*SACJ* 32(2) December 2020
Research Article

# Improved semi-supervised learning technique for automatic detection of South African abusive language on Twitter

Oluwafemi Oriola, Eduan Kotzé

Department of Computer Science and Informatics, University of the Free State, South Africa

**ABSTRACT**
Semi-supervised learning is a potential solution for improving training data in low-resourced abusive language detection contexts such as South African abusive language detection on Twitter. However, the existing semi-supervised learning methods have been skewed towards small amounts of labelled data, with small feature space. This paper, therefore, presents a semi-supervised learning technique that improves the distribution of training data by assigning labels to unlabelled data based on the majority voting over different feature sets of labelled and unlabelled data clusters. The technique is applied to South African English corpora consisting of labelled and unlabelled abusive tweets. The proposed technique is compared with state-of-the-art self-learning and active learning techniques based on syntactic and semantic features. The performance of these techniques with Logistic Regression, Support Vector Machine and Neural Networks are evaluated. The proposed technique, with accuracy and F1-score of 0.97 and 0.95, respectively, outperforms existing semi-supervised learning techniques. The learning curves show that the training data was used more efficiently by the proposed technique compared to existing techniques. Overall, n-gram syntactic features with a Logistic Regression classifier records the highest performance. The paper concludes that the proposed semi-supervised learning technique effectively detected implicit and explicit South African abusive language on Twitter.

**Keywords:** South Africa, Twitter, abusive language, machine learning, semi-supervised learning

**Categories:** • **Computing computing** ∼ **Natural language processing and sentiment analysis** • *Computing methodologies* ∼ *Text classification and information extraction*

**Email:**
Oluwafemi Oriola OriolaO@ufs.ac.za, oluwafemi.oriola@aaua.edu.ng,
Eduan Kotzé KotzeJE@ufs.ac.za (CORRESPONDING)

**Article history:**
Received: 30 May 2020
Accepted: 23 October 2020
Available online: 08 December 2020

## 1 INTRODUCTION

Abusive language is defined as an oral or textual expression that contains dirty words or phrases (Ibrohim & Budi, 2018). It is often inciteful or may be derogatory. The abusive language, especially hate speech and cyberbullying, may be directed towards an individual,

Oriola, O. and Kotzé, E. (2020). Improved semi-supervised learning technique for automatic detection of South African abusive language on Twitter. *South African Computer Journal 32*(2), 56–79. https://doi.org/10.18489/sacj.v32i2.847

*SACJ* is a publication of the South African Institute of Computer Scientists and Information Technologists. ISSN 1015-7999 (print) ISSN 2313-7835 (online).

a group of people, or an organization. In the case of profanity, however, there may be no direct target at all. Over the last decades, South Africa has experienced an upsurge in various degrees of violent behaviour (Kotzé et al., 2020), such as violent protests and xenophobic attacks, which have led to the loss of human lives and material resources. Many of these violent incidents can be attributed to the fast-spreading of inciteful and abusive comments, perpetrated through social media networks. However, a tool to check the soaring volumes of these kinds of communication has not yet been developed.

Machine learning is a viable tool for the automatic detection of abusive language, such as profanity, offensive words, hate speech and cyberbullying, on social media (Fortuna & Nunes, 2018). There are three machine learning techniques, namely supervised, unsupervised and semi-supervised machine learning. The most widely used of these three is supervised machine learning (Schmidt & Wiegand, 2017), but it performs poorly when applied to datasets with small amounts of labelled data. Unsupervised machine learning techniques are used to observe the relationship between features by relying on the similarities among the data or on probabilistic approaches. They have been mostly applied to zero resourced problems (Kamper et al., 2017). When the first two methods are combined, a technique called semi-supervised learning is created, which can be applied to datasets with small amounts of labelled data and large amounts of unlabelled data (Gunasekara & Nejadgholi, 2018; Khatri et al., 2018). However, semi-supervised learning techniques all rely on active learning and pseudolabels, which are inadequate or computationally expensive.

Algorithms based on semi-supervised learning techniques can harness a large amount of unlabelled data to produce effective performances comparable with state-of-the-art supervised learning techniques (Tran, 2019). Semi-supervised machine learning can be categorised into self-learning and active learning approaches. In the self-learning approach, unlabelled data is automatically annotated and the instances with high confidence are added to the training datasets iteratively. The self-learning approach can be categorised into self-training methods (Livieris et al., 2018; Tran, 2019) and generative self-learning methods such as cluster-and-label approach (Albalate et al., 2010; Kumar et al., 2017). Active learning was developed to improve the selection process of unlabelled samples and to solve the class imbalance problem, but it often relied on manual or randomised methods to select representative labelled data (Chegini et al., 2019).

## 1.1 South African Abusive Language

In South Africa, communication through social media is mostly in English but can be code-mixed with indigenous languages such as Afrikaans, isiZulu, isiXhosa, Sesotho or slang peculiar to South Africa. Recent investigation by Peace Tech Lab in South Africa[1] indicated that racially divisive comments were mainly propagated via social media before the 2019 elections. This report claims that hateful post increased by one hundred and seventy percent in this period. A large percentage of the comments were propagated via Twitter despite the available tools

[1] https://www.peacetechlab.org/south-africa-report-6.

that combat abusive language. This is as a result of inadequate lexical resources and high cost of annotation for automatic detection of South African abusive tweets.

The following are examples of the abusive and non-abusive South African tweets (with translation where necessary):

- Did Orania participate on Elections2019? (Non-abusive tweets: All English)
- Why did the DA white voter base vote ff+. They didnt want Miamane leading them and yena he dreams of equal South Africa. Vuka muntu omnyama. (Abusive tweet: English and isiZulu)

  *Why did the DA white voter base vote ff+. They didnt want Miamane leading them and he dreams of equal South Africa. Wake up black man.*
- There's a Xhosa chick somewhere asking for Imali yokuvota (Abusive tweet: English and isiZulu)

  *There's a Xhosa chick somewhere asking for money to vote.*
- Needa ask yo momma how you should treat a man with polish..fuck wrong wit em (Abusive tweet: English and non-standard English)

  *Need to ask your mother how you should treat a man with polish..fuck wrong with them.*
- If you don't understand that what trevornoah just said is dangerous and can justify a war then your short sighted as fuck, just because he mentioned Julius Malema in a bad way you forget that he literally just mentioned Genocide in South Africa. wake up black child (Abusive Tweet: All English)
- To all the moffies and gays, don't post your mom! You don't appreciate them (Abusive tweet: All English)
- Let's not make this about religion. The Bible says moffies must die. Not very tolerant. Neither should immigrants and slaves fight about how to practice their religions. At least the KhoiSan didn't make a moer of a noise with kerkklokke or adhan. Né? (Abusive tweet: English and Afrikaans )

  *Let's not make this about religion. The Bible says gays must die. Not very tolerant. Neither should immigrants and slaves fight about how to practice their religions. At least the KhoiSan didn't make a moer of a noise with church bell or Islamic call to prayer. No?*
- Small parties have a role to play in election process, analyst says. (Non-abusive tweets: All English)

The abusiveness of a tweet might be explicit or implicit. The following are examples of explicit abusive tweets (with translation where necessary):

https://doi.org/10.18489/sacj.v32i2.847

- AfterVotingIExpect Xhosa women to stop cheating. Phela Xhosa women don't just cheat. They cheat mercilessly, they show no mercy. The kinda cheating that when you find out, you have no choice but to join the church choir. (English and non-standard English)

  *AfterVotingIExpect Xhosa women to stop cheating. Truly, Xhosa women don't just cheat. They cheat mercilessly, they show no mercy. The kind of cheating that when you find out, you have no choice but to join the church choir.*

- Can't believe people believe whites over umuntu omnyama (English and isiZulu)

  *Can't believe people believe whites over black man*

The following are examples of implicit abusive tweets (with translation where necessary):

- We are not a patriotic country. That's why you offer people land for saying 1 Sotho word. That's why Jan van Riebeck and and friends just took everything. (All English)

- Remember the battle of Isandlwane, blood river mAfrika okhokho bekhusela umhlaba (English and isiZulu)

  *Remember the battle of Isandlwane, blood river where African ancestors protected the land*

Based on the above tweets, it is clear that using a classifier to distinguish between abusive and non-abusive tweets will require vast amounts of human-annotated datasets, making it a very expensive activity. Some of the publicly available abusive language resources can be found in Hatebase (Tuckwood, 2017), which is a dictionary of slurs. The resources, however, for detecting South African abusive language with large feature space are limited and inadequate. This paper, therefore, focuses on enhancing the small amount of labelled tweets to improve the performance of classifiers by using a semi-supervised learning technique.

