applied sciences

MDPI

Article

Meta-Learner for Amharic Sentiment Classification

Girma Neshir [1,2,*], Andreas Rauber [3,†] and Solomon Atnafu [4,†]

1 IT Doctoral Program, Addis Ababa University, Addis Ababa P.O. Box 28762, Ethiopia
2 Department of Software Engineering, Addis Ababa Science and Technology University, Addis Ababa P.O. Box 16417, Ethiopia
3 Institute of Information Systems Engineering, Technical University of Vienna, Favoritenstraße 9-11/194-01, A-1040 Vienna, Austria; rauber@ifs.tuwien.ac.at
4 Department of Computer Science, Addis Ababa University, Addis Ababa P.O. Box 1176, Ethiopia; solomon.atnafu@aau.edu.et
* Correspondence: girma1978@gmail.com or girma.neshir@aau.edu.et; Tel.: +251-913021313
† These authors contributed equally to this work.

Abstract: The emergence of the World Wide Web facilitates the growth of user-generated texts in less-resourced languages. Sentiment analysis of these texts may serve as a key performance indicator of the quality of services delivered by companies and government institutions. The presence of user-generated texts is an opportunity for assisting managers and policy-makers. These texts are used to improve performance and increase the level of customers' satisfaction. Because of this potential, sentiment analysis has been widely researched in the past few years. A plethora of approaches and tools have been developed—albeit predominantly for well-resourced languages such as English. Resources for less-resourced languages such as, in this paper, Amharic, are much less developed. As a result, it requires cost-effective approaches and massive amounts of annotated training data, calling for different approaches to be applied. This research investigates the performance of a combination of heterogeneous machine learning algorithms (base learners such as SVM, RF, and NB). These models in the framework are fused by a meta-learner (in this case, logistic regression) for Amharic sentiment classification. An annotated corpus is provided for evaluation of the classification framework. The proposed stacked approach applying SMOTE on TF-IDF characters (1,7) grams features has achieved an accuracy of 90%. The overall results of the meta-learner (i.e., stack ensemble) have revealed performance rise over the base learners with TF-IDF character *n*-grams.

Keywords: ensemble learning; Amharic sentiment classification; stacking; meta-learner; character *n*-grams

Citation: Nehsir, G.; Rauber, A.; Atnafu, S. Meta-Learner for Amharic Sentiment Classification. *Appl. Sci.* **2021**, *11*, 8489. https://doi.org/10.3390/app11188489

Received: 5 August 2021
Accepted: 5 September 2021
Published: 13 September 2021

1. Introduction

With emergence of World Wide Web (WWW) technology, the number of user-generated texts is increasing. This is helping businesses/organizations to enhance their services and products, boosting their revenue and competitiveness by increasing consumer or client satisfaction. As people are using online reviews to promote products and receive feedback about their services/products from their clients anywhere in the world, the amount of opinionated datasets is increasing drastically on a daily basis. On social media platforms, people usually use different formats, such as texts, audio, video, graphics, and images, to express their feelings and opinions about an event/service/product. Of all these, textual data are the most relevant and accessible user-generated content. Text allows social media users to express their feelings, opinions, and views towards an aspect of a product/service.

The process of identification of texts with either subjectivity or objectivity is called subjectivity detection. Subjectivity terms such as opinions, feelings, emotions, affections, and sentiments in text are usually understood as synonymous and are used interchangeably in Natural Language Processing (NLP). The word 'sentiment' refers to a highly conscious

activity of a person's judgmental view towards an issue or object. Sentiment is the consciously judgmental view of a person in favor of an issue (i.e., positive sentiment) and/or express judgmental views of a person against an issue (i.e., negative sentiment). The automated analysis of opinionated text is sentiment analysis (also called opinion mining).

However, major issues with typical NLP approaches include (i) lack of NLP resources for most languages other than English, (ii) mastering natural language understanding, and (iii) the difficulty of reasoning across large and multiple documents [1]. For cases where the language under consideration is less resourced, these problems are intractable for performing computational linguistic research. Less-resourced language problems are primarily associated with the lack of adequate and efficient tools for the processing of natural language: part-of-speech tagger, stemmer, lemmatizer, dependency parser, named-entity tagger, etc. To resolve these challenges, addressing the lack of annotated corpora is to perform specific linguistic tasks, including development of automatic annotated data, role of expert/linguistic knowledge, develop a single language or universal solutions and/or resource creation [2].

The reports in [3,4] state that half of the world's languages will vanish by the end of this century. Thus, proper attention is needed to the language professionals and the speakers of these languages. For the issues related to insufficiency of linguistic resources, techniques such as creating more labeled NLP corpora and leveraging resource-rich languages are undertaken to ease the scarcity in less resourced languages. Semi-supervised, unsupervised methods, and rule-based methods are the most popular approaches to create the required tools. For the challenges related to scarcity of labeled NLP corpora, some approaches use linguistic knowledge to seed unsupervised models and use this linguistic knowledge to adapt models and approaches of familiar languages. In [5], approaches to less-resourced languages are reviewed and suggested addressing these problems.

Amharic is one of the less-resourced languages, which is one of the Semitic languages. Amharic opinionated texts are increasing quickly. However, due to lack of labeled corpus, sentiment lexicon, and other linguistic resources, research works on sentiment classification in this language is challenging. The nature of the labeled data (such as its quality, whether it is balanced, and the type of extracted features) affects the performance of machine learning. The machine learning algorithms are assumed to properly discriminate the target sentiment classes. There are few Amharic sentiment analysis works; they are categorized into lexicon-based [6–9], machine learning-based approaches (SVM, RF,NB) [10–13], deep learning approaches (LSTM and CNN) [14,15], and those that use BERT [16].

There are many sentiment analysis research works in English and many non-English languages. Several approaches trying to transfer sentiment information from high- resourced languages such as English to other languages have been proposed. For example, Yulan et al. [17] exploited three English lexical resources—MPQA, Appraisal, and SentiWordnet—by translating them into three Chinese Sentiment Lexicons using Google translator. The performances of these generated Chinese lexicons were compared with the Chinese NTUSD sentiment lexicon by implementing them to classify sentiment on Chinese product review using SVM and Naive Bayesian classifiers.

Here, in this study, we proposed a combination of machine learning approaches (i.e., ensemble learning). Ensemble learning is a technique for combining various base learners from which a new classifier is created. The new classifier is supposed to gain performance compared to any of its constituent base learners [18]. In ensemble learning, different base learners of the same or heterogeneous types are combined using different fusing strategies (i.e., voting, averaging, and stacking) [19–21].

This work addresses the next four research questions: (1) To what extent does ensemble learning improve Amharic sentiment classification on small set of user generated texts compared to base learners? (2) Which feature representation (TF-IDF uni-gram, TF-IDF character *n*-grams) has better performance of Amharic sentiment classification with the proposed ensemble approach? (3) Does the SMOTE technique improve performance of the proposed approach by balancing the imbalanced labeled user-generated data? (4) On

which feature representation of Amharic texts does SMOTE show higher performance improvement of sentiment classification? The aim of this paper is to investigate the text features along with the effect of SMOTE techniques for Amharic sentiment classification on four sentiment-labeled user-generated data sets and to test to what extent ensemble learning is improving sentiment classification as compared to base learners.

The key contributions of this research is as follows.

- The provision of an annotated data sets [22].
- The effect of SMOTE with TF-IDF character *n*-grams feature is tested on Amharic sentiment classification by using ensemble learning.
- SMOTE with TF-IDF character *n*-grams feature works better than the one with (or without) SMOTE on performance of Amharic sentiment classification of user generated text.
- SMOTE with TF-IDF word uni-gram has shown performance gains of sentiment classification as compared to TF-IDF uni-gram with no SMOTE and
- TF-IDF character (1,7) grams is found to be the most salient feature for discriminating sentiment categories of Amharic user-generated text.

The paper is organized as follows. Section 2 surveys the related works dealing with ensemble learning for sentiment classification. Section 3 describes the materials and methods, which further describe the fusion methods of ensemble learning and followed by specifying the workflow of the proposed stacking approach. Section 4 reports the results of experiments which are carried out to evaluate the proposed ensemble learning to use different Amharic user generated data sets. The last section presents the conclusion drawn from the results of the research.

2. Literature Review

The related works associated with text feature selection, followed by balancing imbalanced data sets. The performance of ensemble learning for sentiment classification is reviewed.

2.1. Feature Selection

Feature selection plays a prominent role in the efficient and successful application of machine learning algorithms. Features are used to discriminate observations (samples) to classify into categories. To apply machine learning algorithms, the first task is feature preprocessing and feature selection. All the input features to the machine learning should be transformed into numerical forms. For example, text features need to be converted into numerical feature sets. The most common feature sets include Bag-Of-Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF), *n*-grams (character/word) language modeling, topic modeling features, and word embedding. Studies revealed that bi-gram word feature sets perform better than uni-gram word feature sets for text classification. In similar research, BOW word feature set shows better performance for distinguishing text categories compared to TF-IDF and word embedding [23].

