IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 36, No. 2, April, 2022, pp. 175-183
Journal homepage: http://iieta.org/journals/ria

Design and Implementation of a Multichannel Convolutional Neural Network for Hate Speech Detection in Social Networks

Zeleke Abebaw[1*], Andreas Rauber[2], Solomon Atnafu[3]

[1] IT Doctoral Program, Addis Ababa University, Addis Ababa 1176, Ethiopia
[2] Institute of Information Systems Engineering, Technical University of Vienna, Favoritenstraße 9-11/194-01, Vienna A-1040, Austria
[3] Department of Computer Science, Addis Ababa University, Addis Ababa 1176, Ethiopia

Corresponding Author Email: zeleke.abebaw@aastu.edu.et

https://doi.org/10.18280/ria.360201

Received: 27 October 2021
Accepted: 16 March 2022

Keywords:
Amharic hate speech detection, multichannel convolutional neural network, deep learning, social media comment, single channel, word embedding

ABSTRACT

As the number of social media comments available online grows, the spread of hate speech has grown gradually. When someone uses hate speech as a weapon to injure, degrade, and humiliate others, their freedom, dignity, and personhood can be jeopardized. Deep neural network-based hate speech detection models, such as the conventional single channel convolutional neural network (SC-CNN), have recently demonstrated promising results. The success of the models, however, is dependent on the type of language they are trained on and the training data size. Even with a small amount of training data, the model's performance can be improved by using a multichannel convolutional neural network (MC-CNN) model. The study assesses and compares the performance of a multichannel convolutional neural network model to single channel convolutional neural network models using a support vector machine (SVM) as a baseline. The models' F1 score values are computed, and promising results are obtained. The MC-CNN model outperforms the SC-CNN models in all three hate speech datasets. The study's findings indicate that the proposed MC-CNN model could be used as a deep learning-based alternative for hate speech detection.

1. INTRODUCTION

Social media platforms such as Facebook, YouTube, etc. are dominating social and political affairs in nearly every part of the world. This has given the way for new and old racist acts such as the spread of hate speech to stir on these platforms rapidly [1]. Hate speech is an expression that denigrates a person or groups of people based on alleged membership in a social group identified by attributes such as race, ethnicity, gender, sexual orientation, religion, age, physical or mental disability, etc. [2].

Social media users spread hate speech using social networks that urge people to send hateful messages, write harsh critiques, perpetrate violent acts, and commit hate crimes [3], which is becoming a worldwide phenomenon. For example, in the United States, the yearly figures of the 2019 FBI report on hate crime revealed that hate crimes had risen by 3%. The main forms being racial, religious, and sexual offenses. In Kenya during the 2007 election, 1,300 people were left dead and more than 650,000 displaced due to the development of hate speech in interethnic conflicts and police violence [4]. Recently, in Ethiopia, hate and aggressive comments have also been widespread in social media, especially during election periods and political instabilities [5]. These phenomena across the country caused many disruptions that affect the lives of millions, hindering businesses, displacement of communities, and even causing hundreds of deaths [6].

In general, the study [7] confirmed that hate speech and hate crime poison society. It endangers individual rights, human dignity, and equality, provoking social group conflicts, disrupting public peace and orders, and putting peaceful coexistence at risk. As a result, many organizations take various steps to reduce the spread of hate speech. While governments utilize law enforcement, social media corporations use machine-learning algorithms to create automatic hate speech detection models.

Deep neural network models such as conventional single channel convolutional neural networks have accomplished remarkable performance in hate speech detection [8-11]. However, the single channel CNN fails to consider features that can improve hate speech detection process. We hypothesize that employing multiple channels of the CNN model could capture additional hidden features from each channel that would otherwise be dropped. Initializing separate channels from the input embedding space and concatenating the max-pooling layer from each channel enables to generate better features from the multiple channels of the CNN model. To realize this, our proposed method has the following objectives:

- to develop a hate speech detection system that does not rely on time-consuming and expensive handcrafted features,
- to find the optimal n-grams for the proposed model for each of the three sampled hate speech datasets used, and
- to evaluate the performance of a multichannel convolutional neural network compared to single channel convolutional neural network models in hate speech detection.

Hypothesizing the multichannel convolutional neural network architecture might learn better features than the single channel convolutional neural network model for small datasets, we design and implement a two-channel convolutional neural network model on top of the word2vec word-embedding layer for hate speech detection. The model based itself on the standard CNN model [12]. It consists of an input layer with a word embedding layer of 100-dimensions, a multichannel convolutional layer, a max-pooling layer, a flatten layer, and the output layer. We are interested in the benefits this approach might bring to the analysis of lesser-resourced languages. Thus, we particularly compare the performance of the MC-CNN not only on English social media posts but also focus specifically on Amharic, the national language of Ethiopia. The following are the major contributions of this work:

- We design and implement a hate speech detection system that learns features automatically using a word2vec model as input.
- We discover that the proposed multichannel CNN model performs better in two channels with 4-grams and 5-grams for Amharic, and 6-grams and 7-grams for English hate speech datasets.
- We evaluate the performance of the multichannel CNN model compared to the single channel CNN models on Amharic and English hate speech datasets and find that the proposed multichannel CNN model shows better performance than the single channel CNN models.