## 2 LITERATURE REVIEW

Much research has been conducted on the automatic detection of abusive language, such as hate speech (Yuan et al., 2016; Zimmerman et al., 2018), cyberbullying (Kargutkar & Chitre, 2020), racist (Greevy & Smeaton, 2004), sexist (Park et al., 2018), and toxic contents (van Aken et al., 2018), using machine learning techniques. However, the same cannot be said of semi-supervised learning techniques. Khatri et al. (2018) proposed a two-stage semi-supervised technique to bootstrap large-scale web data for automatic toxic language detection. They developed a blacklist to rank online discussion forums by level of sensitivity. Through the random sampling of ten million most sensitive and non-sensitive utterances, a training sample was built and used to train a bidirectional long short-term memory recurrent neural network. Gunasekara and Nejadgholi (2018) developed a multi-label classifier to detect types and level of toxicity in online content. They found that by leveraging on word embeddings,

semi-supervised learning with pseudolabels derived from gradient boosted decision trees performed better than recurrent neural networks, attention mechanism and stacking of classifiers.

As our proposed model is building on the cluster-and-label generative method because they addressed the missing data problem, existing literature relevant to this study will now be reviewed.

Kumar et al. (2017) applied the cluster-and-label approach to cross-domain adaptation problem, in which unlabelled data in a source domain was merged with unlabelled data in the target domain and clustered using Fuzzy K-means algorithm. Labels were assigned to the clusters using common knowledge from experts, while the classification to predict target dataset was carried out using a dual margin binary hypersphere-based support vector machine. Albalate et al. (2010) applied the labels of the labelled samples to the clusters of the unlabelled data using optimum cluster labelling approach of Hungarian algorithm and removed uncertainty through silhouette cluster pruning. Leng et al. (2014) proposed an adaptive semi-supervised clustering algorithm with label propagation to label unlabelled dataset. The available labels of the labelled samples were used to assign labels to the unlabelled data based on K-Nearest Neighbour to core objects defined by the adaptive threshold. The adaptive threshold was estimated by the density of each cluster that the label data point belonged to. Also, new clusters were detected by the distance from the clusters core objects.

Peikari et al. (2018) clustered labelled and unlabelled datasets and mapped out the high-density regions in the data space. Fuzzy C-Means was used to assign labels to the identified clusters, while Support Vector Machine was used to label the data on the low-density region. Forestier and Wemmert (2016) focused on how multiple clustering algorithms can be combined with a supervised learning algorithm to achieve better results than classical semi-supervised and supervised algorithms. They proposed supervised learning ensembles with multiple clustering. The clustering combined labelled and unlabelled objects, and maximised intra-cluster similarity using multiple observations.

The above semi-supervised learning approaches rely on the labels of the labelled data to assign labels to the unlabelled data despite the class-imbalance nature of the labelled data and partially matched features of the labelled and testing data, which occurs in many real-life scenarios (Lee & Grauman, 2009). Considering the limitation of the labelled data, this paper seeks to improve the labelling of the unlabelled data by reducing the bias towards the labelled data.

## 3 PROPOSED SEMI-SUPERVISED LEARNING TECHNIQUE

The semi-supervised learning method proposed in this work is motivated by the following assumptions which indicate that labelling decision cannot be skewed towards labelled data, in the context of South African abusive language.

- Features of the testing data might not be an exact match of the features of either the labelled or the unlabelled data.

https://doi.org/10.18489/sacj.v32i2.847

- Datasets of similar contexts share asymmetrical features.

## 3.1 Classification problem

Let X be a set of n tweet samples $x_i \in X$. Given a binary-class classification problem with L being the very low labelled instances and U being a large set of unlabelled data such that $U > L$; the set of labelled instances $L = \{(x_1, y_1), ..., (x_l, y_l)\}$ and the set of unlabelled instances $U = \{x_{l+1}, ..., x_{l+u}\}$, where $y = (0, 1)$ are the class values of the data.

Since the objective of semi-supervised learning is to build a classification model based on the training dataset, we define our approach as presented in equation (1).

$$y = C_X(x) : y \in \{0, 1\} \tag{1}$$

where C is the Classifier (Schmidtler et al., 2008)

The schematic diagram in Figure 1 depicts the semi-supervised learning process. This involves three procedures: labelling of unlabelled data as described in Section 3.2 and 3.3, training of merged unlabelled and labelled data, and testing with test data as described in Section 4.3.3. The classification makes use of different classifiers to evaluate syntactic and semantic features.

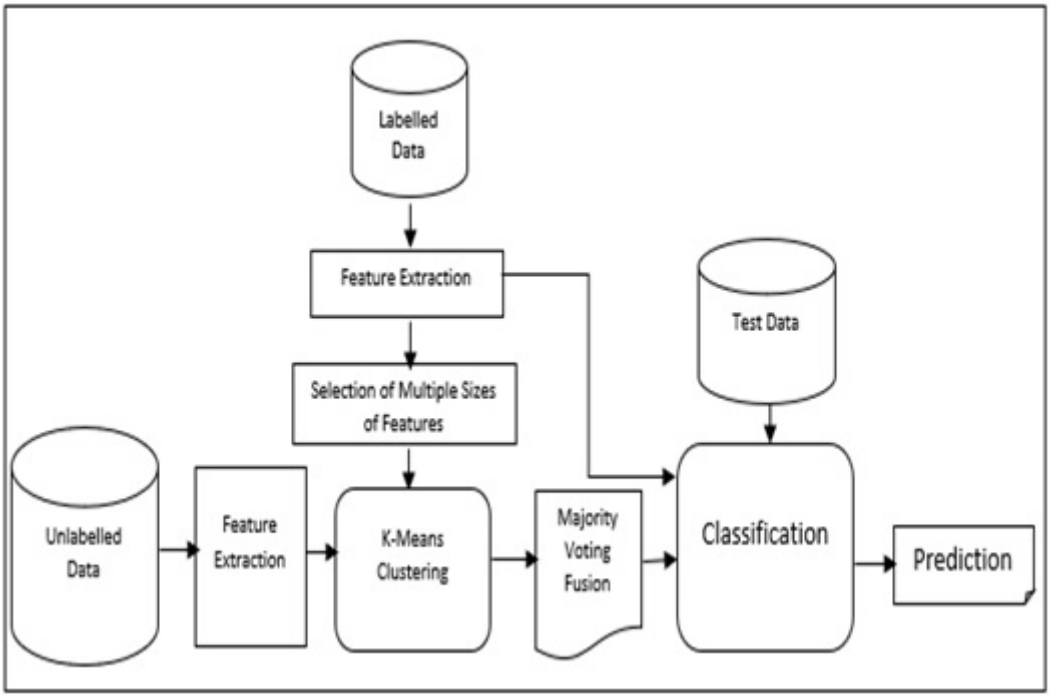


Figure 1: Proposed semi-supervised learning process

https://doi.org/10.18489/sacj.v32i2.847

## 3.2 Clustering

To apply both unlabelled and labelled data samples as training data without skewness towards labelled data, the features of the unlabelled data are fused with different sizes of features from labelled data.