In contrast, character *n*-grams are working effectively in different tasks of NLP, such as language identification, authorship attribution, text categorization, plagiarism detection, sentiment analysis, and so on. Character *n*-grams can also be used in a task, regardless of the language. Character *n*-grams can be used with new languages, and they can also detect rare words that are out of vocabulary (OOV) or misspelled [24–26]. According to the findings, *n*-grams with $n = 3$ are the most common. However, for a more salient *n*-grams feature, a higher value of $n >= 4$ is suggested [27]. The choice of the best text feature representation for an application like sentiment classification has no strict scientific recommendation and no alignment in the existing literature, as it depends on application data and the type of language being processed.

2.2. Imbalanced Learning

When a machine learning method applies to imbalanced data sets, it causes overfitting and reduces classification accuracy. If a machine learning algorithm is trained using highly imbalanced data sets, the model is overwhelmed by the majority class data points. This will lead to the tendency of high false negative rates [28]. In the literature, applying Synthetic Minority Oversampling TEchnique (SMOTE) has shown improvement in performance of machine learning in various application domains [29]. For example, the effectiveness of SMOTE is evaluated for balancing three botnet datasets for a malware and intrusion detection system [30]. In other research, SMOTE was applied successfully for tweet polarity classification using three publicly available datasets [31]. These are some studies applying SMOTE which have shown an improvement of recognition of minority classes.

2.3. Ensemble Approaches for Sentiment Classification

Khalid et al. [32] proposed an ensemble learning that combines gradient boosting and support vector machines using voting classifier. With term-frequency and TF-IDF (uni-, bi-, and tri-gram) features, the proposed approach is evaluated on a dataset of 64,295 Google App user reviews, which are labeled either positive, negative or neutral. Their approach has achieved the highest accuracy of 0.93 with term-frequency feature compared to the other variants of TF-IDF features and compared to individual classifiers. The authors did not mention why they chose TF-IDF word *n*-grams over TF-IDF character *n*-grams. In similar research, Wan et al. [33] assessed and compared three most popular ensemble approaches (i.e., Bagging, Boosting, and Random Subspace) relying on base learners (i.e., Naive Bayesian (NB), Maximum Entropy, Decision Tree, K-Nearest Neighbor (KNN), and SVM) for sentiment classification using BoW features of ten publicly available datasets. With 1200 comparative group experiments, the results have shown that ensemble learners achieved performance gains over the base learners for sentiment classification.

In contrast to the work in [32], the author of [33] selected BOW features over TF-IDF term features. In other work, Wan et al. [34] developed an ensemble of NB, SVM, Bayesian Net, C4.5, and RF using majority voting for sentiment classification of Airline customer service feedback. The proposed ensemble outperformed with F-score of 84.2% for three class datasets and 91.7% for two class datasets using 10 folds CV of 12,864 tweets. Yet another similar approach was done by Alnashwan et al. [35] who proposed an ensemble approach which improved sentiment classification relying on base learners (SVM, NB, LR, and RF) using seven lexical resources. For training, the base learners used sentiment lexicon features from the three tweeter datasets. The performance of the proposed ensemble learning was better than the individual learners. However, the proposed ensemble (i.e., accuracy of 81.0%) has no significant difference in performing the RF (i.e., accuracy of 82.4%).

In [36], Nazlia et al. proposed ensemble methods for the problem of subjectivity and sentiment classification of Arabic texts by combining base learners, Rochio, SVM, and NB. The results show that the ensemble classifier performed better than base learners' performance. Kennedy et al. in [37] built combined two approaches (lexicon-based and Machine Learning (ML) that uses SVM) based on weighted voting meta-classifier. Machine learning with SVM outperformed the lexicon-based method. The author recommended that combining multiple classifiers using the meta-classifier scheme could enhance the performance of sentiment classification. In a similar study, P.P. Tribhuvan et al. [38] proposed stacking ensemble model for feature-based sentiment analysis by combining SVM, NB, and KNN as base learners and SVM as meta classifier. Using the Laptop product review dataset (44 features of 4096 laptop reviews), the stack model achieved an accuracy of 92.5%. In another research, A. Hassan et al. [39] developed a bootstrap ensemble framework for sentiment classification of English Tweets.

The intention of the proposed framework is to tackle the challenges of tweeter sentiment classification, such as class imbalance, sparsity, and representational richness issues. In the framework, first text features are extracted, then seven different machine learning algorithms are trained, finally selected models are combined. This proposed approach

has shown more accurate and balanced prediction of sentiment of tweeter compared to other algorithms.

Yet another similar approach was done by Martinez et al. [40], which integrated two or more unsupervised approaches using meta-classifiers for Spanish movie review sentiment classification. Stacking classifiers (SVM, Naïve Bayesian, and Logistic Regression) were used as the meta-learners to combine the unsupervised models. The results of the integrated system using stacking (Naïve Bayesian) were performing better than the results of the individual experiment.

Table 1 summarizes the features used, the base learners used, the ensemble method used, and the sentiment analysis performance of the proposed ensemble method on associated sets of data in multiple languages (i.e., English, Arabic, and Spanish). The above works differ with our proposed method in the following aspects: we use (i) stack classifier rather than voting, (ii) TF-IDF character *n*-grams rather than other features, (iii) SMOTE strategy for balancing the datasets, and (iv) we developed preprocessing techniques for Amharic user-generated texts considering the preservation of semantics while preprocessing (such as normalization, stemming, and stop word removals).

Table 1. Summary of key related work of ensemble-based sentiment classification.

Paper	Year	Ensemble Approach	Average Accuracy/F1-Score	Languages	Domain/Dataset
[36]	2013	Ensemble methods of three base learners (SVM, NB, Rocchio)	With lexicon features, the ensemble (macro F1 of 90.95%)	Arabic	customers' reviews datasets
[32]	2020	Ensemble-based: GBSVM which combines Gradient boosting and SVM using voting.	With TF features, GBSVM outperformed with accuracy of 93%.	English	64,295 Google App user reviews
[33]	2014	Three popular ensemble methods based on five base learners (NB, ME, DT, KNN, and SVM)	With BoW terms, and TF, TF-IDF features (i.e., Uni- and Bi-grams), total of 1200 comparative group experiments (6 feature sets × 20 classifiers × 10 datasets), the highest average accuracy of the Laptop dataset is 92.62% using Random Space—ME using the bi-gram TF features.	English	ten public sentiment analysis datasets
[34]	2015	Ensemble Learning: NB, SVM, Bayesian Net, C4.5, and RF	With BoW terms, F-score of 84.2% for three class dataset and 91.7% for two class dataset	English	12864 tweets of Airline customer feedback
[35]	2016	Ensemble methods of four base learners (SVM,RF, NB, LR)	With 7 lexicon features, the ensemble (accuracy of 81.0%) is not significant compared to RF (accuracy of 82.4%)	English	three tweeter datasets
[37]	2006	Hybrid methods of (SVM and lexicon based	With valence shifter bi-grams, the hybrid performed slightly better than its constituents.	English	three tweeter datasets
[38]	2018	Ensemble of (SVM, NB and KNN)	With Feature-Opinion Negation triple, accuracy of 92.5%	English	4096 Laptop product review dataset along with 44 features
[39]	2013	Proposed Step-wise Iterative Model Selection (SIMS)	By hierarchical search process, accuracy of SIMS rises 10–20% relative to Genetic Algorithm (GA) to select the best models.	English	Three Tweeter datasets
[40]	2014	Integrated two or more lexical features using meta-classifiers	64.68% (best in the combined approach)	Spanish	3878 movie reviews (MuchoCine)

This research aims to improve Amharic sentiment classification performance by aggregating the prediction of individual-based learners using meta-learner. The input features from the existing approach are different because our approach is using TF-IDF character level (1,7) grams feature set. To the best of our knowledge, there has been no study using ensemble approach to test whether it is improving Amharic sentiment classification on user generated comments. The meta-learners are not only improving sentiment classification performance, but they also help avoid overfitting [41].

3. Materials and Methods

3.1. Overview

A method of combining machine learning classifiers is called ensemble learning (or ensemble method). Ensemble learning is a technique for combining various base learners from which a new classifier is created [22]. The new classifier is supposed to gain performance compared to any of its constituent base learners. In ensemble learning, differ-

ent base learners of the same or heterogeneous types are combined using different fusing strategies. Combining classifiers (i.e., either homogeneous or heterogeneous type) in either sequential (i.e., boosting) or in parallel (i.e., bagging) configurations aims to achieve improved classification/regression performance than the performance of individual models. Besides, the other objective of the ensemble method is to reduce variance and bias. That is, ensemble classifiers are not only designed to get a model that achieves performance gains, but also a model that can generalize well.