The remaining part of the paper is structured as follows. Section 2 presents the related work. In Section 3, the research methodology is provided. The proposed model is presented in Section 4. Section 5 contains experiments, results, and discussion parts. Finally, Section 6 concludes the paper and point out future works.

2. RELATED WORK

2.1 Hate speech

Although the term hate speech has no common definition, there are different attempts to understand the elements it contains. The work [13] defined hate speech as "*any type of communication that causes the damages that proponents for suppression attribute to hate speech such as loss of self-esteem, economic and social subordination, physical and emotional stress, victim silence, and effective exclusion from the political arena*". Moran [14] examined hate speech as "*speech meant to incite hate against traditionally disadvantaged populations*". Ward [15] defined hate speech as "*any type of discourse in which speakers principally seek to condemn, humiliate, or inspire hatred against their targets*". Something is hate only if it is a speech or expressive conduct that concerns any members of a group or classes of persons identified by protected characteristics such as race, color, religion, etc. and involves or is intimately connected with emotions, feelings, or attitudes of hatred [16].

The social media corporations such as Facebook, YouTube, and Twitter also have their definitions of hate speech. Facebook considers it as "*any content that explicitly insults individuals based on their race, ethnicity, national origin, religious affiliation, sexual orientation, sex, gender, or gender identity, or severe impairments or illnesses*" [17]. On the other side, YouTube considers it "*any content that promotes violence against individuals or groups based on race or ethnic origin, religion, disability, gender, age, nationality, veteran status, or sexual orientation/gender identity or content whose primary purpose is inciting hatred based on these core characteristics*" [18]. Finally, Twitter considers hate speech as "*any content used by someone for inciting violence against or actively assaulting or threatening other individuals because of their race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, handicap, or disease*" [19].

In all of the above hate speech definitions, one common aspect that can be observed is that hate speech attacks people's identity. Taking into account this aspect, we develop a working definition of hate speech for this research. We define hate speech as '*speech in the form of text that fuels discrimination against individuals or groups based on their nationality, ethnic and religious affiliation, sex or disabilities'*.

2.2 Hate speech detection approaches

Research works in automatic hate speech detection focuses on feature engineering techniques and classification algorithms. The success of classification algorithms highly depends on feature engineering techniques used. Researchers use two types of feature engineering: handcrafted features in machine learning and automatic feature learning in deep learning.

Finding the right features to tackle a problem can be one of the most demanding tasks in machine learning, particularly, in hate speech detection. We shortlist the handcrafted features mostly used in text mining to the specific problem of automatic hate speech detection. Among these dictionary-based [20], bag-of-words [21], n-grams [22], TFIDF [22], part-of-speech [23], lexical syntactic features [24], rule-based [21], word sense disambiguation [25], and topic modelling [3] are the most common ones. These methods are labor-intensive and error-prone. Hence, alternatively automatic feature learning methods are proposed.

For automatic feature learning, the enormous amount of data available on social media opens up great opportunities for new knowledge discoveries by analyzing patterns of relations that coexist in the data. Learning algorithms can find the optimal parameters to create the best performing model. As a result, hate speech detection using automatic feature learning in deep learning models have shown remarkable performance [11], particularly, CNN models because of training speed and low computational cost while maintaining better results [26].

2.3 Multichannel Convolutional Neural Network

Originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results [27]. CNN is designed to automatically and adaptively learn features. CNN has three fundamental building layers: convolution, pooling, and fully connected layers. While the first two, convolution and pooling perform feature extraction, the third a fully connected layer, maps the extracted features into a final output such as classification [28].

The model shown in Figure 1 is a slightly different variant of the CNN architecture [12] for a two channel CNN. Let $xi \epsilon \mathbb{R}^k$ be the k-dimensional word vector corresponding to the i^{th} word in a sentence. A sentence of length n padded where necessary is presented as:

$$x1=x1\oplus x2\oplus\ldots\oplus xn. \quad (1)$$

where, $\oplus$ is the concatenation operator, and the convolutional operational consists of a filter $w\in\mathbb{R}^{hk}$, which is applied to a window of h words to produce a new feature. For instance, feature ci is generated from a window of words xi: $i+h-1$ by:

$$ci=f(w.xi{:}i+h-1+b) \quad (2)$$

where, $b\in\mathbb{R}$ a bias is a term and f is a nonlinear function (e.g., rectifier or tanh). This is done for every step of the input sequence $\{xi{:}h, x2{:}h+1, \ldots, xn-h+1{:}n\}$ to produce a feature map of:

$$ci=[c1, c2, \ldots, cn-h+1] \quad (3)$$

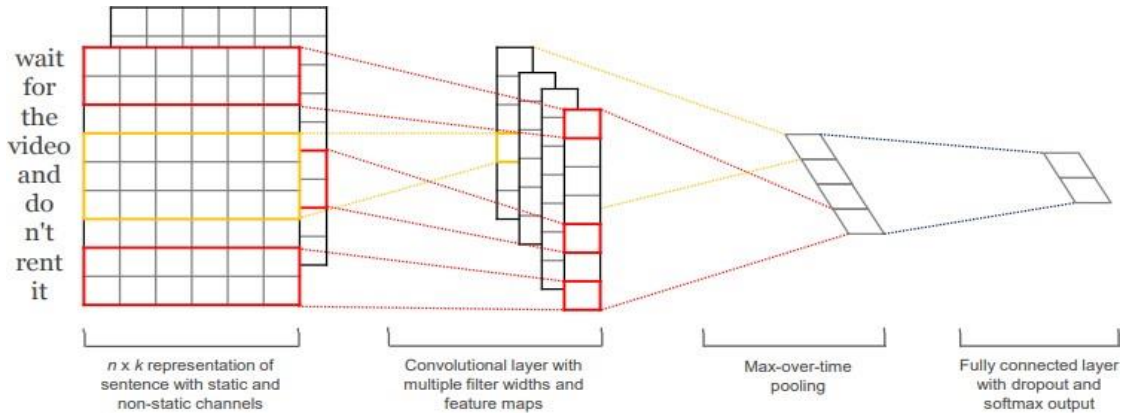

Figure 1. MC-CNN architecture for natural language processing [12]