Given that R is the set of features of unlabelled data U and S is the set of features of labelled data, $R = r_{l+1}, ..., r_{l+u}$ and $S = s_1, ..., s_l$

Then, $L \in (a_1, a_2, ..., a_q)$ and $a_i < a_{i+1}$, where $a \in A$ is the linearly selected set of features and q is the size of features.

By Matrix Multiplication,

$$R^0 = R * S \tag{2}$$

Applying K-Means algorithm to cluster partition of $R^0$ into k disjoint clusters $C = 0, 1$ in $t_i \in T$ given that $k - y_i = 0$, we get J sets of cluster partitions $(j_1, j_2, ..., j_i)$

$$j_i = argmin_T \sum_{i-1}^{k} \sum_{r \in t_i} \|R^0 - \mu_i\| \tag{3}$$

$\mu$ is the mean of data points in the clusters.

The pseudocode is presented below:

**Algorithm 1** Clustering Steps

1: Get Labelled data features R; unlabelled data instances U
2: **for** $R = L + 1, L + 2, ..., L + U$ **do**
3: **for** $q_i = 3, 5, 10, 15, 20, 25, 30$ **do**
4: **for** $S = 1, 2, ..., L$ **do**
5: $R^0 = R * S$; $j_i = argmin_T \sum_{i-1}^{k} \sum_{r \in t_i} \|R^0 - \mu_i\|$; $V_T = j$ ▷ $V_T$ are the labels for unlabelled data instances

## 3.3 Fusion and labelling

To assign labels to the unlabelled data, majority vote rule is applied to every instance of $R^0$ in A such that $V_T > 0$.

The selected label

$$y_i = C_u(max(V_t)) \tag{4}$$

https://doi.org/10.18489/sacj.v32i2.847

Where $R^0 = r_{l+1}, \ldots, r_{l+u}$ , $V_T =$ labels for each instance for all sizes of features that were tested.

The pseudocode is presented below:

**Algorithm 2** Fusion and Labelling Steps

1: Labels for unlabelled data instances, $V_T$
2: **for** $T = 1, \ldots, t$ **do**
3: $maxcount = Max(AllCounters)$
4: **if** $maxcount > \frac{t}{2}$ **then**
5: $V_m =$ class label corresponding to maxcount ▷ Majority Voting labels, $V_m$
6: **end if**

## 4 EXPERIMENTAL SET-UP

This section presents the experimental steps used to evaluate the technique. These steps include data collection and annotation, data pre-processing, data processing and performance evaluation. The performance metrics used for the evaluation are also presented.

### 4.1 Data collection and annotation

A total of 21,350 tweets of South African discourses on Twitter between the period of May 5, 2019 and May 13, 2019 were collected using the Twitter Archival tool, a Google Sheets plugin, based on Twitter Search API. A report[2] by Peace Tech Lab that indicated an increase of racially divisive tweets during the 2019 South African national election, motivated the collection of targeted tweets related to this period. Tweets that contained non-English words, except the names of individuals, towns, people and organizations, were removed. Retweets and repeated tweets, as well as tweets with empty word characters, were also removed. After these filters have been applied, 10,245 tweets from 2,624 users remained. The tweets were randomly divided into three data samples, namely labelled, unlabelled and testing data. A total of 1,697 tweets were randomly selected for annotation, while the remaining 8,548 tweets were not annotated. The selected samples were annotated as either 'abusive' (A) or 'non-abusive' (NA) by two experienced annotators from both white and black South African communities. A total of 1,690 tweets were selected on label agreement. The following guidelines were used for the annotation:

- Abusive annotation (A) is assigned to a tweet if it contains derogatory terms, profane words, inciteful comments or discriminatory meanings.

[2]https://www.peacetechlab.org/south-africa-report-6.

- Non-abusive annotation (NA) is assigned to a tweet if it contains neither derogatory terms, profane words, inciteful comments nor discriminatory meanings.

Cohen's Kappa agreement score (Fleiss et al., 1969) was 0.8490, indicating almost excellent agreement. The seven (7) tweets that were disagreed upon were added to the unlabelled dataset, thus, increasing it to 8,555. The 8,555 data included 8,128 non-abusive tweets ($95.01\%$) and 427 abusive tweets ($4.99\%$). The remaining 1,690 tweets were divided into 338 labelled tweets ($NA = 286$ '$84.62\%$' and $A = 52$ '$15.38\%$') and 1,352 testing data ($NA = 1118$ '$82.69\%$' and $A = 234$ '$17.31\%$') testing data. In total, unlabelled data accounted for $83.50\%$, labelled data for $3.30\%$ and testing data for $13.20\%$ of the entire 10,245 dataset used for the model. The resulting distribution of the datasets is presented in Table 1.The seven tweets (with translations) that were disagreed upon are presented below:

- I have voted for LAND. #Umhlaba (English and isiZulu tweet)

  *I have voted for LAND. #land*

- Abelungu hey!!! (English and isiZulu tweet)

  *Whites hey!!!*

- This is so true.... but also unfortunately as black children we were never taught that it's ok to talk. That sweeping under the carpet shit just needs to go. I'm not saying abelungu don't have problems but from when they are kids they are made comfortable with therapy. Healing!!! (English and isiZulu tweet)

  *This is so true.... but also unfortunately as black children we were never taught that it's ok to talk. That sweeping under the carpet shit just needs to go. I'm not saying the whites don't have problems but from when they are kids they are made comfortable with therapy. Healing!!!*

- Who are the 21 people that voted for EFF in Orania? I thought it's a strictly YT racist settlement. (All English tweet)

- O mopedi, you give your kid a Zulu name... Uthandeka makenzani umtana (English and isiZulu tweet)

  *Mopedi, you give your kid a Zulu name… Love what you do Utana*

- See me looking like trash on campus, leave me be. (All English tweet)

- Niggers ain't see the chest hair? (All English tweet)

https://doi.org/10.18489/sacj.v32i2.847

Table 1: Distribution of datasets

| Dataset | Sample | Number of Instances | Non-abusive (NA) | Abusive (A) |
|---|---|---|---|---|
| Training | Unlabelled data | 8,555 | 8,128 | 427 |
| | Labelled data | 338 | 286 | 52 |
| Testing | Testing data | 1,352 | 1,118 | 234 |

## 4.2 Data pre-processing

Various stages of pre-processing were performed on the samples of the dataset before they were suitable for text processing. These stages included the removal of unwanted terms such as usernames, punctuation, special characters, symbols, emoticons, emojis, hash symbols in hashtags and English stop words. Stemming was performed to avoid duplicate terms, and all upper case texts were changed to lower case.

## 4.3 Data processing

Three major stages of data processing were involved, namely feature engineering, clustering and classification.