The most popular ensemble approaches: voting, bagging, boosting, and stacking [22,42,43]. Voting is the simplest of all fusing strategies, which takes the most frequent predicted class of multiple predictors. For sample x, class i is assigned if class i is predicted most frequently. Mathematically,

$$c_i = mode\{h_1(x); h_2(x); \ldots; h_N(x)\}. \quad (1)$$

where c_i is the predicted class, mode is statistical mode of prediction by classifiers $h_1, h_2, \ldots h_N$ for sample x. Bagging is a method for creating N base learners by corresponding N sample data generated randomly from the training dataset (with replacement). The bagging method is also called Bootstrap aggregation. Suppose from N iterations, N random samples from training data are generated. For N base learners, the final prediction of their ensemble is given by averaging all predictions from all N models using Equation (2).

$$c_i = \frac{\sum_{j=1}^{N}(h_j(x) = i)}{N}. \quad (2)$$

Popular bagging algorithms are random forest and bagging meta-estimators, just to name a few.

Boosting is an ensemble of base learners in a sequence where each classifier is started with equal weight, but after all models are trained once, weight is assigned to each model based on its performance. After model evaluations, a larger weight is assigned to a misclassified sample for providing greater focus in the next iteration, and vice versa. The final model relies on a weighted averaging method. Mathematically,

$$c_i = \frac{\sum_{i=1}^{N} \frac{(h_i w_i)}{\sum w_i}}{N}. \quad (3)$$

where $h_1, h_2, \ldots h_N$ are base learners, $w_1, w_2, \ldots w_N$ are weights, N is the number of classifiers, and h is the final classifier. It is noted that boosting considers weighting in training data which is one of the feature making it different from bagging.

Boosting classifiers include gradient boosting classifier, adaboost classifier, extreme gradient boosting classifier, and light gradient boosting classifiers. Unlike bagging classifiers, boosting classifiers are sequential, i.e., the input of the next base learner is the output of its previous base learners.

Ensemble learning is proposed to address bias–variance trade-offs on performance of classifiers. Variance is the error caused by limitation of learning data, whereas bias is caused by limitation of the algorithm itself. Boosting tries to address bias, whereas variance is addressed by bagging [44]. However, boosting is sensitive to overfitting as it tries to fit the data into the model [45,46].

Unlike voting methods, which rely on user adjusted weights, stacking (or meta-learners) can adjust their weight themselves. The proposed approach in this research is using the meta-learner for aggregating the predictions of the base learners.

3.2. Evaluation Metrics

To measure the performance of a classification system, evaluation metrics are required. The most popular metrics include accuracy, precision, recall, and F1-score. Before describing the evaluation metrics, let us define important terms. The terms such as True Posi-

tive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are defined as follows:

- (i) True Positive (TP) is the number of samples belonging to the positive class that are correctly predicted by the model.
- (ii) True Negative (TN) is the number of samples belonging to the negative class that are correctly predicted by the model.
- (iii) False Positive (FP) is the number of samples belonging to the positive classes that are wrongly predicted by the model. This is also called Type I Error.
- (iv) False Negative (FN) is the number of samples belonging to the negative classes that are wrongly predicted by the model. This is also called Type II Error.

The model's performance evaluation metrics which are used in this study are described as follows:

- (i) Accuracy (A) is the percentage of correctly predicted samples, i.e.,

$$A = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4}$$

- (ii) Precision (P) is the evaluation metric that measures the correctly predicted samples actually turned out to be positive, i.e.,

$$P = \frac{TP}{TP + FP}. \tag{5}$$

  Note that precision is a metric that measures the reliability of the model.
- (iii) Recall (R) is a measure of the number of actual positive samples which are correctly predicted by the model, i.e.,

$$R = \frac{TP}{TP + FN}. \tag{6}$$

  Recall is also called sensitivity. Note that precision is more important to tell when the model predicted more false positive samples than false negative samples. In contrast, recall is a more important metric to tell if the model predicts more false negative than false positive samples.
- (iv) F1-Score (F1) is the harmonic mean between precision and recall. It is calculated from precision and recall, i.e.,

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{7}$$

- (v) Area Under Curve(AUC) is the most popular evaluation metric for binary classification problem. AUC is a measure of the probability of a classifier that will rank a randomly chosen positive example higher than a randomly chosen negative example. AUC ranges [0,1], the higher the AUC implies, the better the model.
- (vi) Logarithmic Loss (LogLoss) is used to measure how close or far model's predicted value from actual value. For the binary classification problem, LogLoss is also called binary cross-entropy, which is the negative average of the log of corrected predicted probabilities. For N samples, LogLoss is given by

$$LogLoss = \frac{1}{N} \cdot \sum_{i=1}^{N} -(y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)) \cdot \tag{8}$$

  where y_i is the actual value, p_i is the probability of class 1, and $1 - p_i$ is the probability of class 0.
- (vii) Cross-Validation (CV): K-fold Cross-Validation is the model evaluation technique where all data set samples are randomly divided into k folds of equal size. As there

is no standard rule for selecting *k*, usually the value of *K* is 5 or 10. We chose *K* = 5. For each run, *K*-1 folds are used for training the model and the remaining are used as the testing set. This is repeated k times so that each of the folds is used once for testing. The average accuracy of the 5 models and standard deviation is returned. This is illustrated in Figure 1.

The benefits of using cross-validation include (i) its capability of avoiding overfitting/underfitting, (ii) it can evaluate model consistency by producing accuracy/error rate for all the test sets, and (iii) it has also capability of training/testing on a small data by using the total datasets [47].

- (viii) Mean of Accuracy and Standard Deviation (SD): Note that from statistics, 'mean' is defined as the average of accuracy of the models in five folds using testing sets, whereas 'variance' is the sum of the squares of the model's accuracy deviated from the mean divided by the total number of models. Larger standard deviation shows that the prediction of a model is more sensitive to future observations, and vice versa.

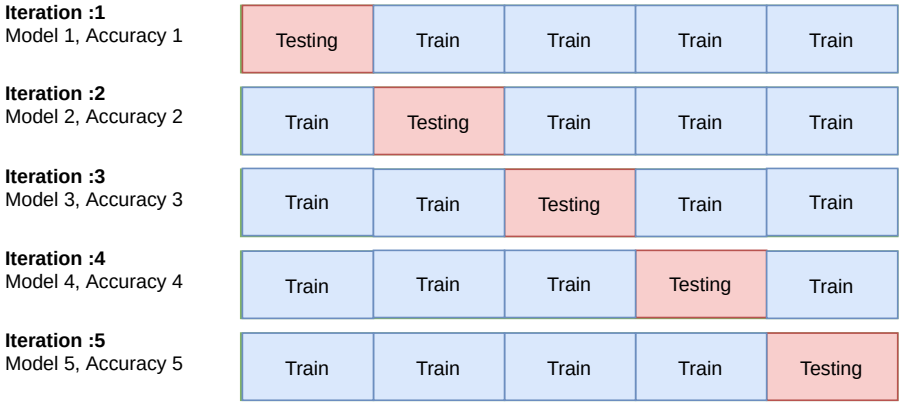

Figure 1. Five-fold cross-validation for stacking sentiment classifier.

3.3. Proposed Stacking Algorithm

This research presents sentiment classification that uses a combination of machine learning techniques like base learners. The most common machine learning algorithms (SVM, NB, and RF) are chosen as base learners with default parameters, while LR is employed as a meta-learner. The proposed ensemble learner for sentiment classification is depicted in Figure 2.

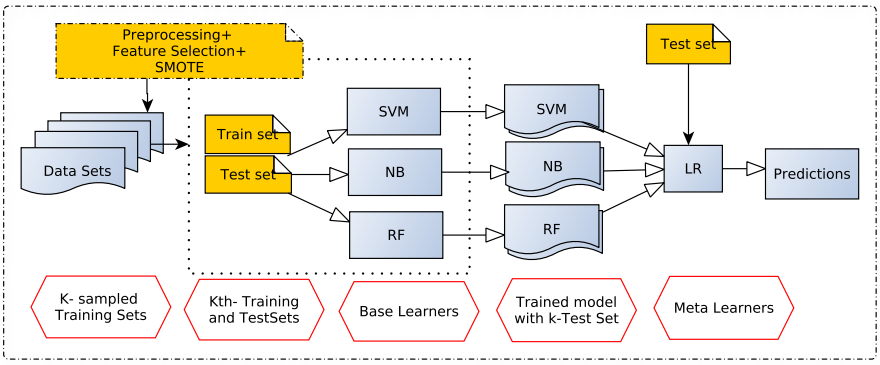

Figure 2. Proposed stacking sentiment classifier.

