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

Revue d'Intelligence Artificielle
Vol. 34, No. 1, February, 2020, pp. 81-88
Journal homepage: http://iieta.org/journals/ria

Hate Speech Detection on Multilingual Twitter Using Convolutional Neural Networks

Aya Elouali[1*], Zakaria Elberrichi[1], Nadia Elouali[2]

[1] EEDIS Laboratory, Djillali Liabes University, Sidi Belabbes 22000, Algeria
[2] LabRi Laboratory, Ecole Superieure en Informatique, Sidi Bel Abbes 22016, Algeria

Corresponding Author Email: n.elouali@esi-sba.dz

https://doi.org/10.18280/ria.340111

Received: 18 October 2019
Accepted: 29 December 2019

Keywords:
neural networks, hate speech, multilingual, convolutional neural network, text classification, character level representation

ABSTRACT

Hate speech detection on Twitter is often treated in monolingual (in English generally) ignoring the fact that Twitter is a global platform where everyone expresses himself with his natal language. In this paper, we created a model which, taking benefits of the advantages of neural networks, classifies tweets written in seven different languages (and even those that contains more than one language at the same time) to hate speech or non hate speech. We used Convolutional Neural Networks (CNN) and character level representation. We carried out several experiments in order to adjust the parameters according to our case study. Our best results were (in terms of accuracy) 0.8893 for a dataset containing five languages and 0.8300 for a dataset of seven languages. Our model solves properly the problem of hate speech on Twitter and its results are, compared to the state of the art, more than satisfactory.

1. INTRODUCTION

Twitter is today one of the most used social networks in the world. It allows users to express their feelings, ideas and opinions through 280-character mini-texts. Since everything is virtual, people dare to say what they cannot say in real life, such as racist or sexist expressions. Hence, hate speech that we were thinking was a thing of the past, have just moved to a different venue [1]. The term hate speech was defined as "any communication that disparages a person or a group on the basis of some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics." [2].

The challenge is to be able to detect and eliminate this type of tweet, as they incite to commit crimes and violent acts. So much so, in April 2017, the German government decided to impose a fine up to 50 million euros on social networks if they fail to remove hateful postings quickly [3]. Twitter is fighting this problem. However, it is still being criticized for not doing enough, which is largely because it involves manual reviewing and therefore it is labor intensive and time-consuming [4].

Hate speech detection, or opinion detection in general, has always been an interesting subject for researchers in different scientific fields such as Machine Learning and Natural Language Processing (NLP). However, scaling this type of solution is usually difficult because of preprocessing on the data [5]. Deep learning is a research field that has received a lot of attention in recent years. It is a subset of machine learning, used in machine translation, speech interpretation, image and/or object recognition and even in natural language processing such as sentiment analysis, opinion mining, etc.

On the other hand, since Twitter is used all over the world, tweets are written in different languages. Therefore, analyzing Twitter data should not be done in one language, because even English is used in 34% of tweets only [6]. Older solutions for multilingual problems were to translate all data into one language and then apply machine learning algorithms on the translated data. However, no translation is perfect, and even if we could have a perfect translation, there are also cultural differences from one country to another, which can influence the performance of the created model [7].

Neural networks do not require translation. Therefore, unlike traditional solutions, we can even classify tweets that contain more than one language [7].

In this work, we present an effective method for classifying tweets in different languages as hate speech or non hate speech by taking advantage of the powerful characteristics of neural networks.

The paper is organized as follows. In the next section, we discuss related work. Then, sections 3 and 4 present the datasets building as well as the data cleaning process. We give details on the architecture of our approach in section 5. In Section 6, the different results of experiments are presented and discussed. Section 7 concludes the paper and outlines our future work.

2. STATE OF THE ART

We grouped text classification works into two categories. The first category includes research about the broader domain of sentiment analysis while the second regroups works concerning specifically hate speech detection.