### 4.3.1 Feature engineering

The texts in the tweets were transformed into Term Frequency and Inverse Document Frequency (TF-IDF) feature space, where weights were created as indicated in equation (5) (Forestier & Wemmert, 2016). TF-IDF was chosen over Bag of Words (BoW) because TF-IDF considers the IDF of each term unlike BoW and performed better than most surface-level feature representations (Lee & Grauman, 2009). The TF-IDF weights for a given term t in a document d is given as:

$$TF - IDF(t,d) = TF(t,d) * IDF(t) \tag{5}$$

when $IDF(t) = log[\frac{n}{(DF(t)+1}]$, $n =$ total number of documents in the document set; $DF(t) =$ document frequency of t.

### 4.3.2 Clustering step

We employed TF-IDF vectorization on the features of the labelled and unlabelled data, without over or under-sampling. The important features of the labelled samples were extracted using the Chi-Square statistics (Davidson et al., 2017) with K values of 3, 5, 10, 15, 20, 25 and 30.

https://doi.org/10.18489/sacj.v32i2.847

This was followed by fusing the features of the labelled and unlabelled samples as described in equation (4). The K-means unsupervised learning algorithm (number of clusters = 2) was used to cluster the fused samples, resulting in seven different cluster samples of two cluster partitions each. By applying the majority voting rule presented in equation (4), the most reliable cluster partitions were obtained. The *abusive* label was assigned to the cluster partition with more abusive words, while the *non-abusive* label was assigned to the cluster partition with fewer abusive words.

### 4.3.3 Classification step

Two types of features, namely syntactic features and semantic features were evaluated using the classical classifier and the neural network models for benchmark purpose. The syntactic features were extracted to capture the word-to-word relationships in the tweets, while the semantic features were extracted to evaluate the relationships among words in contexts.

The syntactic features were evaluated using Logistic Regression (LogReg) and Support Vector Machine (SVM) classifiers because of their impressive performances in (Davidson et al., 2017; Gaydhani et al., 2018) while the semantic features were evaluated using neural network models because of their performance in (Yuan et al., 2016; Zimmerman et al., 2018). Since two classes (abusive and non-abusive) were involved, it was classified as binary.

We handcrafted syntactical features to create a sparse vector space and used word and character n-gram features as well as part-of-speech (PoS) features derived from Penn Treebank in NLTK (Bird et al., 2009), for their performances in (Fortuna & Nunes, 2018). The word n-gram models included unigram, unigram and bigram, and unigram, bigram, and trigram word features. The character n-gram models included character n-gram with length sizes from 2 to 6 (2-to-6), 3 to 7 (3-to-7), and 4 to 8 (4-to-8). We applied n-gram features weighted by TF-IDF vectorization on the combination of semi-supervised labelled data and the originally labelled data samples. They were used because of their effective performance in previous text classification problems (Gaydhani et al., 2018; Zampieri et al., 2019). For the PoS features, unigram weighted by TF-IDF vectorization was used. The SMOTE oversampling technique (Chawla et al., 2002) was applied to reduce class imbalance. The testing data sample was also transformed in the same manner. SVM (kernel = linear kernel) and LogReg (kernel = liblinear) classifiers were used to train the merged data samples and detect abusive tweets from the testing data sample.

To select the best training model, grid-search hyperparameter optimization over several variables were followed using 10-fold cross-validation. For the SVM classifier, different C-regularization values ranging from 0.001 to 1000 were tested. For the LogReg classifier, both L1 and L2 penalty functions with np.logspace values over -4, 4 and 20 were tested.

Since the sparse vector space of syntactical features cannot provide information about the context of the words, we also experimented with semantic features such as word embeddings (Mikolov et al., 2013). Word2Vec skip-gram model (W2V) (Mikolov et al., 2013) was used to create a 300-dimensional dense vector space with word embeddings as features. The model

was implemented with minimum word count of 1, context window size of 10, workers of 8 and word vector dimensionality of 300 features.

The word embedding features were evaluated using neural network models. The first neural network was a simple feed forward neural network without convolution block (NN), while the second was a neural network with a 1-D convolution block (CNN). The CNN model consisted of a 1-D convolution block connected to max pooling. The max pooling was connected to the output layer by a ReLU and softmax activation functions. The dimension depth was 128. The neural models were implemented with maximum feature size of 20000, trainable (true), batch size of 64, epoch of 100, and softmax activation. The models were trained with ADAM optimizer (Da, 2014) and binary cross-entropy with accuracy metric. To validate the training process, ten percent (10%) of the training data was used as the validation dataset.

To select the best training model, grid-search hyperparameter optimization over several variables was followed using 10-fold cross-validation. For the SVM classifier, different C-regularization values ranging from 0.001 to 1000 were tested. For the LogReg, both L1 and L2 penalty functions with np.logspace values over -4, 4 and 20 were tested.

## 4.4 Performance evaluation

The proposed semi-supervised learning technique (proposed method) illustrated above was compared with the baseline supervised learning model (SL), active learning in method A, cluster-and-label in method B and self-supervised learning in method C.

- SL: This is a supervised learning method, in which only labelled data was used for training data.
- Method A: This is a semi-supervised learning method involving active learning, with the seed training dataset being randomly (Danka & Horvath, 2018) or manually selected. This was with the aim of improving the generalization of training data. This method entailed the selection of more generalised representative training sets from large chunks of training sets. However, the process of making the right selection is non-trivial as being practised.
- Method B: This is a cluster and label approach (Leng et al., 2014), where available labels of the labelled samples were used to assign labels to the unlabelled data based on k-nearest neighbour. It was proposed that the label data points should be the same as the majority k-nearest neighbour. It used a threshold that was generated based on the cluster to expand the neighbours of each labelled data. New clusters were detected by using the distance between the clusters.
- Method C: This is a semi-supervised learning method, with label propagation (Iscen et al., 2019). It produces impressive results with deep neural networks. The method is based on the assumption that similar examples should have same labels. The authors in [34] employed a transductive label propagation approach that was based on the manifold

assumption. The pseudo-labels assigned to the unlabelled data were generated from the entire dataset.

## 4.5 Performance metrics

Precision, Recall, F1_score and Accuracy are metrics used to evaluate the performance of the proposed semi-supervised learning method. The performance metrics are defined as presented in equations (6) to (9). The equations rely on the true positive (TP), which is the number of correctly predicted non-abusive tweets; true negative (TN), which is the number of correctly predicted abusive tweets; false positive (FP), which is the number of incorrectly predicted non-abusive tweets; and false negative (FN), which is the number of incorrectly predicted abusive tweets.

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

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

$$F1\_score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{8}$$

$$Accuracy(A) = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{9}$$

# 5 RESULTS

The following methods were implemented in Python 3.6 and several libraries. NLTK (Bird et al., 2009) was used for preprocessing; Scikit-learn (Pedregosa et al., 2011) was used for classification with classical classifiers while Gensim (Rehurek & Sojka, 2011) and Keras (Chollet, 2018) were used for the classification with neural networks.

The results of accuracy, precision, recall and F1-score for the proposed method based on word n-gram, character n-gram and part-of-speech (PoS) features with Logistic Regression and Support Vector Machine classifiers are presented in Table 2 and Table 3, respectively in comparison with the performances of SL, method A, method B and method C. The results for the evaluation of W2V with NN and CNN are presented in Table 4 and Table 5, respectively.

The results in Table 2 showed that the proposed method with character n-gram recorded the highest accuracy of 0.97, precision of 0.95, recall of 0.94 and F1-score of 0.95 followed

by word n-gram with accuracy of 0.96, precision of 0.94, recall of 0.94 and F1-score of 0.94. The proposed method with PoS, however, performed worse than the word and character n-gram features for SL, method A, method B and method C having recorded accuracy of 0.76, precision of 0.67, recall of 0.75 and F1-score of 0.68. Nevertheless, it performed better than other methods for the same PoS.