The steps of the workflow illustrated in Figure 2 are briefly described as follows.

(1) Data Sets Collections: Four data sets [22]—GCAO (2871), PMO(6637), EBC (2444), and ZEMEN (1440)—of comments are used for evaluation. The first three datasets are collected from Facebook comments and the fourth is collected from YouTube movie comments (i.e., Zemen Drama [48])). Specifically, the Facebook comments are collected from

(i) Facebook page of Government Communication Affairs Office (GCOA [49]), (ii) the official Facebook page of Prime Minister Office (PMO [50]), and (iii) the Facebook page of Ethiopian Broadcasting Corporate (EBC [51]). The statistical summary of the above-mentioned four datasets are depicted in Table 2.

Table 2. User-generated Amharic Comments on Social Media Dataset Description.

Dataset	#Texts	Avg.chars	Avg.words	#Samples	#POS	#NEG	Year
GCOA	2871	41.31	8.34	2871	1728	1143	2016–2017
EBC	2444	99.75	20.41	2444	1707	737	2017
PMO	6637	192.03	39.03	6637	4589	2048	2018
ZEMEN	1440	63.21	13.42	1440	490	850	2016–2018

From Table 2, we can observe that the text samples of GCOA have short length where the average number of words and average number of characters are 8 and 41, respectively. In contrast, the PMO data set has the largest average length of words and characters, i.e., 39 and 192, respectively. The features of this sample have a strong potential of discriminating and assigning it to a certain class as the length of user-generated text samples increases, whereas a shorter sample (i.e., one with fewer words) would not have enough features to discriminate it from the rest of the data set's samples. This makes it tough for a machine learning system to extract meaningful information from such a sample.

Figure 3 shows that almost all the datasets are skewed. The number of samples with negative class is less than the number of samples with positive class across all three data sets (GCOA, EBC, and PMO). The negative class samples are under-represented (i.e., minority class) in these data sets, whereas the positive class samples are over-represented (i.e., majority class). In both PMO and EBC, about 69% of the samples are from the majority class. On the other hand, the majority of the samples (63.4%) are negative class in the ZEMEN dataset. If we use machine learning techniques in this setting, the model will be biased towards the negative class. To minimize this bias, we need to use the SMOTE procedure to balance these datasets prior to using them for machine learning algorithms.

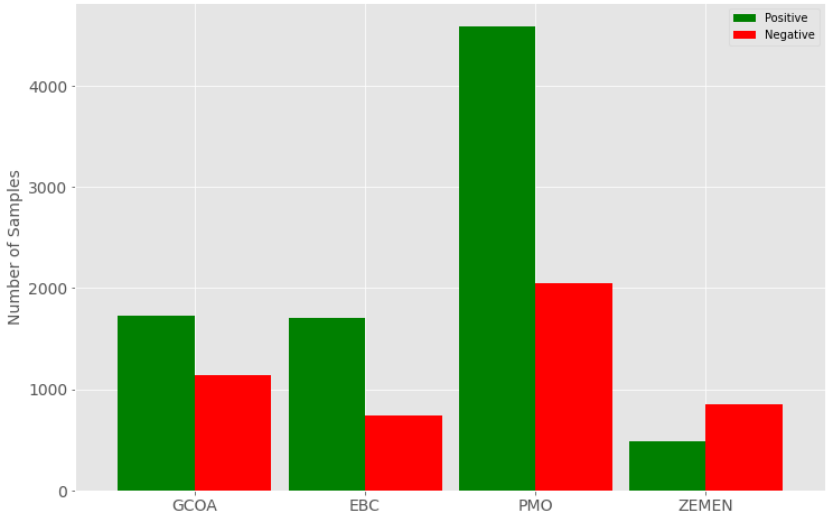

Figure 3. The Labeled Sentiment Corpora's Class Distributions.

(2) Preprocessing and Feature Extraction: As preprocessing is crucial in text mining, procedures including removing all digits, punctuation marks, and non-Amharic characters; spelling correction; stop word removal; and normalization are performed.

In this research, normalization is the process of replacing all letters with the same sound (replaced by a single letter). Because of the many spelling variants employed, different persons write certain Amharic words in various forms. For example, the word ቴሌቪዥን

('television') can be written as **ቴሌቭዠን, ቴሌቭዥን, ቴሌቪዥን** [52]. As a result, Amharic texts comprise many characters with the same sound that needs to be substituted by a single common character. That is, ('**ሀ,ሃ,ሐ,ሓ,ኀ,ኃ,ኻ→ሀ**)፤ (**ሰ,ሠ→ሰ**)፡ (**ፀ,ጸ→ፀ**)፡ (**ዐ,አ,ኣ,ዓ→ዐ**)፡ (**ቆ,ቈ,ቇ→ቆ**)፡ (**ቀ,ቊ→ቀ**)፡(**ኸ,ኈ→ኸ**)፡ (**ጎ,ጐ→ጎ**)፡ (**ኋ,ኧ,ኯ→ኋ**)), where the arrow (→) means 'replaced by'. If one of the left-hand symbols appears in the text, it is replaced by the symbol to the right of the arrow (→).

Furthermore, stop words are recognized as the top most common (i.e., redundant) tokens in the data sets. However, some words such as **አይደለም**/it is not/, **ምንም**/nothing/, **የለም**/none/, **ሳይሆን**/not happened/, and **የለበትም**/has nothing in it/ are examples of negative words in Amharic language. The performance of sentiment classification is affected when these words are included in the stop word list. As a result, these terms were not included in the stop word list.

Because Amharic is morphologically dense, we discovered that stemming removes the salient characters/most significant features/that might aid in determining a text's sentiment class [15]. Therefore, stemming is not considered in preprocessing procedures.

Text data sets are transformed into numerical features using TF-IDF vectorizations after they have been preprocessed. TF-IDF vectorization is a method of converting documents into numerical features. By combining local weights and global weights of a text, the TF-IDF feature of a document contains more discriminant information to encode texts. We compute TF-IDF by using the formula $tfidf_{t,d} = tf_{t,d} \cdot idf_{t,d} = tf_{t,d} \cdot log(\frac{N}{df_t})$, where $tf_{t,d}$ refers to the number of occurrences of term t in document d, df_t is the number of documents containing term t, and N refers to the total number of documents. $tf_{t,d}$ captures the local weights of a term t in terms term-frequency, whereas $idf_{t,d}$ captures the global weight of a term t representing feature with respect to text document d.

After applying the Grid Search algorithm to the TF-IDF vectorizer implemented in the Scikit learn Python package [53], the TF-IDF character (1,7) grams feature set and maximum features of 5000 has been chosen for optimal Amharic sentiment classification. For several NLP applications, the TF-IDF character level n-grams feature outperforms the word-level grams feature, according to the literature [25,29,54]. Specifically, character n-grams features outperform word grams features for dealing negation in Amharic sentiment classification [22]. As a result, the proposed approach is tested with the TF-IDF character (1,7) grams feature set, and the results have been compared to the TF-IDF word uni-gram feature set (as baseline).

Besides, the Synthetic Minority Oversampling Technique (SMOTE) is proposed for balancing imbalanced datasets. Employing SMOTE for balancing imbalanced datasets improves sentiment classification tasks [31]. SMOTE is also popular for balancing imbalanced non-textual data sets for other applications [30,55]. As a result, we proposed SMOTE as a strategy for balancing vectorized sentiment datasets. SMOTE augments the minority class of samples in datasets to balance out imbalanced data sets. The average accuracy of the 5-fold cross-validation (CV) is measured in each of the four data sets with and without SMOTE using both the TF-IDF character (1,7) n-grams and TF-IDF word uni-gram features.

(3) Base Learner Algorithms: Support Vector Machine (SVM), Naive Bayesian (NB), and Random Forest (RF) are the most commonly used supervised machine learning algorithms in NLP [54,56], and they were chosen for Amharic sentiment classification in this study. For the sake of simplicity and comparison, we choose Logistic Regression (LR) as a meta-learner for combining the base learners. They are briefly stated as follows.

(i) SVM is one of the most powerful supervised machine learning approaches and it is closely connected with neural networks. SVM is built on mapping and classifying data members into distinct output spaces. Support vectors are the data points that are closest to the decision hyperplane. The computational inefficiency of SVM is one of its shortcomings [57].

(ii) RF is a combination of multiple decision trees (also known as bagging) in various configurations. This should address the shortcomings of decision trees, which do not

update themselves when new training samples are supplied. Random forest is robust because it combines multiple tree classifiers which rely on a subset of the training set's input features. Finally, it decides by voting for predicting new sample. Random forest classifier is built in two phases: create several decision trees and then get predictions with those trees for test sets and finally combine their predictions by majority voting (averaging).