2.4 Multichannel Convolutional Neural Network for text mining

Multichannel Convolutional Neural Network (MC-CNN) models have been used to address different NLP problems including hate speech detection. Yoon [12] evaluated shallow learning techniques, including multichannel CNN, and found that multichannel CNN outperformed other approaches on the Stanford sentiment Treebank and consumer review datasets [26]. Further, Brownlee [29] showed how to implement a basic multichannel CNN model for NLP applications. He noted that *"the model can be expanded by using convolutional neural networks that read the source document using different kernel sizes. This, in effect, creates a multichannel convolutional neural network for the text that reads text with different n-gram sizes"*. Following that, several researchers used multichannel CNN for various NLP applications.

Van Dinter et al. [26] presented a MC-CNN method to automate the citation selection process. According to the author, the suggested technique outperforms existing deep learning algorithms such as LSTM in terms of performance accuracy.

Dahou et al. [30] suggested a multichannel embedding convolutional neural network to enhance Arabic sentiment classification. By learning sentiment variables from multiple text domains, word, and character n-gram levels, the suggested model shows high classification accuracy.

Wang and Pedersen [31] developed a MC-CNN model based on sub-word embedding for the representation of tweets to predict emoji categorization results. By eliminating ambiguity, the suggested method aids in boosting classification accuracy and ground truth information.

The multichannel convolutional neural network approaches are also used for hate speech detection. Alotaibi et al. [32] developed a multichannel deep learning model that combines three networks, the bidirectional gated recurrent unit, transformer block, and convolutional neural network, to classify 55,788 Twitter comments into aggressive and non-aggressive classes split into 75% training and 25% for testing. The suggested technique outperformed the previous methods with an accuracy of 87.99%.

Shah et al. [33] developed a multichannel convolutional neural network with a bi-directional gated recurrent unit for hate speech detection evaluating it on a total of 34,178 hate tweets and 1,521,857 non-hate tweets of six separate Political and COVID-19 datasets. By reducing ambiguity, the suggested method helps in boosting classification accuracy over 99.0% and ground truth information.

The works reviewed above present interesting solutions. However, to the best of our knowledge, the integration of multiple channels of convolutional neural network model to generate better (hidden) features from each channel for low-resourced language is under-investigated. Thus, the main purpose of this work is to determine how shared features of the multichannel convolutional neural network model on top of the word2vec word-embedding layer efficiently perform hate speech detection compared to features generated from a single channel convolutional neural network model on a limited dataset as in the case of the under-resourced language, Amharic. For the model generalization purposes, we also test the model on English hate speech datasets.

3. RESEARCH METHODOLOGY

3.1 Datasets

To train the proposed model, we use three different datasets. Amharic dataset publicly available at Zenodo [34] abbreviated as "Amh" and English hate speech datasets accessed from the publicly available datasets of Davidson et al. [35] abbreviated as "Dvd" and the White Supremacy of de Gibert et al. [36] abbreviated as "Whs". Each of the three datasets contains 2,000 social media users' comment classified as hate or not-hate. In all cases, we split the data into train and test set instances: 1,600 posts (80%) of the data are used to train the classification models and 400 posts (20%) are used to evaluate the performance of the models.

3.2 Data preprocessing

We remove Amharic punctuation marks, URLs, unnecessary white spaces, and non-Amharic characters. Since there are different ways of writing the same Amharic word using different characters, we also perform Amharic character normalization using a normalization tool available in Ref. [37] for dimension reduction. For instance, the Amharic word "ዓለም" (world) can have multiple writing styles such as "አለም", "ዐለም" or "አለም". However, we do not use stop word removal for dimension reduction in this work. Because we find that it carries significant meaning in hate speech detection. For example, the statement "መንግስቱ ገዳይ ነው" ("Mengistu is killer"). The stop word "ነው" (is) plays a significant role in labeling the statement as hate speech. We can capture this concept using word n-gram models.

We also remove hashtags, emoticons, user mentions, and URLs in social media posts and comments of the English dataset while also not removing stop words in the English hate speech datasets.

3.3 Feature engineering method

Machine learning algorithms cannot learn classification rules unless the raw texts are converted to numerical features. Hence, feature extraction is one of the fundamental steps in text classification. This step is used to extract the key features from the raw text to represent it in numerical forms. In this work, we apply two feature-engineering techniques: n-grams and word2vec.