The results in Table 3 showed that the proposed method with character n-gram recorded the highest accuracy of 0.97, precision of 0.95, recall of 0.93 and F1-score of 0.94 followed by word n-gram with accuracy of 0.96, precision of 0.94, recall of 0.90 and F1-score of 0.92. The accuracy of method C was however higher than the accuracy of proposed method with word n-gram but lower for other metrics. The precision of method B was also higher than word and character n-gram features but lower for other metrics. The proposed method with PoS performed worse than the word and character n-gram features for SL, method A, method B, method C having also recorded accuracy of 0.77, precision of 0.67, recall of 0.76 and F1-score of 0.69. Nevertheless, it performed better than other methods for the same PoS.

The results in Table 4 showed that method B, with word2vec recorded the highest accuracy of 0.94, precision of 0.91, recall of 0.87 and F1-score of 0.89 followed by the proposed method with W2V, which recorded the highest accuracy of 0.93, precision of 0.88, recall of 0.88 and F1-score of 0.88. Surprisingly, method A recorded the lowest precision, recall and F1-score.

The results in Table 5 showed that method C, with W2V recorded the highest accuracy of 0.96 in line with (Iscen et al., 2019) but performed poorly with precision of 0.47, recall of 0.50 and F1-score of 0.49. Also, method A recorded the lowest precision, recall and F1-score. In general, the performances of the CNN models were poor.

Table 2: Test results of the LogReg model

| Technique | Feature | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SL | Word | 0.85 | 0.79 | 0.71 | 0.73 |
| | Char | 0.87 | 0.81 | 0.77 | 0.79 |
| | PoS | 0.61 | 0.55 | 0.57 | 0.54 |
| Method A | Word | 0.83 | 0.71 | 0.58 | 0.59 |
| | Char | 0.85 | 0.81 | 0.59 | 0.61 |
| | PoS | 0.60 | 0.58 | 0.62 | 0.55 |
| Method B | Word | 0.93 | 0.86 | 0.92 | 0.88 |
| | Char | 0.95 | 0.92 | 0.92 | 0.92 |
| | PoS | 0.71 | 0.63 | 0.70 | 0.63 |
| Method C | Word | 0.94 | 0.69 | 0.80 | 0.73 |
| | Char | 0.95 | 0.75 | 0.77 | 0.76 |
| | PoS | 0.64 | 0.53 | 0.67 | 0.47 |
| Proposed Method | Word | 0.96 | 0.94 | 0.94 | 0.94 |
| | Char | **0.97** | **0.95** | **0.94** | **0.95** |
| | PoS | 0.76 | 0.67 | 0.75 | 0.68 |

https://doi.org/10.18489/sacj.v32i2.847

Table 3: Test results of the SVM model

| Technique | Feature | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SL | Word | 0.83 | 0.87 | 0.58 | 0.60 |
| | Char | 0.80 | 0.40 | 0.50 | 0.44 |
| | PoS | 0.59 | 0.56 | 0.60 | 0.53 |
| Method A | Word | 0.83 | 0.76 | 0.52 | 0.50 |
| | Char | 0.84 | 0.84 | 0.53 | 0.52 |
| | PoS | 0.57 | 0.58 | 0.62 | 0.53 |
| Method B | Word | 0.95 | 0.93 | 0.90 | 0.91 |
| | Char | 0.96 | 0.96 | 0.91 | 0.93 |
| | PoS | 0.71 | 0.63 | 0.70 | 0.63 |
| Method C | Word | 0.95 | 0.72 | 0.72 | 0.72 |
| | Char | 0.97 | 0.87 | 0.73 | 0.78 |
| | PoS | 0.63 | 0.53 | 0.65 | 0.46 |
| Proposed Method | Word | 0.96 | 0.95 | 0.90 | 0.92 |
| | Char | **0.97** | **0.95** | **0.93** | **0.94** |
| | PoS | 0.77 | 0.67 | 0.76 | 0.69 |

Table 4: Test results of the NN model

| Technique | Feature | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SL | W2V | 0.76 | 0.62 | 0.61 | 0.62 |
| Method A | W2V | 0.82 | 0.41 | 0.50 | 0.45 |
| Method B | W2V | 0.94 | 0.91 | 0.87 | 0.89 |
| Method C | W2V | 0.90 | 0.62 | 0.76 | 0.66 |
| Proposed Method | W2V | **0.93** | **0.88** | **0.88** | **0.88** |

Table 5: Test results of the CNN model

| Technique | Feature | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SL | W2V | 0.76 | 0.43 | 0.48 | 0.44 |
| Method A | W2V | 0.83 | 0.41 | 0.50 | 0.45 |
| Method B | W2V | 0.82 | 0.63 | 0.52 | 0.50 |
| Method C | W2V | 0.96 | 0.47 | 0.50 | 0.49 |
| Proposed Method | W2V | **0.82** | **0.46** | **0.50** | **0.46** |

https://doi.org/10.18489/sacj.v32i2.847

Table 6: Information Gain for the top twenty most significant and least significant features

| SN | Most Significant BoW | IG | Least Significant BoW | IG |
|---|---|---|---|---|
| 1 | Africa | 0.1727 | Away | 0.0000120 |
| 2 | Anc | 0.0642 | Ball | 0.0000118 |
| 3 | African | 0.0171 | Bought | 0.0000118 |
| 4 | Abelungu | 0.0077 | Air | 0.0000113 |
| 5 | Bbc | 0.0056 | Arguing | 0.0000111 |
| 6 | Bitch | 0.0039 | Behalf | 0.0000111 |
| 7 | Book | 0.0029 | Bio | 0.0000111 |
| 8 | Ace | 0.0023 | Active | 0.0000105 |
| 9 | Bbcnews | 0.0022 | Adhabb | 0.0000105 |
| 10 | Anger | 0.0021 | Allows | 0.0000105 |
| 11 | Amid | 0.0020 | Ammo | 0.0000105 |
| 12 | Black | 0.0019 | Anniversary | 0.0000105 |
| 13 | Artist | 0.0019 | Arya | 0.0000105 |
| 14 | Announces | 0.0017 | Assurance | 0.0000105 |
| 15 | Apatheid | 0.0015 | Ave | 0.0000105 |
| 16 | Author | 0.0014 | Average | 0.0000105 |
| 17 | Africabiz | 0.0014 | Ay | 0.0000105 |
| 18 | Absolute | 0.0013 | Beach | 0.0000105 |
| 19 | Best | 0.0011 | Beating | 0.0000105 |
| 20 | Auto | 0.0010 | Bird | 0.0000105 |