(iii) NB is a probabilistic method which is based on Bayes' rule, in which the input features are assumed to determine the output variable independently. Even though this method worked effectively in most times, this assumption of independence is rarely used in practice. The other strength of this algorithm is that it can learn incrementally and update its probability distribution [58,59], and (iv) LR is a statistical approach for training binary categorical classes, rather than continuous variables.

(4) Meta-Features: The meta-learner method is trained using the predicted values from the base learners. The base learners' predicted information is employed as meta-features, which are considered being essential for discriminating the target class categories.

(5) Meta-Learner Algorithm: The meta-learner is acting as a combiner of the proposed approach. However, unlike other fusing strategies, it uses a machine learning model (i.e., logistic regression in our case) rather than voting/averaging. Voting and averaging, weighted averaging combine base learners relying on Equations (1)–(3), respectively.

The procedure for the proposed ensemble learner with stack cross-validation algorithm is presented in Algorithm 1.

Algorithm 1: Proposed Ensemble Learning.

Input: Labeled Data Set
Output: Average Accuracy of Trained Meta-learner Model M

1 Create the 3 base learners (SVM, RF, and NB) and meta-learner (LR)
2 With 5-fold cross-validation, partition the training set into 5 disjoint sets.
3 **foreach** *k fold in the partitioned trainingSet* **do**
4 Split each partion into training and testing sets.
5 Apply SMOTE on the respective trainingSet.
6 Train proposed stack classifier using trainingSets.
7 Collect prediction of the trained stacked classifier using testing sets and discard the model.
8 return mean accuracy of the models on the complete 5-fold CV

Description: Algorithm 1 takes a labeled dataset as input and average accuracy is returned as an output. Three base learners (i.e., SVM, NB, and RF) and one meta-learner (i.e., LR) are created (line 1). In line 2, with a 5-fold CV, the dataset is randomly partitioned into 5 disjoint sets and stored. For each k-fold cross-validation set (lines 3–7), each *k*th disjoint set is randomly split into training and testing set (with a ratio of 80:20, respectively) (line 4). In those experiments where we want to evaluate the impact of balancing the classes, apply SMOTE to the trainingSet of the respective run (line 5). Line 6 builds the stack classifier using the trainingSet. Line 7 stores the accuracy of each of the model using the testingSet. Finally, line 8 computes the mean accuracy of the 5 models.

The proposed stack configuration is intended to improve the performance of Amharic sentiment classification by aggregating the prediction (meta-feature) of base learners.

4. Results and Discussions

For evaluating the performance of the proposed stacked classifier, four experiments have been carried out using the four data sets. The results show the performance of the proposed model for Amharic sentiment classification.

4.1. Experimental Settings

The experimental settings of the hyperparameters of machine learning algorithms for carrying out experiments are specified as SVM (C = 1, kernel = rbf), NB (alpha = 1), RF

(n_estimators = 100), LR (C = 1) and the TF-IDF vectorizer (analyzer = char, ngram_range = (1,7), max_features = 5000) and the other parameters are set to their own default settings, as we can see in Table 3.

Table 3. Hyperparameter of the base learners and the TF-IDF vectorization.

Algorithm	Hyper-Parameter	Type	Default Value	Selected Value
TF-IDFVectorizer	analyzer	discr	word	Char
	max_df	cont	1	None
	max_features	discr	None	5000
	ngram_range	disc	(1,1)	(1,7)
LR	C	cont	1	1
	alpha	cont	None	None
	average	discr	None	None
	penalty	disc	l2	l2
	power_t	cont	None	None
	tol	cont	0.0001	0.0001
Multinominal NB	alpha	con	1	1
NB	fit prior	cat	TRUE	TRUE
SVM	C	con	1	1
	coef0	con	0	0
	degree	discr	3	3
	gamma	con	scale	scale
	kernel	disct	rbf	rbf
	tol	con	0.001	0.001
Random Forest	bootstrap	discr	TRUE	TRUE
	criterion	disc	gini	gini
	max features	con	auto	auto
	min samples split	disc	2	2
	min samples leaf	discr	1	1
	n_estimators	discr	100	100

4.2. Results

We have carried out four experimental groups (I–IV). Each of the experimental groups has 80 runs (four data sets × four algorithms × 5 metrics). With all the data sets, the aggregated results of those experiments undertaken are reported. In each experimental group, the mean of the results of each metrics along with its SD for each algorithm (three base learners and one stack classifier) are computed and shown in the Table 4.

The four experimental groups include experiment I (Exp I), experiment II (Exp II), experiment III (Exp III), and experiment IV (Exp IV). Those experiments are grouped into four, relying on the features sets used and whether SMOTE is applied. Experiment I is undertaken in all four datasets using TF-IDF character (1,7) with maximum features of 5000 without application of SMOTE technique, whereas experiment II has the same settings to experiment I, but in this case with the application of SMOTE technique for balancing the data sets. In contrast, both experiments III and IV are undertaken with TF-IDF word unigram feature sets in all the data sets. Unlike experiment III, experiment IV makes use of SMOTE strategy for balancing the data sets.

Table 4. Comparison of Performance of Ensemble Classifier over base learners using TF-IDF character (1,7) grams with (no) SMOTE and TF-IDF word uni-gram with (no) SMOTE and using CV of 5 folds relying on annotated user comments data (i.e., all four data sets). CoS = TF-IDF character (1,7) + NoSMOTE, CS=TF-IDF character (1,7) + SMOTE, WoS = TF-IDF word uni-gram + NoSMOTE, and WS = TF-IDF word uni-gram + SMOTE, Exp = Experiment. The numeric values which are formatted bold showing high performance values of the respective classifiers.

Model	Metric	Exp I: CoS	Exp II: CS	Exp III: WoS	Exp IV: WS
		Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
SVM	Accuracy	0.78 ± 0.02	**0.85** ± 0.07	0.72 ± 0.01	0.74 ± 0.02
	Recall	0.70 ± 0.02	0.85 ± 0.07	0.64 ± 0.01	0.74 ± 0.02
	Precision	0.79 ± 0.03	**0.87** ± 0.06	0.70 ± 0.02	0.75 ± 0.03
	F1	0.71 ± 0.02	0.85 ± 0.07	0.64 ± 0.02	0.73 ± 0.02
	AUC	0.84 ± 0.02	**0.94** ± 0.04	0.72 ± 0.03	0.78 ± 0.04
	LogLoss	-	5.02 ± 2.38	9.63 ± 0.46	9.11 ± 0.85
RF	Accuracy	0.76 ± 0.02	0.83 ± 0.07	0.72 ± 0.02	0.73 ± 0.03
	Recall	0.69 ± 0.02	0.83 ± 0.07	0.65 ± 0.02	0.73 ± 0.03
	Precision	0.76 ± 0.03	0.84 ± 0.06	0.69 ± 0.02	0.75 ± 0.04
	F1	0.70 ± 0.03	0.83 ± 0.07	0.65 ± 0.02	0.73 ± 0.03
	AUC	0.81 ± 0.02	0.92 ± 0.06	0.74 ± 0.02	0.81 ± 0.04
	LogLoss	-	5.77 ± 2.33	9.81 ± 0.53	9.18 ± 1.16
NB	Accuracy	0.76 ± 0.02	0.79 ± 0.03	0.72 ± 0.02	0.68 ± 0.02
	Recall	0.69 ± 0.02	0.79 ± 0.03	0.64 ± 0.02	0.68 ± 0.02
	Precision	0.76 ± 0.02	0.79 ± 0.03	0.72 ± 0.03	0.69 ± 0.02
	F1	0.70 ± 0.02	0.79 ± 0.03	0.64 ± 0.02	0.68 ± 0.02
	AUC	0.82 ± 0.02	0.87 ± 0.03	0.75 ± 0.02	0.77 ± 0.02
	LogLoss	-	7.35 ± 1.18	9.51 ± 0.52	11.06 ± 0.82
Stack	Accuracy	0.79 ± 0.02	**0.86** ± 0.06	0.73 ± 0.01	0.74 ± 0.03
	Recall	0.74 ± 0.02	0.86 ± 0.06	0.64 ± 0.01	0.74 ± 0.03
	Precision	0.77 ± 0.02	**0.87** ± 0.05	0.71 ± 0.03	0.75 ± 0.04
	F1	0.75 ± 0.02	0.86 ± 0.06	0.64 ± 0.02	0.73 ± 0.03
	AUC	0.84 ± 0.02	**0.94** ± 0.04	0.76 ± 0.02	0.82 ± 0.03
	LogLoss	-	**4.85** ± 1.95	9.49 ± 0.45	9.13 ± 1.11

4.3. Discussions

In this subsection, we discuss the above-mentioned results from four aspects, which are presented as follows.