3.3.1 N-gram based feature selection

Based on previous research works for text classification, for our hate speech detection experiment, we use word n-grams as features and pass their TFIDF (term frequency-inverse document frequency) values to the SVM machine-learning model used as a baseline classifier. We perform comparative analysis considering different values of n in the model. In this experiment, we test unigram (n=1), bigram (n=2), and the combination of the two for the SVM classifier (see *Appendix A*).

3.3.2 Word2vec feature learning

Given the high volume of available textual data, classification models in most resourceful languages (e.g., English) benefit from automatic feature learning methods. For instance, pre-trained publicly available models using word2vec [38] and FastText [39] are key components of neural language models for text classification. Though the same efforts are made to prepare such types of models for under-resourced languages such as Amharic, using FastText [40] the model is not representative enough for hate speech detection problems because hate speech contents posted by users have their unique characteristics that are not observed in the standard texts. For instance, the acronym "10Q" is a short form of the phrase "Thank you" commonly used by Facebook users. Similarly, in Amharic social media, users write uncompleted words to express hate expressions such as አህ* for the insult expression አህያ (donkey). Therefore, to incorporate the meaning of such words in the feature space and avoid out-of-vocabulary problems, we use the continuous bag of words (CBOW) word-embedding model [41] to generate features for our hate speech detection problem.

4. THE MULTICHANNEL CNN MODEL

We propose a multichannel CNN model for hate speech detection, shown in Figure 2, hoping that the multichannel architecture would learn better features than the single channel model, especially for smaller datasets. The model involves using multiple versions of the standard model [12] with different kernel sizes on the dataset. This allows the dataset to be processed at different widths of n-grams at a time, whilst the model learns how to best integrate these interpretations. We define a multiple input model with two input channels for processing different n-grams on each channel. Each channel is comprised of the following elements (detailed configuration is given in Section 5.3):

- Input layer defines the length of input sequences
- One-dimensional convolutional layer
- Max Pooling layer to consolidate the output from the convolutional layer.
- Flatten layer to reduce the three-dimensional output to two dimensional for concatenation.
- The output from the two channels are concatenated into a single vector and processed by a dense layer and an output layer.

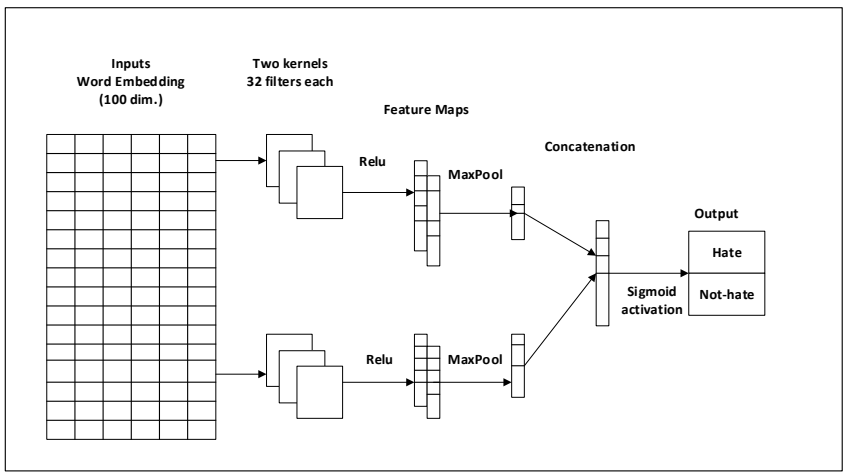

Figure 2. Architecture of the proposed multichannel CNN model for hate speech detection

5. EXPERIMENTS

We run a range of experiments to assess model performance (SVM, SC-CNN-1, SC-CNN-2, and MC-CNN) in detecting hate speech in Amharic and English datasets. We perform a binary classification in which comments are classified as hate or not-hate.

5.1 Implementation procedures

We conduct the experiments on Dell Vostro 35 with 8 GB of RAM, 500GB of HD, with processor speed of 2.8GHz. A Jupyter Notebook using anaconda to setup a python 3.5.2 environment is used. For deep learning, TensorFlow 2.3.0 and Keras 2.4.3, and for the machine learning tasks, the main SciPy libraries such as NumPy, Pandas, Matplotlib, Scikit-learn, SciPy, and Stats models are used.

5.2 Defining and training the models

5.2.1 Single channel CNN-1 (SC-CNN-1)

The first single channel CNN (SC-CNN-1) model is defined as embedding size of 100, for the convolution a filter size of 32, kernel size of 4 for the Amharic, and kernel size of 6 for the English dataset and the activation is ReLu with a dropout rate of 0.5. The max-pooling size is 2. The output layer is Dense 2 with a sigmoid activation function, corresponding to two classes. Each of the specific configurations of the model is displayed in Table 1. To train the model, we follow a validation split of 0.1 with epochs of 16 and the batch size is 20.

5.2.2 Single channel CNN-2 (SC-CNN-2)

We build the second single channel CNN (SC-CNN-2) model with an embedding layer size of 100 and one Conv layer. The Conv layer has 32 filters and the size of the kernel is now 5 to incorporate more n-gram words (Table 1). The Conv layer activation is ReLu. The output layer is Dense 2 with a sigmoid activation function, corresponding to two classes. To train the model a validation split of 0.1 with epochs of 16 and the batch size 20 is used.