2.1 Sentiment analysis

According to the study [8], sentiment analysis of short texts (tweets for example), is difficult because of the limited amount of contextual data. Solving such problems effectively requires strategies that go beyond the bag-of-words. Dos Santos and Gatti [8] presents a new neural network architecture that uses character-level, word-level, and sentence-level representations

to analyze feelings.

Let a sentence consist of n words, each word is converted into a vector, which is composed of two sub-vectors: word-level embedding and character-level embedding. While word-level embedding is intended to capture syntactic and semantic information, character-level embedding captures form information. For character-level embeddings they represented characters in one word differently, some characters are given more weight than others. For example:

• Hashtags: for a text analysis hashtag are considered one word, but their letters do not have the same weight. For example, “S”, “a” and “d” are given more weight compared to other letters in "#SoSad”.

• Adverbs: Similarly, “l” and “y” letters are given less weight than the other letters in "beautifully", "perfectly" and "badly".

Authors used a convolutional neural network for classification and the best accuracy was 0.864.

The approach proposed by Araque et al. [9] uses a recurrent neural network (RNN) composed of Long Short-Term Memory (LSTM) cells to do sentiment analysis on tweets written in Spanish. They used sentiment word level representation by adding the feeling associated with each word:

1. First, every tweet is broken down into words.

2. A dictionary of feelings is used to classify each word (positive, neutral or negative).

3. The word representation is the concatenation of the vector representing the word (word level representation) and the class of feeling associated.

The proposed method has been tested on different datasets twice:

a- with word level representation only.

b- with word level representation + feelings.

The use of word’s feeling has improved the results for the first two of three datasets by a difference of 0.036 of accuracy.

Old feature representation methods do not consider the context of the word. Lai et al. [10] used a recurrent bidirectional structure in order to represent the word according to its context. This structure is inserted as a first place in a convolutional neural network to classify the inserted text. Three datasets were used (20Newsgroupy [11], Fudan set, ACL Anthology Network [12], Stanford Sentiment Treebank [13]) to test the effectiveness of the architecture.

The proposed solution has shown the best results (except for the last dataset) compared to other methods and even compared to the ordinary CNN (without the recurrent structure). The reason, according to the authors, is that the recurrent structure captures better contextual information.

Yuan and Zhou [14] used different types of recursive neural networks (RNNs) to do sentiment analysis on Twitter. Tweets are divided into words and then converted to a binary tree where each non leaf node has exactly two children. To classify these tweets as positive, neutral or negative, they used three types of RNNs: RNN with a one hidden layer, RNN with two hidden layers and a RNTN (Recursive Neural Tensor Network). They also used a regularization method to avoid the problem of over-learning.

The RNN with one hidden layer and the RNN with two hidden layers gave approximately the same result (0.6371, 0.6245 of accuracy and 0.512, 0.517 of F score respectively) but the RNTN architecture gave worse results (0.5932 of accuracy and 0.483 of F score).

Becker et al. [7] used a neural network to do multilingual sentiment analysis on Twitter since there is no need to translate or separate languages. They used character level as they found it more practical (the matrix’s size is reduced and permit avoiding conflicts between languages). They also used different architectures of CNN while proposing their own architecture (Conv-Char-R) which consists of changing the size and number of convolutional and pooling layers.

Articles in this category present different and interesting solutions for sentiment analysis for short texts (considering the context, adding words’ feeling, etc.). However, the integration of the multilingual aspect remains limited.

2.2 Hate speech detection

Warner and Hirschberg [15] presents a hate speech detection approach for online texts. They define hate speech as abusive speech targeting specific characteristics of a group, such as ethnicity, religion, or gender. The authors used a template-based strategy of the study [16] to generate features from texts and inject them as input to an SVM classifier. They obtained an accuracy of 0.94, a precision of 0.68 and recall at 0.60, and an F1 of 0.675.

According to the study [17], in Twitter, hateful tweets are those that contain abusive words targeting particular individuals or groups. Detecting this type of tweet is important for the feelings of one group of users towards another, deter terrorist actions and filter tweets before a recommendation. This article aims to classify tweets into racist, sexist or none by using different methods and then making a comparison between the results.