https://doi.org/10.18489/sacj.v32i2.847

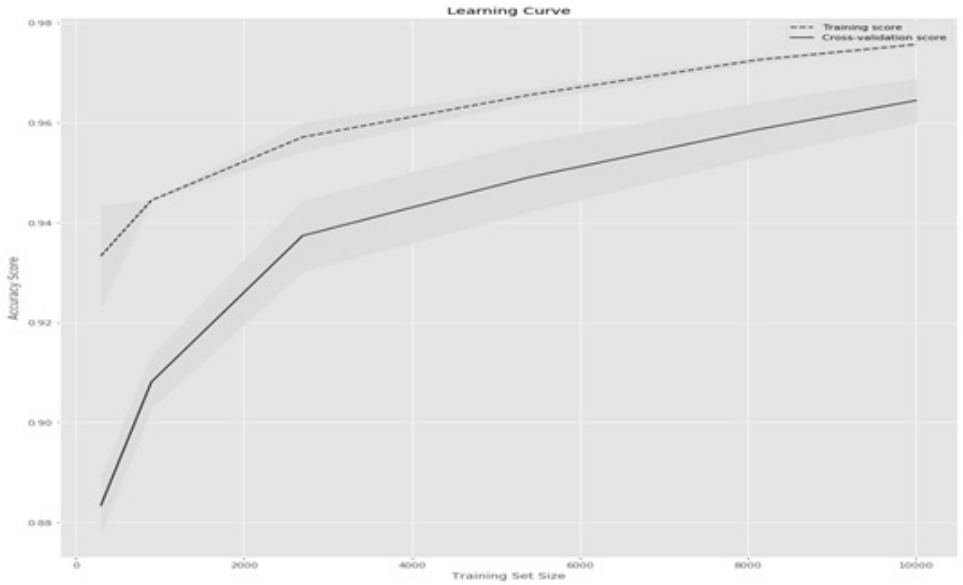


Figure 2: Learning curve for the proposed method

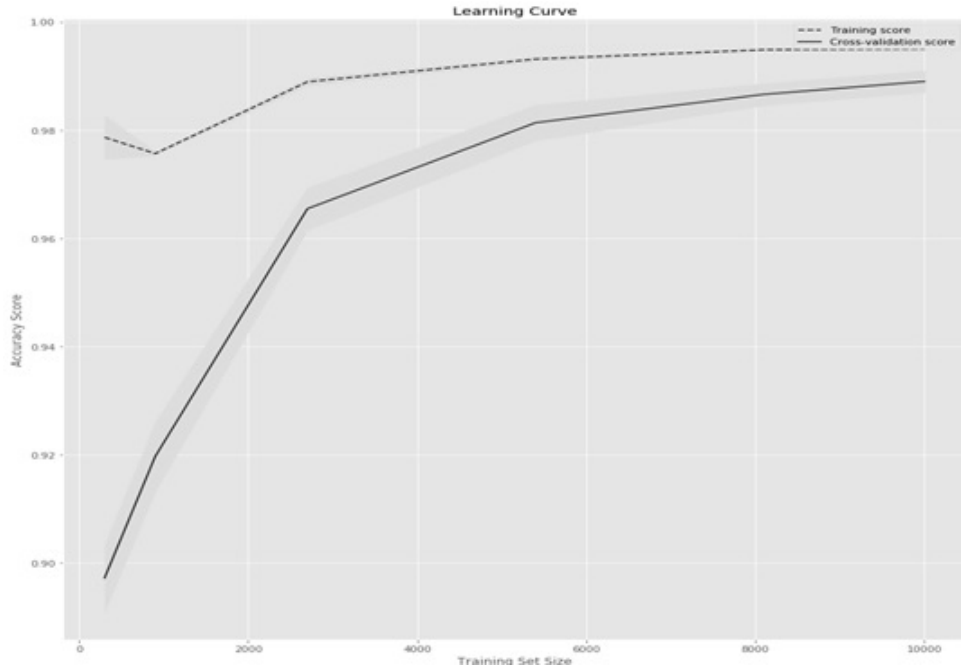


Figure 3: Learning curve for method A

Since training samples for abusive language detection in the South African context are expensive, the learning curve for each method was plotted to observe the progression of clas-

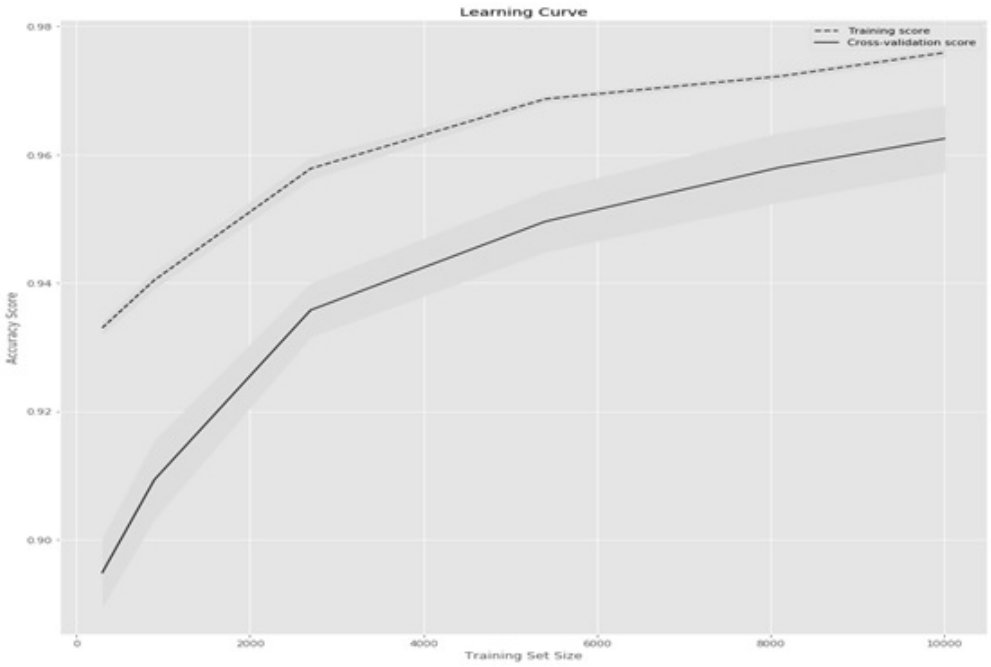


Figure 4: Learning curve for method B

sification, with cross-validation based on different training sizes. Six training sizes (300, 900, 2700, 5400, 8100, 10000) were evaluated. The learning curves for the char n-gram features and LogReg are presented in Figure 2–Figure 5.

The Learning curves in Figure 2 showed that using the proposed method, there was a continuous and consistent increase in the cross-validation accuracy as the training size increases from 300 to 10000. Figure 3 showed that there was little increase and inconsistency in the cross-validation accuracy for method A; Figure 4 showed there was continuous increase and partial inconsistency in the cross-validation accuracy for method B; while the lowest increase with inconsistency was recorded by Method C in Figure 5. The observations show that the proposed method used the training data more efficiently than the other methods. In fact, at the training size of 1000, the predictive accuracy was approximately 0.95.

The outcomes indicated that the proposed method, which aimed to improve the generalization of training data by reducing skewness towards labelled data, improved the detection of South African abusive language on Twitter.

Figure 6 and Figure 7 present the confusion matrices for the proposed method. Figure 6 showed that word and character n-gram features performed better with a much higher TP and TN than FN and FP compared to PoS features. Figure 7 showed that NN performed better with a much higher TN compared to CNN.