(i) Effects of SMOTE: As we can see in Table 4, the application of SMOTE on imbalanced data sets is improving performance of all the classifiers (i.e., both stack classifier and base learners) on sentiment classification of Amharic user generated texts. With character *n*-grams feature, the results of stack classifier with the five evaluation metrics (accuracy of 86%, recall of 86%, precision of 87%, F1 of 86%, and AUC of 94%) has revealed that the use of SMOTE is significantly improving sentiment classification performance over the other experimental settings. Figure 4 is visually revealing that application of SMOTE has shown a rise in performance when it is used in both character *n*-grams (those bars in green) and word gram feature sets (those bars in orange).

(ii) Effects of Features: As shown in Table 4, the use of the character *n*-grams features increases the performance of sentiment classification of all the classifiers while comparing them to the word *n*-grams features regardless of application of SMOTE. The result of all the classifiers trained with TF-IDF character *n*-grams (as shown Exp I) has better performance over classifiers trained with TF-IDF word uni-grams (as shown Exp III). For example, the stack classifier in experiment I has achieved better results in accuracy of 79%, recall of 74%, precision of 77%, F1 of 75%, and AUC of 84% as compared to the results in experiment III (accuracy of 73%, recall of 64%, precision of 71%, F1 of 64%, and AUC of 76%). To clearly analyze the result in Table 4, Figure 5 is plotted to show that similar performance gains have achieved by all the individual classifiers with character *n*-grams features (those bars

in green and blue). As a result, the results revealed that character *n*-grams feature is most salient feature which helps the classifiers to discriminate the sentiment class categories of Amharic texts.

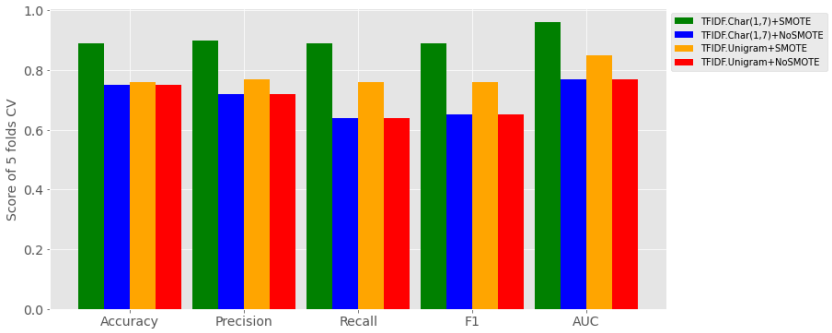

Figure 4. Comparison of the Performance of Proposed Stacking Sentiment Classifier using different features (TF-IDF character (1,7) grams + with(no) SMOTE) and (TF-IDF uni-grams + with(no) SMOTE) using all four data sets.

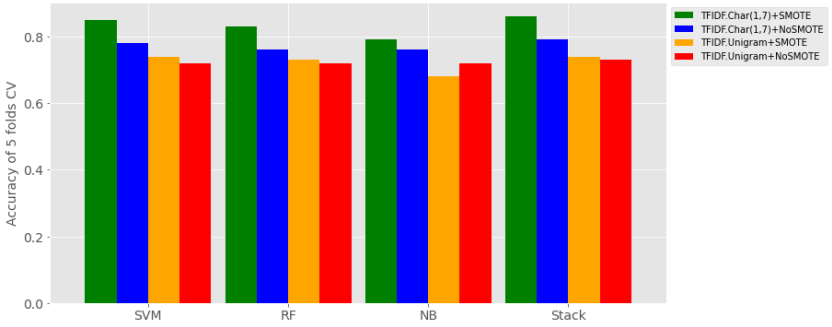

Figure 5. Comparison of the Performance of Proposed Stacking Sentiment Classifier with base Machine Learning Classifier using all four Data Sets.

(iii) Comparison of Classifiers: As depicted in Table 4, in each experiment, the stack classifier has better performance over the base learners. For example, experiment IV with TF-IDF uni-gram feature with application of SMOTE, stack learner has achieved better accuracy of 74%, recall of 74%, precision of 75%, F1 of 73%, and AUC of 82% over the base learners' performance with the same settings. However, SVM has a comparable performance of accuracy of 74%, recall of 74%, precision of 75%, F1 of 73%, and AUC of 78%. Besides, SVM has less log loss of 9.11 as compared to stack classifier's log loss 9.13 with SD of $\pm$1.11 which is a bit deviated as compared to the deviation of SVM (i.e., $\pm$0.85) in the same experimental settings. Figure 6 shows that the average scores of five metrics of all classifiers' performances of sentiment classification on all data sets. As a result, it is clear to look for the difference that performance of NB classifier is poorly performed in all data sets, whereas SVM is performing better than the other base learners (i.e., RF and NB).

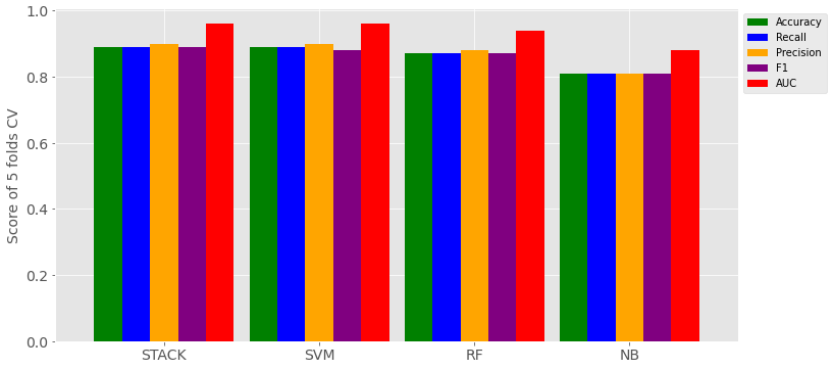

Figure 6. Comparison of the Performance of Proposed Stacking Sentiment Classifier with base Machine Learning Classifier using all four data sets with character(1,7) + SMOTE based on five metrics (accuracy, precision, recall, F1 score, and Area Under the Curve (AUC)).

(iv) Effects of the Nature of the Data sets: As we can see from a statistical summary of the data sets in Table 2, the nature of the four data sets have varying length of samples, number of sample size, and the type of majority class in the imbalanced data sets [22]. Figure 7 shows the overall performance of the classifiers across the four data sets using average log loss. All classifiers have least loss on the PMO data set (those bars in blue) as compared to the loss on other data sets. This is because the PMO dataset has a larger sample length and larger number of samples. This would help to extract several features, either character or word grams, which help the machine learning classifiers to differentiate each sentiment class category of Amharic texts. In contrast, all classifiers' performance in any of the experimental settings has the worst sentiment classification performance on ZEMEN data sets over the classifiers' the performance. The highest log loss (those bars in red) is recorded as it is illustrated in Figure 7. This is because of ZEMEN dataset (i) has smaller average length of samples, (ii) has smaller number of samples, and (iii) is of a different domain compared to the other data sets (ZEMEN is movie domain whereas the other three datasets are related to political Facebook comments).

In summary, the nature of the data sets (such as the size, the length, quality of samples, and the domain) has a direct effect on the classifiers' performance. For example, we observed that as the length of samples and the size of the datasets is larger, the classifiers' performance are also getting better for sentiment classification performance [22].

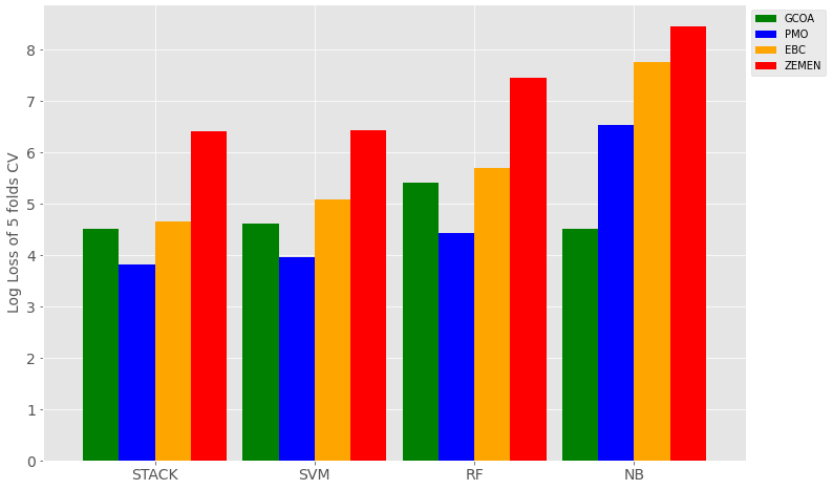

Figure 7. Comparison of the performance of the proposed stacking sentiment classifier with base machine learning classifier across the four data sets using TF-IDF character (1,7) grams + SMOTE using log loss.