Methods used:

• Baseline Methods: use different word representations (TF-IDF, Bag of Words, Char n-grams) with traditional classifiers.

• DNNs: use three types of neural network: CNN, LSTM, FastText.

• DNNs + GBDTs: Combine the neural networks used in the second experiment with the classifier GBDTs (Gradient Boosted Decision Trees). DNNs are used for feature extraction for GBDTs.

The methods DNNs and DNNs+ GBDTs proposed in this article performed better than traditional solutions. The architecture using LSTM + Random Embedding + GBDT gave the best result (0.930 of F score): the representation of tweets was initialized with random vectors (Random Embedding), then LSTM was applied on these vectors after that the generated features were used to train the GBDT classifier.

In the same context, the study [18] is a research work about offensive tweets’ detection tested with a dataset in Hinglish language (Hindi words written with English alphabet). They used the Transfer Learning method which consists of reusing a model that was created and trained for one task, as the starting point for a similar task. It is to benefit from the learning of one neural network by using the learned weights. In this case, they reused the weights of a convolutional neural network trained on a dataset in English.

The comparison was made between the results of the initial neural network trained with dataset A (in English), a normal CNN with HEOT dataset (in Hinglish) and using transfer learning method with HEOT dataset. The results were 0.754, 0.587 and 0.839 of accuracy respectively.

For hate speech detection in tweets, Georgios et al. [19] proposes an RNN (Recurrent Neural Network), which integrates various characteristics such as the tendency of the user to racism or sexism. For each tweet, they add the user's

tendencies:

tN, a = | mN, a | / | ma |

tR, a = | mR, a | / | ma |

tS, a = | mS, a | / | ma |

Where tN,a, tR,a, tS,a are the neutral, racist and sexist tendencies of the user a.

| mN,a |, | mR,a |, | mS,a | represent respectively the numbers of the neutral, racist and sexist tweets of the user a.

| ma |: number of tweets of the user a.

They used several LSTMs (Long Short-Term Memory) and a dataset of 16k tweets (1943 racist tweets, 3166 sexist tweets, 10889 neutral tweets and tweets that belong to more than one class) to test the adding of the tendencies gradually.

Compared to the LSTM with no additions, adding contextual features improved performance and even compared to other search's results. The combination of several LSTMs gives results that are even more relevant.

Zhang et al. [4] presents a method based on combining convolutional neural networks (CNN) and Long Short-Term Memory (LSTM).

The first layer is for word embedding, it represents each message by a real number vector (dimension = 300). Each sequence is of dimension of 100. This is done by truncating the long messages and completing the short messages by zeros. The output feeds a 1D convolutional layer of 100 filters with a window 4*4. They used Rectified Linear Unit (ReLU) as an activation function. This converts the input space to a 100 × 100 representation. This is then down-sampled by a 1D max pooling layer with a pool size of 4, producing a 25 × 100 shape output. Each of the 25 dimensions is considered as an extracted feature. The LSTM layer processes the extracted feature as time steps and generates 100 hidden units for each time step.

They used seven datasets: WZ-L [20], WZS.amt [21], WZ-S.exp [21], WZ-SGb [22], WZ-LS [23], DT [24], RM (created by the authors). The proposed architecture performed better than the state of the art in 6/7 of the datasets.

The high production rate of data in social media makes it difficult to collect, store, and analyze all this data using traditional methods. Thus, Zewdie and Jenq-Haur [25] defined a classification of hate speech in a Big Data context. The authors developed a model based on Apache Spark to classify Facebook posts written in Amharic into hate speech or non hate speech. They used Random Forest and Naïve Bayes as classifiers and Word2Vec and TF-IDF for feature selection. The best result was 0.7983 of accuracy.

The work [26] aims to classify English and Hindi sentences/posts to manifestly aggressive, secretly aggressive or non-aggressive. The representation of the words was done with fastText (a Word2vec extension) which represents each word by N-gram of characters. The vector of a word is the sum of its N-gram characters. Since social network users make a lot of spelling and typing errors, fastText is more convenient than Glove and Word2vec which consider the word as a single unit represented by a vector.