5.2.3 Multichannel CNN (MC-CNN)

Table 1. Hyper parameter configurations of the proposed model

Models	Filters	Kernel (n-grams)		Activation
		Amharic	English	
SC-CNN-1	32	4	6	ReLu
SC-CNN-2	32	5	7	ReLu
MC-CNN	32	4 & 5	6 & 7	ReLu

To see the unified feature's behavior of the multichannel CNN model parameters compared to the single channel CNN models in all of the three hate speech datasets, we conduct experiments by defining the MC-CNN model concatenating the above two single channel CNN models. We build the MC-CNN model with an embedding layer with an embedding size of 100 and two Conv layers. Each Conv layer has 32 filters and the sizes of the kernel are 4-grams and 5-grams for Amharic, and 6-grams and 7-grams for English concatenated words (Table 1). The Conv layer activation is ReLu. The output layer is Dense 2 with a sigmoid activation function, corresponding to two classes. To train the MC-CNN model, we use the same validation split of 0.1, epochs of 16, and batch sizes of 20. The experimental evaluations are reported using all three hate speech datasets.

5.2.4 The baseline

As a baseline, we use linear Support Vector Machine (SVM) classifier because it has shown effective performance in previous studies using TFIDF-weighted bag-of-word features [22]. Further, we use the grid search optimization strategy to select the best parameters for each datasets. Specific configuration of parameters is shown in ***Appendix A***. We use the python scikit-learn library to implement the classification model.

5.3 Evaluation

The performances of the proposed model classifiers using the test dataset are evaluated recording the statistics of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These are defined as follows:

- True Positives (TP) are the number of correctly predicted hate comments.
- True Negatives (TN) are the number of correctly predicted not-hate comments.
- False Positives (FP) are the number of incorrectly predicted hate comments.
- False Negatives (FN) are the number of incorrectly predicted not-hate comments.

Moreover, three performance metrics are used to evaluate the classifiers. These are recall, precision, and F-measures.

Recall (R): the proportion of actual positives, which are predicted positive.

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

Precision (P): the proportion of predicted positives which are actually positive.

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

F-measure (F1): the harmonic mean of precision and recall.

$$F - measure = 2.\frac{Recall.Precision}{Recall+Precision} \tag{6}$$

5.4 Experimental results and discussion

5.4.1 Results of the baseline (SVM)

To figure out which n-gram best matches the classifier, we start with the baseline (SVM) and experiment with 1-gram, 2-gram, and a mixture of the two with TFIDF vectorizer (see parameter configurations in ***Appendix A***). Table 2 shows the performance of the SVM classifier in precision, recall, and F-score.

The performance of the SVM classifier is greater when applying 1-gram in all three datasets (Table 2). This is demonstrated by Davidson's F1 score of 67.8, White Supremacy's F1 score of 66.8, and the Amharic dataset's F1 score of 85.8. In all three datasets, the SVM classifier performs poorly when using 2-grams. When the 1-gram and 2-gram feature sets are combined, the classifier's performance improves but not as in the unigrams. As a result, in the multichannel CNN model comparisons, we present the performance of the SVM classifier using 1-gram features.

Table 2. Performance of linear SVM in different n-grams using TFIDF features

Datasets	N-grams	P	R	F1
Dvd	1-gram	68.4	68.0	**67.8**
	2-gram	61.2	60.2	59.4
	combined	68.0	67.5	67.3
Whs	1-gram	67.6	67.0	**66.8**
	2-gram	57.6	57.3	56.6
	combined	65.0	64.0	63.3
Amh	1-gram	83.7	88.2	**85.8**
	2-gram	74.6	76.8	74.3
	combined	83.2	87.6	85.3

We also test, the training efficiency of the SVM classifier while using different n-grams. The training and prediction time of the classifier for each n-gram is displayed in ***Appendix A***. Compared to the 1-grams, the model takes relatively longer training times for the combined features and 2-grams. This shows that there is a positive correlation between training time and vocabulary sizes. The vocabulary size of the combined features and 2-grams are relatively larger than the 1-grams. For instance, there are 83,826 terms in the Amh datasets in the 2-gram feature settings without any pruning of the feature space, i.e. taking all terms with a minimum document frequency of 1 and a maximum document frequency of 100% of the collection size. On this vocabulary size, the SVM classifiers perform poorly with precision of 74.6, recall of 76.8, and F1 score of 74.3 with longer period of training time of 3.468758s. Instead applying the pruning method in the TFIDF vectorizer with minimum document frequency of 10 and a maximum document frequency of 40% of the collection, the SVM model produces better results with precision of 83.7, recall of 88.2, and F1 score of 85.8 with relatively smaller training time of 0.120911s with a vocabulary size of 517 terms. Hence, the pruning strategy used helps the model to produce better results.

5.4.2 Determining the optimal n-gram setting

To find the optimal n-grams for each of the three hate speech datasets, we conduct a number of experiments using a single channel CNN model. We evaluate 2, 3, 4, 5, 6, 7 and 8-grams. The performance of the models' F1 score is shown in Table 3.