To evaluate the significance of the performance of the semi-supervised learning techniques, Cochran's Q test (Raschka, 2018) at significance ($\alpha$) equal to 0.05 was applied on the predic-

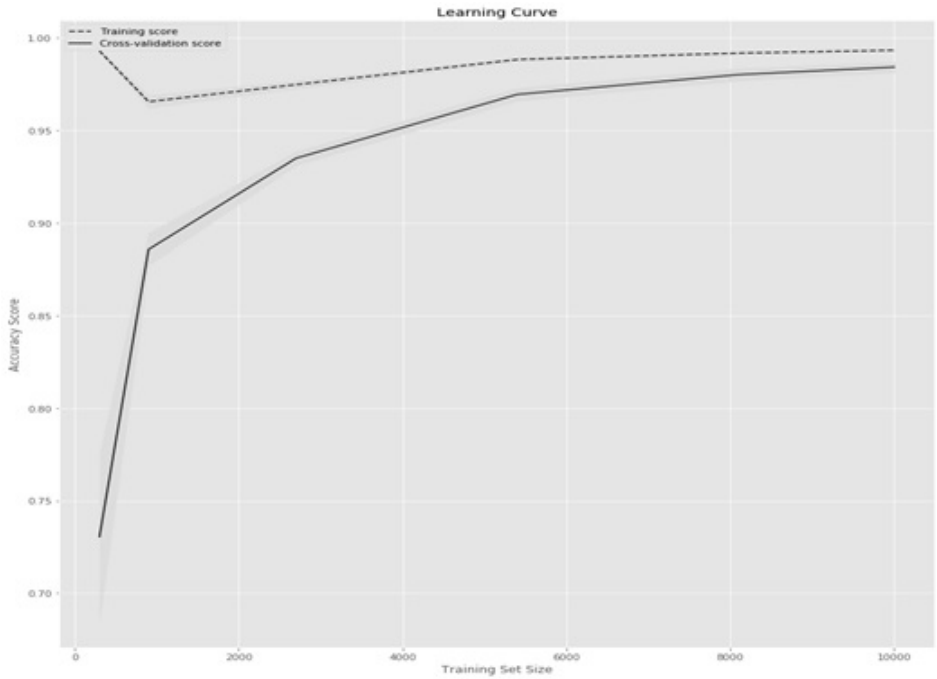


Figure 5: Learning curve for method C

tions of the proposed method, method A, method B and method C, with character n-gram and LogReg. A $\rho$-value of $1.67 \times 10^{31}$ was obtained, which showed the methods performed differently. Furthermore, McNemar's test (van Rossum et al., 2018) was used to perform multiple post hoc pair-wise tests to compare the proposed technique with existing techniques and determine which pairs have different population proportions. The proposed method in comparison with method A recorded a $\rho$-value of $9.74 \times 10^{-28}$; the proposed method in comparison with method B recorded a $\rho$-value of 0.25; and the proposed method in comparison with method C recorded a $\rho$-value of $1.67 \times 10^{-31}$. The results showed that the differences in performances of the proposed method and other methods except method B were significant. Since the $\rho$-value for the proposed method and method B was not less than 0.05, the difference was not significant. Nevertheless, there was a difference of 0.03 in their F1-score.

The analysis of the confusion matrices for the explicit abusive tweets totalling 229 ($97.86$%) and implicit abusive tweets totalling 5 ($2.14$%) showed that $89.95$% of the tweets in the testing data with explicit abusive meaning were correctly classified as abusive tweets while $60.00$% of tweets with implicit abusive meaning were correctly classified as abusive tweets. The incorrectly classified tweets have an average of $85.50$% rare words, with information gain less than 0.00005. The twenty most significant and least significant bag-of-word features with their corresponding information gain are presented in Table 6. Also, it was observed that the seven tweets that were disagreed upon during annotation were labelled as non-abusive tweets by the proposed method.

https://doi.org/10.18489/sacj.v32i2.847

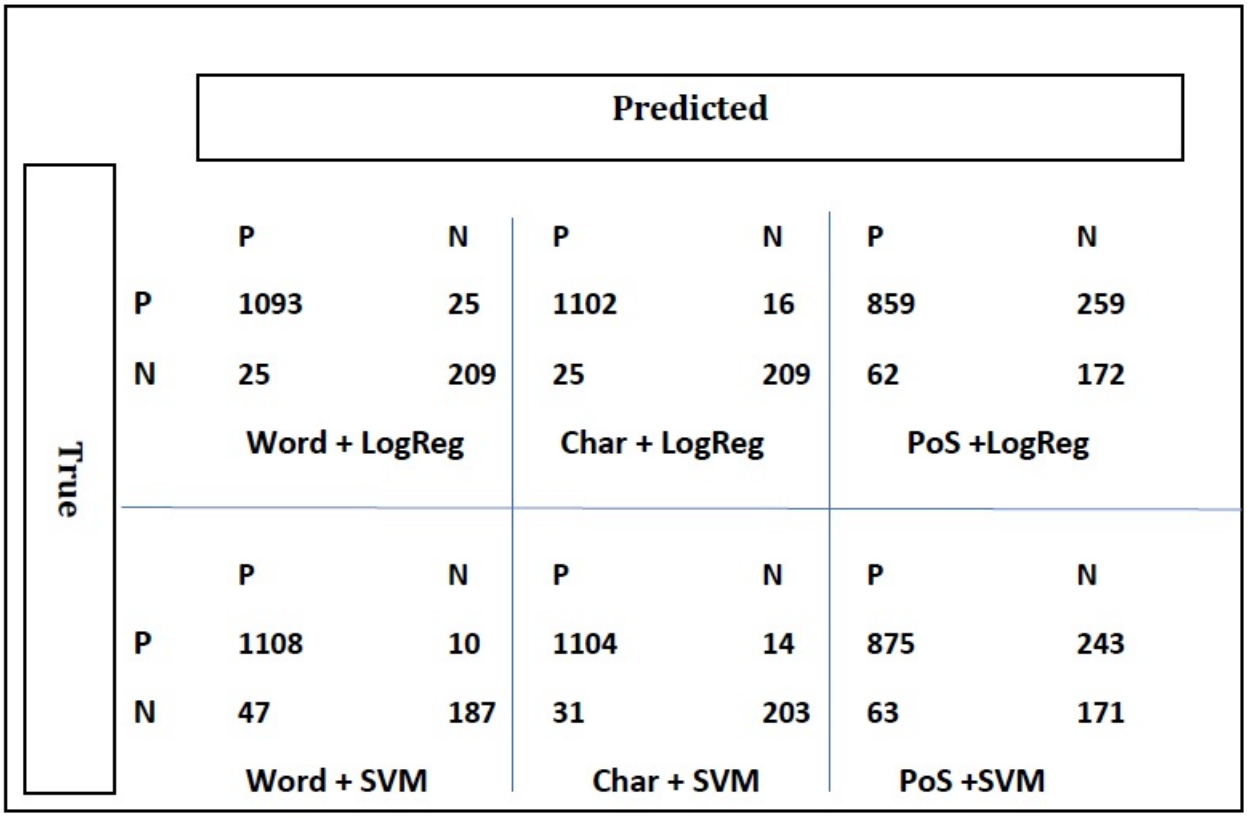


Figure 6: Confusion matrices for sparse features

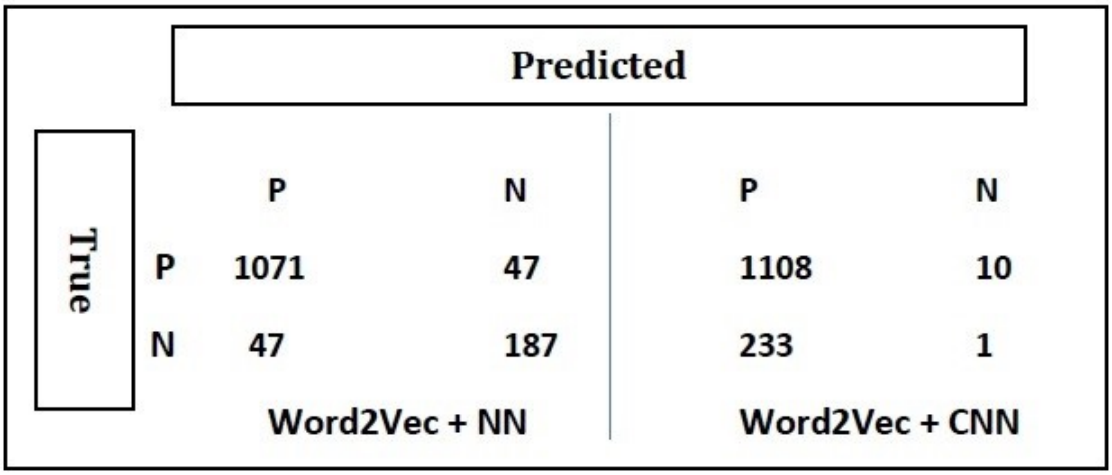


Figure 7: Confusion matrices for dense features

https://doi.org/10.18489/sacj.v32i2.847

Information Gain (IG) (Quinlan, 1986) was applied to measure the impact that different BoW features have on the detection of abusive tweets. A high IG score indicates that the feature has a greater impact on detection. Based on the analysis of 957 features in the training data, Table 6 presents 20 most significant and least significant features.The most significant words are Africa, ANC and African, in descending order. However, few of the most significant features such as abelungu, bitch, black, anger and apartheid, which are offensive and negative words, have less than 0.0008 IG scores. The least significant features included away, ball and bought, in ascending order. These showed that, despite the high number of explicit abusive tweets in the test data, offensive words occurred less frequently compared to normal words in the training data. Therefore, more training instances with offensive words might be required for improved performance.