Several architectures were used to classify the data:

- Bidirectional LSTM
- Single LSTM with higher dropout
- Model based on Neural Network Convolution
- Model based on Neural Network Convolution with different Filter height
- Model based Bidirectional GRU and Convolution Neural Network
- Voting based ensemble model
- Model based on Logistic Regression Deep learning methods with fast Text and the necessary parameterization performed better than the traditional data mining algorithms for this problem of aggression detection. This article was considered as multilingual, but in fact they just used two languages (Hindi and English) and they even used it separately.

As for articles in the sentiment analysis category, hate speech detection articles present interesting solutions. However, the integration of the multilingual aspect remains limited. On sentiment analysis, the study [7] was the only work that used multiple languages. In the other works, and especially concerning hate speech detection, the number of languages considered generally does not exceed two languages. This affects the results generalization and limits the automatic detection of hate speech. Thus, the main purpose of this paper is to deal with (1) multilingual (2) hate speech detection (3) on Twitter (4) using deep learning methods. Twitter represents our use case. However, our approach can be generalized for any type of short text.

3. BUILDING THE DATASET

In deep learning or machine learning, the dataset plays a very important role. The dataset's size, type and the data distribution are factors that influence the model's performance. So, the first step of our work was to define the dataset. However, no multilingual hate speech dataset was available. So, instead of creating a brand new dataset, classifying millions of tweets to hate speech and non hate speech, we decided to use existing datasets in different languages and combine/unify them to have one multilingual dataset.

The creation of our dataset has gone through several stages/versions:

1. The first version of the dataset contains:

(a) The dataset [27] "Religious Hate Speech Detection for Arabic Tweets" [28]: 5569 tweets for the training and 567 for the test. However, since we aren't only using this dataset, the test data cannot be chosen this way, so we combined the two parts (train and test) of this dataset and we choose our own test data from the final dataset.

(b) The dataset [29] "Italian Twitter Corpus of Hate Speech" [30]: 1827 tweets about immigrants, Muslims and Roma.

(c) The research's dataset [31] "Hate speech dataset annotated for Portuguese" [32]: 5668 tweets manually annotated, collected from 1156 distinct user accounts.

(d) The dataset [33] "id-hatespeech-detection" [34]: 713 tweets in Indonesian language, 453 of them classified as non-Hate Speech and 260 as Hate speech.

(e) The "Automated Hate Speech Detection and the Problem of Offensive Language" [24, 35] dataset: 24784 tweets in English.

2. The second version of the dataset contains in addition to the five listed datasets:

(a) The dataset [36] "GermEval-2018 data repository" [37]: 8541 tweets in German language manually annotated by German annotators.

(b) The "IWG hatespeech public" dataset [38]: German corpus of annotated tweets containing hate speech against the refugees in Germany.

(c) The dataset [39] "HateSpeech Hindi-English Code Mixed Social Media Text" [40]: 4575 tweets in Hindi-English Code-Mixed language (a tweet contains both Hindi and English at the same time) including 1661 tweets containing hate speech.

Thus, the first version contains 6136 tweets in Arabic, 1827 in Italian, 5668 in Portuguese, 713 in Indonesian and 24784 in English. The second version contains, in addition to the existing tweets, 4575 tweets in Hindi-English and 9010 in German (both datasets).

Twitter allows data scientists to use its data. However, strict conditions are imposed to protect their platform. One of the conditions is that tweets and user's information are confidential and cannot be published on the internet. For this, most of the datasets we used did not contain the text of the tweet, it is usually replaced by its ID (unique identifier of the tweet), and so we used Twitter developer API [41] to convert IDs into corresponding tweet texts.