Table 3. F1 score (%) of different n-grams on single channel CNN model on the three datasets using word2vec features

Datasets	n-grams						
	2	3	4	5	6	7	8
Amh	75.7	75.5	**78.3**	**76.9**	70.3	67.6	66.4
Dvd	63.5	60.0	57.9	64.4	**64.8**	**66.5**	65.3
Whs	63.9	63.6	61.1	65.3	**65.8**	**67.3**	64.6

In Table 3, for the Amharic dataset, the optimal performance of the model is with 4-grams and 5-grams with F1 score of 78.3 and 76.9, respectively. On the other hand, for the English datasets it is with 6-grams and 7-grams with an F1 score of 64.8 and 65.8 for 6-grams, and F1 score of 66.5 and 67.3 for the 7-grams of Davidson and White Supremacy datasets, respectively. In the experiment, as we increase the number of n-grams (6-grams and above for Amh, and 8-grams and above for both Dvd and Whs) the performance of the model decrease.

We also observe different behavior on the number of n-grams and the type of languages trained. When the model is trained on the Amharic dataset, the model performs with relatively smaller numbers of n-grams (4-grams and 5- grams) than on the English datasets. On the other hand, the model performs on a relatively large number of n-grams (6-grams and 7-grams) on the English datasets. The reason could be due to the rich morphology of the Amharic language. For example, an entire phrase consisting of " ደበላለቀባቸው " ('*debelaleqebachew'*) can be translated using nine English words ("*He mixed up the object which belongs to them*"). Due to such word formation, a small number of n-grams can already be rich in information.

5.4.3 Results of single and multichannel CNN, and baseline

Table 4 shows the F1 score of all the four models (SC-CNN-1, SC-CNN-2, MC-CNN, and SVM). For the English language datasets, the MC-CNN slightly outperforms the SVM baseline and also the single-channel CNNs with an F1 score of 68.5 and 68.0 for the Dvd and Whs datasets, respectively. For the Amharic datasets, however, while the MC with an F1 score of 80.2 performs better than two single channel models (F1 of 78.3 and 74.9 for CNN-1 and CNN-2), it does not outperform the baseline SVM model, which achieves an F1 score of 85.5.

Table 4. F1 score of SC-CNN-1, SC-CNN-2, MC-CNN, and SVM on the three datasets

Models	Features	Average F1 score		
		Amh	Dvd	Whs
SVM	TFIDF	**85.5**	67.8	66.8
MC-CNN	word2vec	80.2	**68.5**	**68.0**
SC-CNN-1	word2vec	78.3	64.4	65.8
SC-CNN-2	word2vec	74.9	66.5	67.3

5.4.4 Discussion

In this work, we assess the impact of n-grams, multiple channels, and feature types (TFIDF and word2vec) on model performance. In the first challenge, to determine the number of n-grams that best fit for model performance, we run several experiments on all of the three datasets using single and multichannel CNN models. From the experiments, we find that, while the models perform well in 4-grams and 5-grams for the Amharic dataset, they perform better in 6-grams and 7-grams on both of the two English datasets. This implies that the performance of the models differs according to the number of n-grams across the different languages. Hence, we need to determine the number of n-grams before we implement the models in the actual environment. Furthermore, as indicated in Table 1, increasing the number of n-grams beyond certain points reduces the performance of the models in all of the three datasets.

Secondly, we assess the impact of the TFIDF and word2vec vecortizers on model performance. The performance of the models differs across the datasets. The SVM model performs better using the TFIDF vectorizer on the Amharic dataset with F1 score of 85.5, followed by Davisdson's datasets with F1 score of 67.8, and White Supremacy with F1 score of 66.8. Closely inspecting the datasets, the Amharic social media comments are written in a more structured way than the English texts. Hence, the SVM classifier using TFIDF vectorizer performs better on this dataset. On the other hand, on both the Davidson's and White Supremacy's datasets, the MC-CNN model using word2vec vectorizer performs better than the TFIDF vectorizer with F1 score of 68.5 and 68.0, respectively.

Finally, we assess the impact of multiple channels on model performance. Comparing the variants of the CNN models such as SC-CNN-1, SC-CNN-2, and MC-CNN separately, the MC-CNN performs better than any of the single channel models in all of the three datasets. This is due to the effect of shared features generated from the multiple channels of the CNN model with different hyper-parameter settings. Hence, the proposed MC-CNN model can be considered as an alternative approach for hate speech detection in a deep learning environment where scarcity of labeled training datasets is a problem as in the case of under-resourced languages, Amharic.

6. CONCLUSIONS

This work proposes a multichannel CNN-based deep learning model to classify social media user comments as hate speech or not-hate speech to support the effort in the development of a safe social media ecosystem. The method uses word2vec word embedding as the input layer for the proposed model to learn better features automatically. This approach has multiple advantages. First, the automatic hate speech feature learning implemented minimizes the manual efforts and errors in designing hate speech features. Second, the multichannel approach boosts model performance because of the shared learned features from the basic CNN model. Third, the research work determines the number of sequences of words (n-grams) across different languages. The performance of the proposed model is better with 4-grams and 5-grams for the Amharic dataset and 6-grams and 7-grams for the English dataset without removal of stop words. The experimental findings show promising results. The finding of the study implies that the proposed MC-CNN model can be an alternative solution for hate speech detection using a deep learning approach. This work provides one additional contribution to the research undertaken for under-resourced languages.

While the experimental evaluations are promising for a two-class hate speech detection and a monolingual small dataset scenario, further works can be done in the space of improving the model performance by considering the potential effect of large datasets, multilingual datasets, and multiple hate speech classes such as hate speech in religion, ethnicity, gender, etc. Hence, a code-mixed dataset from both the resource-rich languages and under-resourced ethnic-based local languages can be tested to alleviate the fundamental problem of labeled hate speech dataset scarcity.

