of Complementary Naïve Bayes can only improve performance to a certain extent before a more complex classifier is needed.

B.1 SVM

The improvement due to the additional of class weights can be clearly observed in Fig. 11 below as the number of correctly classified samples in the poorly represented classes increases greatly. This is particularly pertinent for the *afya* class which increases from 1 correct classification to 97 correct classifications. This verifies the importance of counteracting class imbalance as the tendency of the algorithm to always predict the majority class is diminished.

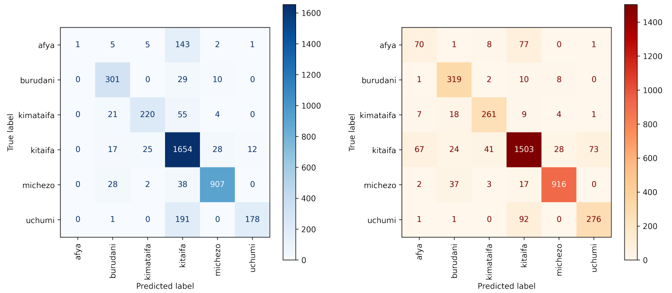


(a) SVM

(b) SVM + Class Weight

Fig. 11: Confusion matrix indicating classification performance for the unoptimized SVM and SVM optimized with class weights respectively.

Furthermore, the effects of the hyperparameters optimized through grid search are depicted in Fig. 12, with an increase in correct classifications along the diagonal leading to improved accuracy overall.



(a) SVM

(b) SVM + Grid Search

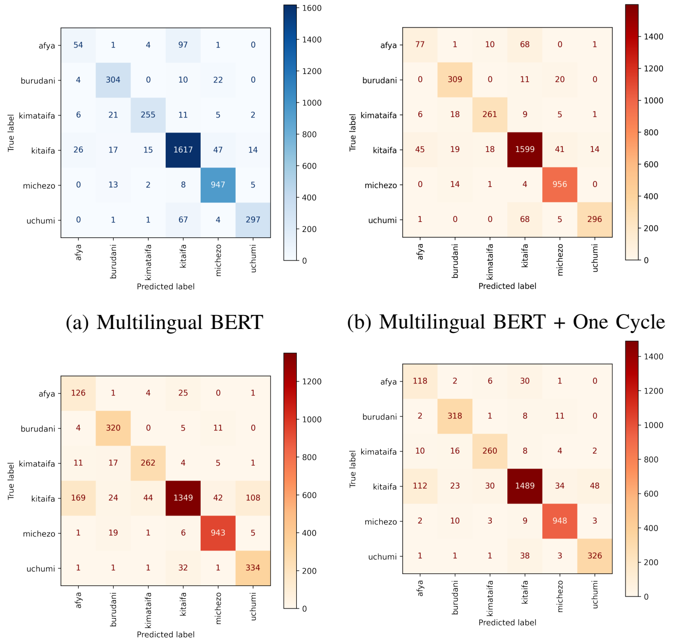
Fig. 12: Confusion matrix indicating classification performance for the unoptimized SVM and SVM optimized with grid search respectively.

B.3 Deep Learning Methods

Lastly, the effect of the different optimization techniques on the transformer model can be investigated.

It can be observed from Fig. 13 that the one cycle learning rate in isolation produces the best classification results in terms of least overall misclassifications, and thus with respect to accuracy. In contrast, the class weight strategy over-corrects slightly for the underrepresented class, causing some majority *kitaifa* samples to be classified as the least represented *afya* class.

This effect is then reduced using the half class weight strategy, as the effects of the class weighting is restrained by halving the value of each weight. This indicates that while class weights can be an effective strategy, the tendency for over-correction must be managed to ensure recall is not gained at the expense of precision, leading to a worse overall F1-score.



(a) Multilingual BERT

(b) Multilingual BERT + One Cycle

(c) Multilingual BERT + Class Weight

(d) Multilingual BERT + Half Class Weight + Additional Epochs

Fig. 13: Confusion matrix indicating classification performance for the Naïve Bayes and Support Vector Machine approaches respectively.