Unfortunately, not all tweets can be recovered because of issues such as "Account suspended", "Tweet deleted" and "not allowed to access tweet" if the account is private. The tweets we couldn't convert were deleted from the dataset. Thus, the number of tweets decreased to: 4085 in Arabic, 1425 in Italian, 2721 in Portuguese, 713 in Indonesian, 24783 in English, 3570 in Hindi-English and 8876 in German. This gives a total of 33727 in the first version and 46173 in the second.

4. DATA CLEANING

Since we are using tweets as data, we know which kind of information is useless for our study such as mentions, punctuation, etc. So, we decided to remove them for better performance.

Several cleaning operations were applied:

1. HTML decoding: Some HTML parts can't be correctly converted in text such as "& amp;", "& quot", etc. We used BeautifulSoup [42] (Python library, data extraction from HTML and XML files) to do HTML decoding.

2. Deleting mentions: Even if mentions contain information about the tweet (calling another user to see this tweet), this information adds no value to the hate speech detection problem. So, we proceeded by deleting them.

3. Deleting URL links: Just like mentions. Even though URLs contain information, they can be ignored for the detection of hate speech.

4. Removing Hashtags and special characters: Sometimes, the text used with the hashtag can provide useful information about the tweet. It could be a little risky to get rid of all the text. For that we removed the "#" only. Also, we deleted special characters such as "!" ":" "; ", etc.

5. Deleting diacritics: Arabic words with diacritics are not correctly retrieved in text format, so we used " pyarabic" [43] library to delete them.

5. THE ARCHITECTURE OF OUR APPROACH

The model we used for multilingual hate speech detection on twitter, was inspired by the architecture [44]. They presented a very interesting CNN architecture for text classification with character level representation.

The initial version of the architecture (Figure 1) contains:

- Embedding layer.
- Six convolutional layers (256 filters of 7*7 and 3*3) three of them followed by a pooling layer (window of 3*3).
- Two fully connected layers, each containing 1024 neurons.
- Output layer, the number of neurons depends on the number of classes of the problem being treated (2 in our case).

We chose character-level representation to represent tweets for the following reasons:

1. The size of the representation matrix is reduced (number of characters).

2. Avoiding the out-of-vocabulary (OOV) problem: Out-of-vocabulary words (OOV) are words that exist in test data, but do not exist in the training data and therefore they will be misclassified.

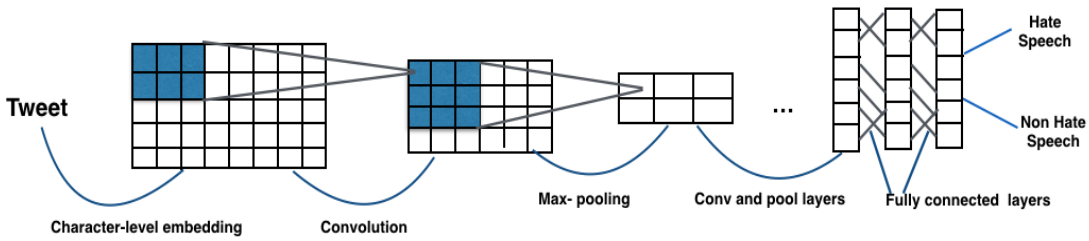

Figure 1. The architecture of our network

3. With a multilingual aspect, using word representation, syntactically identical words, but with different meanings across languages, will certainly confuse the model and harm the classification performance [3].

As CNNs take a fixed size of data input, so we added a padding of zeros. We used two padding sizes: 500 for tweets before the data cleaning and 280 (maximum length of a tweet) after data cleaning. So, each sentence is represented by a vector of 500 (or 280) characters (Figure 2). We represent each of these characters by a vector of 0 which contains a 1 in the position of the character concerned (Figure 3).

```
[61 11 68 69  8  2 15  8 18  3  2 18  3 14 55  2 18  8 12 17  3  2 18  5
 12  9 18 22  8 11 17  8 61  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
```

Figure 2. Tweet's padding with zeros

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

Figure 3. Character representation with a 0,1 vector

6. EXPERIMENTS

Several experiments were carried on, in order to define the best neural network architecture for multilingual hate speech detection. In each experiment, we changed a detail in the architecture such as the number of convolutional layers, the number of filters, the optimizer, etc. Table 1 includes all experiments and results.