ACKNOWLEDGMENT

We are thankful for the partial financial support of Addis Ababa University.

REFERENCES

[1] Matamoros-Fernández, A., Farkas, J. (2021). Racism, hate speech, and social media: A systematic review and critique. Television & New Media, 22(2): 205-224. https://doi.org/10.1177/1527476420982230

[2] William, M.C. (2010). Hate speech. University of Portland 2010. https://www.britannica.com/topic/hate-speech, accessed on February 26, 2021.

[3] Agarwal, S., Sureka, A. (2017). Characterizing linguistic attributes for automatic classification of intent based racist/radicalized posts on tumblr micro-blogging website. arXiv preprint arXiv:1701.04931. https://doi.org/10.48550/arXiv.1701.04931

[4] Rawlence, B. (2013). High stakes: Political violence and the 2013 elections in Kenya. Human Rights Watch.

[5] Mossie, Z., Wang, J.H. (2020). Vulnerable community identification using hate speech detection on social media. Information Processing & Management, 57(3): 102087. https://doi.org/10.1016/j.ipm.2019.102087

[6] Yimam, S.M., Ayele, A.A., Biemann, C. (2019). Analysis of the Ethiopic Twitter dataset for abusive speech in Amharic. arXiv preprint arXiv:1912.04419. https://doi.org/10.48550/arXiv.1912.04419

[7] Bayer, J., Petra, B.Á.R.D. (2020). Hate speech and hate crime in the EU and the evaluation of online content regulation approaches. Policy Department for Citizens' Rights and Constitutional Affairs.

[8] Badjatiya, P., Gupta, S., Gupta, M., Varma, V. (2017). Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759-760. https://doi.org/10.1145/3041021.3054223

[9] Elouali, A., Elberrichi, Z., Elouali, N. (2020). Hate speech detection on multilingual twitter using convolutional neural networks. Revue d'Intelligence Artificielle, 34(1): 81-88. https://doi.org/10.18280/ria.340111

[10] Gambäck, B., Sikdar, U.K. (2017). Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, pp. 85-90. https://doi.org/10.18653/v1/w17-3013

[11] Ribeiro, A., Silva, N. (2019). INF-HatEval at SemEval-2019 Task 5: Convolutional neural networks for hate speech detection against women and immigrants on twitter. In Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 420-425. https://doi.org/10.18653/v1/s19-2074.

[12] Yoon, K. (2014). Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751. https://doi.org/10.3115/v1/d14-1181

[13] Massey, C.R. (1992). Hate speech, cultural diversity, and the foundational paradigms of free expression. UCLA L. Rev., 40: 103. Available at: https://repository.uchastings.edu/faculty_scholarship/1376.

[14] Moran, M. (1994). Talking about hate speech: A rhetorical analysis of American and Canadian approaches to the regulation of hate speech. Wis. L. Rev., 1425.

[15] Ward, K.D. (2016). Free speech and the development of liberal virtues: An examination of the controversies involving flag-burning and hate speech. 52 UMIA Law Review, 733. Available at: https://repository.law.miami.edu/umlr/vol52/iss3/4.

[16] Brown, A. (2017). What is hate speech? Part 1: The myth of hate. Law and Philosophy, 36(4): 419-468. https://doi.org/10.1007/s10982-017-9297-1

[17] Facebook. (2022). Hate speech. https://m.facebook.com/communitystandards/objectionable_content/?privacy_mutation_token=eyJ0eXBlIjowL