5. Conclusions

The main purpose of this study is to investigate building a stacking strategy with a meta-learner using TF-IDF character (1,7) gram feature sets (or using SMOTE) with a CV of five folds for Amharic sentiment classification relying on base learners (i.e., SVM, NB, and RF) across four datasets. Finally, we found that the suggested stack model outperformed the base classifiers for sentiment classification of user-generated text in social media using character/word *n*-grams features with and without application of the SMOTE technique. In conclusion, employing SMOTE approaches to balance datasets using TF-IDF character/word *n*-grams features increased the performance of stack classifiers across datasets when compared to base learners.

Of all the individual base learners, SVM has achieved the highest performance across all the data sets compared to the other base learners. As SVM involves the cost function parameter, which regulates the balance between bias and variance by tuning it depending on the training set, SVM also has a kernel function which maps non-separable data into linear or nonlinear separable problem relying on distance maximization. RF has better performance than NB as RF considers not only the cost criterion function, but it also has an ensemble of multiple tree learners. In contrast, NB performed the worst on all the data sets. One of the reason for this is that NB has the shortcoming of independence assumption; this means it assumes features are independent, which is not the case, specifically for the character/word *n*-grams features.

The key contribution of this research is as follows: (i) using character *n*-grams, the stacking of base classifiers with meta-learners has improved the performance of sentiment classification across all datasets compared to the base learners; (ii) using character/word *n*-grams with and without SMOTE procedure, the stack classifier has also performed better than the base classifiers' performance across all the datasets; and (iii) as stacking classifier is a combination of heterogeneous classifiers, it has also performing better than a combination of homogeneous classifiers.

Below is the highlights of the answers for the corresponding research questions raised in Section 1.

- (1) To what extent does ensemble learning improve Amharic sentiment classification on a small set of user-generated texts compared to base learners? The answer for this research question is reported all four experimental settings (i.e., Experiment I–IV), where its results are reported in Table 4. That means, the proposed stack ensemble model outperforms the individual learners on all sentiment classification data sets using TF-IDF character (1, 7) gram with and without application of SMOTE shown in Table 4, as compared to the data sets using TF-IDF uni-gram with and without SMOTE balancing technique. The proposed stack learner has achieved a rise of accuracy ranging from 10% to 31% over all base learners in the above experimental settings.
- (2) Which feature representation (TF-IDF uni-gram, TF-IDF character *n*-grams) has better performance of Amharic sentiment classification with the proposed ensemble approach? The answer for this research question is provided in Table 4 that reveals ensemble learner outperforms on TF-IDF character *n*-grams over TF-IDF word uni-gram.
- (3) Does SMOTE technique improve performance of the proposed approach by balancing the imbalanced labeled user generated data? The answer to this research question is reported in Table 4, where SMOTE has significantly improved sentiment classification of ensemble learners and base learners when it is applied with both TF-IDF character *n*-grams and TF-IDF word uni-gram feature sets.
- (4) On which feature representation of Amharic texts does SMOTE show higher performance improvement of sentiment classification? The answer is provided in Table 4 showing that SMOTE boosts sentiment classification when it applied to TF-IDF character *n*-grams.

The answers provide a strong indication of the design choices for developing sentiment analysis solutions. However, as the data sets are small, rigorous evaluation of larger data sets is required to see how well these datasets hold in more data-rich settings.

For further research, the hyperparameters of base learners need to be optimized before being combined with meta-learners, which will have potential performance gains over the proposed stacked setting in this research. The proposed approach can also be improved by preparing a sentiment data sets which have more fine-grained sentiment categories.

Author Contributions: Conceptualization, G.N. and A.R.; methodology, G.N.; software, G.N.; validation, A.R., G.N. and S.A.; formal analysis, G.N.; investigation, G.N.; resources, G.N.; data curation, S.A.; writing—original draft preparation, G.N.; writing—review and editing, A.R.; visualization, G.N.; supervision, A.R.; project administration, G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: It is available in zenodo data sharing repository at https://doi.org/10.5281/zenodo.5005968. (accessed on 30 July 2021).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
CV	Cross-Validation
DT	Decision Tree
EBC	Ethiopian Broadcasting Corporate
GCOA	Government Communication Office Affair
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
LR	Logistic Regression
ME	Maximum Entropy
MPQA	Multi-Perspective Question Answering
NB	Naive Bayes
NTUSD	National Taiwan University Semantic Dictionary
PMO	Prime Minister Office
RF	Random Forest
SD	Standard Deviation
SVM	Support Vector Machine
SMOTE	Synthetic Minority Oversampling Technique
TF-IDF	Term Frequency-Inverse Document Frequency

References

1. Ruder, S.; Korashy, H. The 4 Biggest Open Problems in NLP. *Ain Shams Eng. J.* Available online: https://ruder.io/4-biggest-open-problems-in-nlp/ (accessed on 2 March 2021).
2. Palmer, A. Computational Linguistics for Low-Resource Languages. Slide Presentation, Saarland University, Saarbrücken, Germany. Available online: http://www.coli.uni-saarland.de/courses/CL4LRL (accessed on 2 October 2020).
3. Lam, K.; Al Tarouti, F.; Kalita, J. Creating Lexical Resources For Endangered Languages. In Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages, Baltimore, ML, USA, 26 June 2014; pp. 54–62.

4. Janse, M. Language Death And Language Maintenance: Problems And Prospects. In *Language Death and Language Maintenance: Theoretical, Practical And Descriptive Approaches*; Janse, M., Tol, S., Eds.; John Benjamins: Amsterdam, The Netherlands, 2003; pp. 9–17. [CrossRef]
5. King, B. Practical Natural Language Processing for Low-Resource Languages. Ph.D. Thesis, Department of Computer Science, University of Michigan, Michigan, MI, USA, 2015.
6. Gebremeskel, S. Sentiment Mining Model for Opinionated Amharic Texts. Master's Thesis, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia, 2010.
7. Tilahun, T. Linguistic localization of opinion mining from Amharic blogs. *Int. J. Inf. Technol. Comput. Sci. Perspect.* **2014**, *3*, 890.
8. Alemneh, G.N.; Rauber, A.; Atnafu, S. Dictionary Based Amharic Sentiment Lexicon Generation. In Proceedings of the International Conference on Information and Communication Technology for Development for Africa, Bahir Dar, Ethiopia, 28–30 May 2019; pp. 311–326. [CrossRef]
9. Alemneh, G.N.; Rauber, A.; Atnafu, S. Corpus based Amharic sentiment lexicon generation. In Proceedings of the SA Forum for Artificial Intelligence Research, Published at CEUR Workshop Proceedings (CEUR-WS.org), Cape Town, South Africa, 3–6 December 2019.
10. Philemon, W.; Mulugeta, W. A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts. *HiLCoE J. Comput. Sci. Technol.* **2014**, *2*, 8.
11. Dessalew, C. Public Sentiment Analysis for Amharic News. Master's Thesis, Bahir Dar University, Bahir Dar, Ethiopia, 2019.
12. Mihret, M.; Atinaf, M. Sentiment Analysis Model for Opinionated Awngi Text. In Proceedings of the African Conference (AFRICON), Accra, Ghana, 25 September 2019; pp. 1–6. [CrossRef]
13. Tsegaw, M. Sarcasm Detection for Amharic Text. Master's Thesis, Bahir Dar University, Bahir Dar, Ethiopia, 2020.
14. Alemu, Y. Deep Learning Approach For Amharic Sentiment Analysis. Master's Thesis, University of Gondar, Gonder, Ethiopia, 2018.
15. Fikre, T. Effect of Preprocessing on Long Short Term Memory-based Sentiment Analysis for Amharic Language. Master's Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2020.
16. Neshir, G.; Atnafu, S.; Rauber, A. BERT Fine-Tuning for Amharic Sentiment Classification. In Proceedings of the Workshop RESOURCEFUL Co-Located with the Eighth Swedish Language Technology Conference (SLTC), University of Gothenburg, Gothenburg, Sweden, 25 November 2020.
17. He, Y.; Harith, A.; Zhou, D. Exploring English Lexicon Knowledge For Chinese Sentiment Analysis. In Proceedings of the Canadian Information Processing Society (CIPS)-SIGHAN Joint Conference on Chinese Language Processing, Beijing, China, 28–29 August 2010.
18. Opitz, D.; Maclin, R.; Brown, D. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.* **1999**, *11*, 169–198. [CrossRef]
19. Ganaie, M.A.; Hu, M.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *arXiv* **2021**, arXiv:2104.02395.
20. Brownlee, J. A Gentle Introduction to Ensemble Learning Algorithms. Available online: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/ (accessed on 25 May 2021).
21. Ensemble Methods: Combining Multiple Models to Improve the Desired Results. Corporate Finance Institute. Available online: https://corporatefinanceinstitute.com/resources/knowledge/other/ensemble-methods/ (accessed on 25 May 2021).
22. Alemneh, G.N.; Rauber, A.; Atnafu, S. Negation Handling for Amharic Sentiment Classification. In Proceedings of the 4th Widening Natural Language Processing Workshop, Seattle, WA, USA, 8 January 2020; pp. 4–6. [CrossRef]
23. Hofmann, M.; Chisholm, A. *Text Mining and Visualization: Case Studies Using Open-Source Tools*; CRC Press: Boca Raton, FL, USA, 2016.
24. Veres, C.; Kapustin, P.; Veres, C. Enhancing Subword Embeddings with Open *n*-grams. In *Natural Language Processing and Information Systems*; NLDB 2020, Lecture Notes in Computer Science; Métais, E., Meziane, F., Horacek, H., Cimiano, P., Eds.; Springer: Berlin, Germany, 2020; Volume 12089, pp. 3–15. ISBN 978-3-030-51309-2. [CrossRef]
25. Graovac, J.; Kovačević, J.; Pavlović-Lažetić, G. Language Independent n-gram-based Text Categorization with Weighting Factors: A Case Study. *J. Inf. Data Manag.* **2015**, *6*, 4.
26. Piskorski, J.; Jacquet, G. TF-IDF Character *n*-grams versus Word Embedding-based Models for Fine-grained Event Classification: A Preliminary Study. In Proceedings of the Workshop on Automated Extraction of Socio-political Events from News, Marseille, France, 11–16 May 2020; pp. 26–34.
27. Kruczek, J.; Kruczek, P.; Kuta, M. Are n-gram Categories Helpful in Text Classification? In *Computational Science—ICCS 2020*; Lecture Notes in Computer Science; Krzhizhanovskaya, V.V., Závodszky, G., Lees, M.H., Dongarra, J.J., Sloot, P.M.A., Brissos, S., Teixeira, J., Eds.; Springer: Berlin, Germany, 2020; pp. 524–537. ISBN 978-3-030-50416-8. [CrossRef]
28. Thai-Nghe, N.; Gantner, Z.; Schmidt-Thieme, L. Cost-sensitive Learning Methods for Imbalanced Data. In Proceedings of the 2010 International Joint Conference On Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
29. Padurariu, C.; Breaban, M. Dealing with Data Imbalance in Text Classification. *Procedia Comput. Sci.* **2019**, *159*, 736–745. [CrossRef]
30. Gonzalez-Cuautle, D.; Hernandez-Suarez, A.; Sanchez-Perez, G.; Toscano-Medina, L.; Portillo-Portillo, J.; Olivares-Mercado, J.; Perez-Meana, H.; Sandoval-Orozco, A. Synthetic Minority Oversampling Technique for Optimizing Classification Tasks in Botnet and Intrusion Detection System Datasets. *Appl. Sci.* **2020**, *10*, 794. [CrossRef]