Table 1. Experiments and results

Exp	Model	Dataset	Best Accuracy	Best Loss	Number of epochs	Time (in seconds)
1	6 convolutional layers, 2 dense layers	Not cleaned first version of the dataset	0.8835 (3rd epoch)	0.2896 (3rd epoch)	10	46861
2	6 convolutional layers, 2 dense layers	Cleaned first version of the dataset	0.8651 (3rd epoch)	0.3110 (3rd epoch)	10	52263
3	3 convolutional layers, 100 filters	Cleaned first version of the dataset	0.8746 (5th epoch)	0.3151 (5th epoch)	10	1048
4	3 convolutional layers, 256 filters batch size = 200.	Cleaned first version of the dataset + german dataset	0.7822 (5th epoch)	0.4866 (5th epoch)	15	7590
5	3 convolutional layers, 256 filters.	Cleaned first version of the dataset + Hindi-English and one German datasets	0.7851 (3rd epoch)	0.4864 (3rd epoch)	10	4448
6	3 convolutional layers , 256 filters	Cleaned second version of the dataset	0.7794 (3rd epoch)	0.4829 (3rd epoch)	10	5988
7	6 convolutional layers, 256 filters.	Cleaned second version of the dataset	0.7649 (3rd epoch)	0.4997 (3rd epoch)	10	6388
8	3 convolutional layers, 100, SGD	Cleaned first version of the dataset	0.8893 (67th epoch)	0.2554 (67th epoch)	100	40978
9	3 convolutional layers, 100 filters, SGD	Cleaned second version of the dataset	0.8300 (37th epoch)	0.3494 (37th epoch)	100	13801
10	Word level embedding	Cleaned first version of the dataset	0.5965	0.6629	14	55617
11	Word embedding	Cleaned second version of the dataset	0.6934	0.6145	10	635

In experiments 1 and 2, we tested the first version of our dataset before and after data cleaning on the defined architecture. We didn't attend progress in results by adding data cleaning, but it still eases the work of the neural network.

In experiment 3, we minimized the number of convolutional layers to three and the number of filters to 100. We got approximately the same result we obtained using six layers with a time of execution 49 times less (for ten epochs: 52263s using six layers, 1048 s using three layers).

In experiments 4, 5, 6, we gradually added the two German and the Hindi-English datasets in order to test the impact of adding a new language on the results of our neural network. We can easily notice that the addition of languages caused a small deterioration in performance. Especially for German, since the Hindi-English dataset contained English so it was not so difficult to classify the tweets in this dataset. This deterioration is justified, as new information has been added to the neural network without sufficient examples (3570 in Hindi-English and 8876 in German). However, we believe that classifying tweets in one language with a mediocre rate is better than not being able to classify them at all.

The seventh experiment was just to confirm that going back to the old architecture (with six convolutional layers) won't give better results (which means that results' deterioration is due to the language adding not to the architecture).

In previous experiences, we used Adam [45] as an optimizer. Then, we decided to change to SGD (Stochastic gradient descent) [45] hoping to get a better weight adjustment and therefore better results. The best result of experiment 8 was 0.8893 of accuracy, result of the 67th epoch. While the best result of experiment 9 (37th epoch) was 0.8300 of accuracy. We can see that the use of SGD optimizer improved the results for both versions of the dataset. Another advantage is that before the overfitting the gap between the train and test curves (Figure 4 and 5) was too small which proves the effectiveness of the learning process. In addition, the model took more time (epochs) to learn and this leads to better generalization when new data is presented to the model [46].

The model started overfitting from epoch 68 for experiment 8 and epoch 38 for experiment 9. We did not use "Early Stopping" method just to see the behavior of the curves and to confirm that we did not stop at a local maximum.

In order to compare our results and confirm our hypothesis about character level embedding, we created another model using word level embedding. We tested it with the first and the second version of our dataset. As expected, the results were not satisfying. The accuracy got stuck at 0.5965 for the first version and at 0.6934 for the second.