CJjcmVhdGlvbl90aW1lIjoxNjQ0MTA0MTY5LCJjYWxsc2l0ZV9pZCI6Mzg3OTcxMjgyMzgwNTA4fQ%3D%3D.
[18] Youtube (2022). How does YouTube protect the community from hate and harassment? https://www.youtube.com/howyoutubeworks/our-commitments/standing-up-to-hate/.
[19] Tweeter (2022). Hateful conduct policy. https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy.
[20] Chen, Y., Zhou, Y., Zhu, S., Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. In 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 71-80. https://doi.org/10.1109/SocialComPASSAT.2012.55
[21] Haralambous, Y., Lenca, P. (2014). Text classification using association rules, dependency pruning and hyperonymization. arXiv preprint arXiv:1407.7357. https://doi.org/10.48550/arXiv.1407.7357
[22] 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:1809.08651. https://doi.org/10.48550/arXiv.1809.08651
[23] Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N. (2015). Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on World Wide Web, pp. 29-30. https://doi.org/10.1145/2740908.2742760
[24] Chen, K., Zhao, T., Yang, M., Liu, L., Tamura, A., Wang, R., Sumita, E. (2017). A neural approach to source dependence based context model for statistical machine translation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(2): 266-280. 10.1109/TASLP.2017.2772846
[25] Warner, W., Hirschberg, J. (2012). Detecting hate speech on the world wide web. In Proceedings of the Second Workshop on Language in Social Media, pp. 19-26.
[26] van Dinter, R., Catal, C., Tekinerdogan, B. (2021). A multi-channel convolutional neural network approach to automate the citation screening process. Applied Soft Computing, 112: 107765. https://doi.org/10.1016/j.asoc.2021.107765
[27] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12: 2493-2537.
[28] Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G., Plagianakos, V.P. (2018). Convolutional neural networks for toxic comment classification. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence, pp. 1-6. https://doi.org/10.1145/3200947.3208069
[29] Brownlee, J. (2018). How to develop a multichannel CNN model for text classification. Deep Learning for Natural Language Processing. https://machinelearningmastery.com/develop-n-gram-multichannel-convolutional-neural-network-sentiment-analysis/.
[30] Dahou, A., Xiong, S., Zhou, J., Elaziz, M.A. (2019). Multi-channel embedding convolutional neural network model for Arabic sentiment classification. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 18(4): 1-23. https://doi.org/10.1145/3314941
[31] Wang, Z., Pedersen, T. (2018). Umdsub at semeval-2018 task 2: Multilingual emoji prediction multi-channel convolutional neural network on subword embedding. arXiv preprint arXiv:1805.10274. https://doi.org/10.48550/arXiv.1805.10274
[32] Alotaibi, M., Alotaibi, B., Razaque, A. (2021). A multichannel deep learning framework for cyberbullying detection on social media. Electronics, 10(21): 2664. https://doi.org/10.3390/electronics10212664
[33] Shah, V., Udmale, S.S., Sambhe, V., Bhole, A. (2021). A deep hybrid approach for hate speech analysis. In International Conference on Computer Analysis of Images and Patterns, 424-433. https://doi.org/10.1007/978-3-030-89128-2_41
[34] Kassa, Z.A. (2021). Amharic hate speech detection dataset. https://doi.org/10.5281/zenodo.5036437
[35] Davidson, T., Warmsley, D., Macy, M., Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Proceedings of the International AAAI Conference on Web and Social Media, 11(1): 512-515.
[36] De Gibert, O., Perez, N., García-Pablos, A., Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. arXiv preprint arXiv:1809.04444. https://doi.org/10.48550/arXiv.1809.04444
[37] Eshetu, A. Text preprocessing for Amharic. https://abe2g.github.io/am-preprocess.html.
[38] 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, 26: 3111-3119.
[39] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5: 135-146. https://doi.org/10.1162/tacl_a_00051
[40] Eshetu, A., Teshome, G., Abebe, T. (2020). Learning word and sub-word vectors for Amharic (less resourced language). International Journal of Advanced Engineering Research and Science (IJAERS), 7(8): 358-366. https://doi.org/10.22161/ijaers.78.39
[41] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

APPENDIX

Appendix A: SVM parameter settings, running-time and performance

Dataset	svm.SVC (classifier) parameter settings	n-gram	TfidfVectorizer (min_df =x, max_df =, y analyzer='word', ngram_range=(,), use_idf= True)		Vocabulary size	Training time	Testing time	Performance		
			x	y				P	R	F1
Amh	kernel= 'linear', gamma= 'auto', C= 2	1	1	1	15,647	0.326652s	0.018720s	74.8	78.4	76.2
			10	0.4	517	0.120911s	0.015624s	**83.7**	**88.2**	**85.8**
			5	0.3	1,355	0.181143s	0.031243s	83.0	87.3	85.0
		2	1	1	**83,826**	**3.468758s**	0.062626s	**74.6**	**76.8**	**74.3**
			10	0.4	114	0.039973s	0.005995s	63.2	58.4	49.3
			5	0.3	452	0.063960s	0.004995s	61.7	60.0	52.8
		combined	1	1	**99,473**	**4.364228s**	0.093750s	76.2	80.0	77.8
			10	0.4	631	0.130920s	0.031243s	**83.2**	**87.6**	**85.3**
			5	0.3	1,808	0.178770s	0.031247s	82.5	86.8	84.5
Whs	kernel= 'sigmoid', coef0=0.001, C= 3	1	1	1	3,635	0.488615s	0.021986s	50	1.0	67.1
			10	0.4	449	0.358618s	0.062492s	**67.6**	**67.0**	**66.8**
			5	0.3	912	0.472130s	0.048921s	67.2	64.8	63.4
		2	1	1	**39,996**	**2.288705s**	0.069340s	50.0	1.0	67.1
			10	0.4	340	0.160142s	0.015619s	57.6	57.3	56.6
			5	0.3	1,166	0.190387s	0.015621s	58.0	57.3	56.2
		combined	1	1	**43,631**	**2.864083s**	0.086946s	50.0	1.0	67.1
			10	0.4	789	0.394502s	0.063701s	65.0	64.0	63.3
			5	0.3	2,078	0.579860s	0.062492s	69.3	63.8	60.8
Dvd	kernel= 'linear', gamma= 'auto', C= 2	1	1	1	2,950	0.259517s	0.015628s	56.1	55.1	53.7
			10	0.4	285	0.126484s	0.015618s	**68.4**	**68.0**	**67.8**
			5	0.3	620	0.146517s	0.031245s	66.6	66.2	66.0
		2	1	1	**26,760**	**1.135688s**	0.015622s	55.2	54.6	53.8
			10	0.4	156	0.046965s	0.003997s	59.9	58.9	58.1
			5	0.3	564	0.069958s	0.006993s	61.2	60.2	59.4
		combined	1	1	**29,710**	**1.204441s**	0.015622s	52.5	52.3	52.2
			10	0.4	441	0.139507s	0.015621s	68.0	67.5	67.3
			5	0.3	1,183	0.176772s	0.031249s	67.7	67.5	67.4