IX. Conclusion

Due to the widespread focus on English text as the subject of Natural Language Processing studies, there is a severe lack of research into NLP techniques for widely spoken African languages. Accordingly, this investigation explores the application of a variety of text classification techniques to a corpus of Swahili news articles, ranging from simple machine learning algorithms to more complex deep networks.

From the investigation, it has been seen that the multilingual BERT network outperforms the more probabilistic Naïve Bayes and Support Vector Machine approaches, with an optimized F1-score of 85%. However, this performance is obtained at the expense of significantly more computational resources and training time in comparison to the simpler SVM approach.

Critically, model optimization plays a pivotal role in ensuring optimal model performance across all three approaches. Due to the imbalanced nature of the dataset, specific strategies to ensure an appropriate balance of precision and recall were required for viable classification performance across the news categories. Specifically, the use of Complementary Naïve Bayes provided a strong advantage over the baseline multinomial approach, while the addition of class weighting and hyperparameters tuned through grid search allowed for an 11% increase in the SVM F1-score.

However, while the addition of class weights resulted in a clear benefit for the SVM classifier, the multilingual BERT model displayed a tendency towards over-correction in the presence of class weights. Therefore, this investigation indicates that a more subtle approach to class imbalance is required for the multilingual BERT model. Significantly, the use of the one cycle learning rate policy played a crucial role in the final BERT performance as it allowed for significantly reduced training time, additional epochs and improved learning, indicating the great value of effective learning rate strategies.

References

[1] Alexandros Kastanos and Tyler Martin. Graph convolutional network for swahili news classification. *arXiv preprint arXiv:2103.09325*, 2021.

[2] Kathleen Siminyu, Godson Kalipe, Davor Orlic, Jade Abbott, Vukosi Marivate, Sackey Freshia, Prateek Sibal, Bhanu Neupane, David Adelani, Amelia Taylor, et al. Ai4d–african language program. *arXiv preprint arXiv:2104.02516*, 2021.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Eibe Frank and Remco R Bouckaert. Naive bayes for text classification with unbalanced classes. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 503–510. Springer, 2006.

[5] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[6] Yan-Shi Dong and Ke-Song Han. A comparison of several ensemble methods for text categorization. In *IEEE International Conference onServices Computing, 2004.(SCC 2004). Proceedings. 2004*, pages 419–422. IEEE, 2004.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[9] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer, 1994.

[10] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[12] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[13] Alp Öktem, Eric DeLuca, Rodrigue Bashizi, Eric Paquin, and Grace Tang. Congolese swahili machine translation for humanitarian response. *arXiv preprint arXiv:2103.10734*, 2021.

[14] Barack Wanjawa and Lawrence Muchemi. Question answering using automatically generated semantic networks–the case of swahili questions. In *2020 IST-Africa Conference (IST-Africa)*, pages 1–8. IEEE, 2020.

[15] Casper Shikali Shivachi, Refuoe Mokhosi, Zhou Shijie, and Liu Qihe. Learning syllables using conv-lstm model for swahili word representation and part-of-speech tagging. *Transactions on Asian and Low-Resource Language Information Processing*, 20(4):1–25, 2021.

[16] Davis David. Swahili: News classification dataset, 2020.

[17] Noel Masasi and Bernard Masua. Common swahili stop-words, 2020.

[18] Eibe Frank and Remco Bouckaert. Naive bayes for text classification with unbalanced classes. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 503–510. Springer, 2006.

[19] P. Ambika. *Machine learning and deep learning algorithms on the Industrial Internet of Things (IIoT)*, volume 117, chapter 8, pages 321–338. Advances in Computers, 2020.

[20] Thorsten Joachims. A statistical learning model of text classification for support vector machines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 128–136, 2001.

[21] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[23] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[24] 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, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[25] Arun Maiya. ktrain: A low-code library for augmented machine learning. *arXiv preprint arXiv:2004.10703*, 2020.

[26] Leslie Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[27] Jason Rennie, Lawrence Shih, Jaime Teevan, and David Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.

[28] Leslie Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.