## 6 CONCLUSION

We developed a semi-supervised learning approach that combined both labelled and unlabelled data, without skewness towards labelled data for improved detection of abusive tweets in a binary classification model.

Matrix multiplication was used to fuse the labelled and unlabelled features, the K-means algorithm was used to cluster the fused features and the majority voting rule was applied to select reliable labels for the unlabelled samples. The labelled and the previously unlabelled samples were used as training data. The performance of the approach was evaluated using syntactic and semantic features that were modelled by logistic regression, support vector machine and neural network classifiers. The char n-gram feature with logistic regression and support vector machine recorded the best performance with accuracy of 0.97 each and F1-scores of 0.95 and 0.94, respectively. With semantic features, however, a different scenario occurred. The results show that the proposed semi-supervised learning technique with syntactic n-gram features performed better than the existing semi-supervised learning approaches.

The poor performance of the models with Word2Vec semantic features suggests that South African tweets are distinct and larger instances are required to improve the embeddings. The poor level of significance based on low information gain for the bag-of-words surface linguistic features shows that, in future, the focus must be on reducing rare words.

Further research will be conducted to improve the linguistics feature space to improve the performance of classification based on word embedding. Contextual word embedding models such as BERT (Devlin et al., 2018) and ELMO (Peters et al., 2018) will be evaluated in the future. Part-of-speech syntactic linguistic features extracted from South African languages will also be included in future studies.

https://doi.org/10.18489/sacj.v32i2.847

## References

Albalate, A., Suchindranath, A. & Minker, W. (2010). A semi-supervised cluster-and-label algorithm for utterance classification. *Intelligent Environments (Workshops)*, 61–70.

Bird, S., Klein, E. & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

Chegini, M., Bernard, J., Berger, P., Sourin, A., Andrews, K. & Schreck, T. (2019). Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Visual Informatics, 3*(1), 9–17. https://doi.org/10.1016/j.visinf.2019.03.002

Chollet, F. (2018). *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG.

Da, K. (2014). A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Danka, T. & Horvath, P. (2018). modAL: A modular active learning framework for Python. *arXiv preprint arXiv:1805.00979*.

Davidson, T., Warmsley, D., Macy, M. & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fleiss, J. L., Cohen, J. & Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin, 72*(5), 323. https://doi.org/10.1037/h0028106

Forestier, G. & Wemmert, C. (2016). Semi-supervised learning using multiple clusterings with limited labeled data. *Information Sciences, 361*, 48–65. https://doi.org/10.1016/j.ins.2016.04.040

Fortuna, P. & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR), 51*(4), 1–30. https://doi.org/10.1145/3232676

Gaydhani, A., Doma, V., Kendre, S. & Bhagwat, L. (2018). Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv preprint arXiv:1809.08651*.

Greevy, E. & Smeaton, A. F. (2004). Classifying racist texts using a support vector machine. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 468–469. https://doi.org/10.1145/1008992.1009074

Gunasekara, I. & Nejadgholi, I. (2018). A review of standard text classification practices for multi-label toxicity identification of online content. *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 21–25.

Ibrohim, M. O. & Budi, I. (2018). A dataset and preliminaries study for abusive language detection in Indonesian social media. *Procedia Computer Science, 135*, 222–229.

Iscen, A., Tolias, G., Avrithis, Y. & Chum, O. (2019). Label propagation for deep semi-supervised learning. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 5070–5079. https://doi.org/10.1109/CVPR.2019.00521

Kamper, H., Livescu, K. & Goldwater, S. (2017). An embedded segmental k-means model for unsupervised segmentation and clustering of speech. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 719–726.

Kargutkar, S. M. & Chitre, V. (2020). A study of cyberbullying detection using machine learning techniques. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 734–739. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000137

Khatri, C., Hedayatnia, B., Goel, R., Venkatesh, A., Gabriel, R. & Mandal, A. (2018). Detecting offensive content in open-domain conversations using two stage semi-supervision. *arXiv preprint arXiv:1811.12900*.

Kotzé, E., Senekal, B. A. & Daelemans, W. (2020). Automatic classification of social media reports on violent incidents in South Africa using machine learning. *South African Journal of Science*, *116*(3-4), 1–8. https://doi.org/10.17159/sajs.2020/6557

Kumar, S., Gao, X. & Welch, I. (2017). Cluster-than-label: Semi-supervised approach for domain adaptation. *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 704–711. https://doi.org/10.1109/AINA.2017.166

Lee, Y. J. & Grauman, K. (2009). Foreground focus: Unsupervised learning from partially matching images. *International Journal of Computer Vision*, *85*(2), 143–166. https://doi.org/10.1007/s11263-009-0252-y

Leng, M., Wang, J., Cheng, J., Zhou, H., Chen, X. et al. (2014). Adaptive semi-supervised clustering algorithm with label propagation.

Livieris, I. E., Kanavos, A., Tampakas, V. & Pintelas, P. (2018). An auto-adjustable semi-supervised self-training algorithm. *Algorithms*, *11*(9), 139. https://doi.org/10.3390/a11090139

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Park, J. H., Shin, J. & Fung, P. (2018). Reducing gender bias in abusive language detection. *arXiv preprint arXiv:1808.07231*. https://doi.org/10.18653/v1/D18-1302

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Peikari, M., Salama, S., Nofech-Mozes, S. & Martel, A. L. (2018). A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, *8*(1), 1–13. https://doi.org/10.1038/s41598-018-24876-0

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1023/A:1022643204877

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of open source software*, *3*(24), 638. https://doi.org/10.21105/joss.00638

Rehurek, R. & Sojka, P. (2011). Gensim–Python framework for vector space modelling [NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic].

Schmidt, A. & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International workshop on natural language processing for social media*, 1–10. https://doi.org/10.18653/v1/W17-1101

Schmidtler, M. A., Caruso, N. & Borrey, R. (2008). Data classification methods using machine learning techniques [US Patent App. 11/752,691].

Tran, P. V. (2019). Semi-supervised learning with self-supervised networks. *arXiv preprint arXiv:1906.10343*.

Tuckwood, C. (2017). Hatebase: Online database of hate speech. https://hatebase.org

van Aken, B., Risch, J., Krestel, R. & Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*.

van Rossum, G. et al. (2018). Python tutorial: Release 3.6.4.

Yuan, S., Wu, X. & Xiang, Y. (2016). A two phase deep learning model for identifying discrimination from tweets. *EDBT*, 696–697.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N. & Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Zimmerman, S., Kruschwitz, U. & Fox, C. (2018). Improving hate speech detection with deep learning ensembles. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.