31. Ah-Pine, J.; Soriano-Morales, E. A Study of Synthetic Oversampling for Twitter Imbalanced Sentiment Analysis. In Proceedings of the Workshop on Interactions Between Data Mining and Natural Language Processing (DMNLP), Riva del Garda, Italy, 23 September 2016.
32. Khalid, M.; Ashraf, I.; Mehmood, A.; Ullah, S.; Ahmad, M.; Choi, G. GBSVM: Sentiment Classification from Unstructured Reviews Using Ensemble Classifier. *Appl. Sci.* **2020**, *10*, 2788. [CrossRef]
33. Wang, G.; Sun, J.; Ma, J.; Xu, K.; Gu, J. Sentiment Classification: The Contribution of Ensemble Learning. *Decis. Support Syst.* **2014**, *57*, 77–93. [CrossRef]
34. Wan, Y.; Gao, Q. An Ensemble Sentiment Classification System of Twitter Data for Airline Services Analysis. In Proceedings of the International Conference On Data Mining Workshop (ICDMW), Atlantic City, AC, USA, 14–17 November 2015; pp. 1318–1325. [CrossRef]
35. Alnashwan, R.; O'Riordan, A.; Sorensen, H.; Hoare, C. Improving Sentiment Analysis through Ensemble Learning of Meta-level Features. In Proceedings of the 2nd International Workshop on Knowledge Discovery on the WEB, Cagliari, Italy, 8–10 September 2016; p. 1748.
36. Omar, N.; Al-Moslmi, M.; Al-Shabi, A.Q.; Al-Moslmi, T. Ensemble of Classification Algorithms for Subjectivity and Sentiment Analysis of Arabic Customers' Reviews. *Int. J. Adv. Comput. Technol.* **2013**, *5*, 77.
37. Kennedy, A.; Inkpen, D. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Comput. Intell.* **2006**, *22*, 110–125. [CrossRef]
38. Tribhuvan, P.; Bhirud, S.; Deshmukh, R. Stacking Ensemble Model for Polarity Classification in Feature Based Opinion Mining. *Indian J. Comput. Sci. Eng.* **2018**, 9. [CrossRef]
39. Hassan, A.; Abbasi, A.; Zeng, D. Twitter Sentiment Analysis: A bootstrap ensemble framework. In Proceedings of the International Conference on Social Computing, Alexandria, VA, USA, 8–14 September 2013; pp. 357–364.
40. Artinez-Cámara, E.; Martín-Valdivia, M.; Molina-González, M.; Perea-Ortega, J. Integrating Spanish Lexical Resources by Meta-classifiers For Polarity Classification. *J. Inf. Sci.* **2014**, *40*, 538–554. [CrossRef]
41. Sagi, O.; Rokach, L. Advanced Review Ensemble learning: A survey. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]
42. Zhou, Z. *Ensemble Methods: Foundations and Algorithms*; Chapman: Orange, CA, USA, 2019.
43. Mujtaba, H. Ensemble Learning with Stacking and Blending. Mygreatlearning. Available online: https://www.mygreatlearning.com/blog/ensemble-learning-with-stacking-and-blending/ (accessed on 23 April 2021).
44. Singh, J.; Singh, G.; Singh, R. Optimization of sentiment analysis using machine learning classifiers. *Hum. Cent. Comput. Inf. Sci.* **2017**, *7*. [CrossRef]
45. Raschka, S. *Python Machine Learning Unlock Deeper Insights into Machine Learning with this Vital Guide to Cutting-Edge Predictive Analytics*; Packt Publishing: Birmingham, UK, 2015; p. 456.
46. Raschka, S. MLxtend: Providing Machine Learning and Data Science Utilities and Extensions to Python Scientific Computing Stack. *J. Open Source Softw.* **2018**, *3*, 638. [CrossRef]
47. Badvelu, J. Cross-Validation for Classification Models. Analytics. Vidhya. Available online: https://medium.com/analytics-vidhya/cross-validation-for-classification-models-9bb6506dee00 (accessed on 12 October 2020).
48. ZEMENTV. Sparks Film Production. Available online: https://www.youtube.com/channel/UCzfrWFpc5sgVyybMHp5b5sQ (accessed on 14 February 2021).
49. GCAO Ethiopia. Available online: https://www.facebook.com/gcao.ethiopia (accessed on 12 May 2019).
50. Office of the Prime Minister-Ethiopia. Available online: https://www.facebook.com/PMOEthiopia/ (accessed on 25 May 2020).
51. Ethiopian Broadcasting Corporation. Available online: https://www.facebook.com/EBCzena (accessed on 2 June 2020).
52. Kelemework, W. Automatic Amharic Text News Classification: A Neural Networks Approach. *Ethiop. J. Sci. Technol.* **2013**, *6*, 127–137.
53. Scikit-Learn Machine Learning in Python. Available online: https://scikit-learn.org/stable/ (accessed on 2 June 2019).
54. Rezapour, M. Sentiment classification of skewed shoppers' reviews using machine learning techniques, examining the textual features. *Eng. Rep.* **2021**, *3*, e12280. [CrossRef]
55. Chawla, N.; Bowyer, K.; Hall, L.; Kegelmeyer, W. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
56. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. *Information* **2019**, *10*, 150. [CrossRef]
57. Medhat, W.; Hassan, A.; Korashy, H. Sentiment Analysis Algorithms and Applications: A Survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [CrossRef]
58. Llombart, O. *Using Machine Learning Techniques for Sentiment Analysis*; Final Project, Computer Engineering, School of Engineering (EE); Universtat Automata De Barcelona (UAB): Barcelona, Spain, 2016.
59. Brownlee, J. *Master Machine Learning Algorithms: Discover How They Work and Implement Them from Scratch*; Machine Learning Mastery. 2016. Available online: https://bbooks.info/b/w/5a7f34e12f2f40dc87fbfda06a584ef681bc5300/master-machine-learning-algorithms-discover-how-they-work-and-implement-them-from-scratch.pdf (accessed on 1 September 2021).