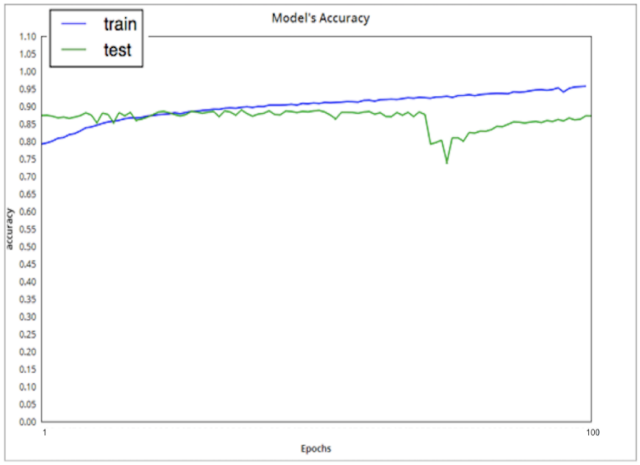

Figure 4. Accuracy for 100 epochs of training with three convolutional layers and SGD optimizer on the first version of the dataset

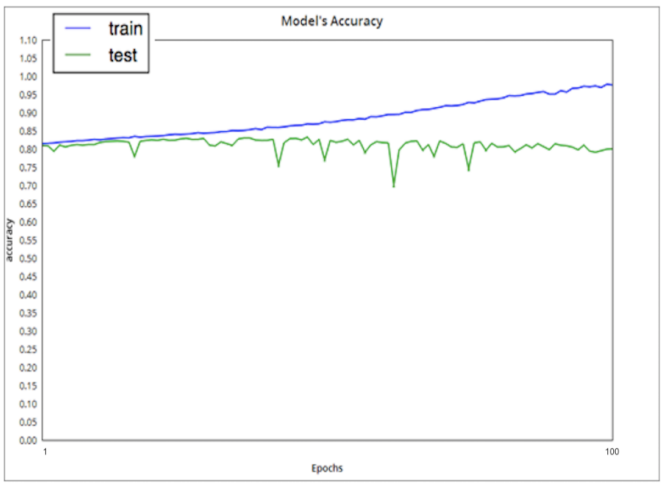

Figure 5. Accuracy for 100 epochs of training with three convolutional layers and SGD optimizer on the second version of the dataset

Table 2 presents a comparison between our results and the results of other works using the datasets we used separately:

Our neural network performed better than most of the native datasets' results. In addition, our model has several advantages over separate models:

- The ability to classify tweets containing multiple languages.- Creating and training just one model.
- Gain in terms of data (use all data to train one model).

Table 2. Comparison of results

Dataset	Best result
Arabic: [27]	Fmesure=0.60
Italian: [29]	/
Portuguese: [31]	Accuracy=0.783
Indonesian: [33]	Fmesure=89.8
English: [24]	Fmesure=0.90
German:[36]	Fmesure=0.76
German: [38]	/
Hindi-English: [39]	Accuracy=0.717
Our work	Accuracy = 0.8893 (first dataset) 0.8300 (second dataset)

7. CONCLUSION AND FUTURE WORK

In this paper, we presented an effective method to classify tweets in different languages as hate speech or non hate speech by taking advantage of the powerful features of deep learning. We defined a model based on the use of CNNs for text classification with character level representation. The various experiments carried out, using our dataset containing seven different languages, showed the effectiveness of our model for hate speech detection in a multilingual context.

To our knowledge, there is no approach in the literature that has grouped these three parameters (1- automatic and 2-multilingual 3- hate speech detection on Twitter's content) and has given good results like our method's. Thus, this work is a step towards an automatic elimination of hate speech from social networks.

To improve our results, we will explore future work in the following directions:

• Doing more experiments by changing more parameters: number of neurons of the fully connected layer, dropout rate, etc.

• Modifying the architecture by adding LSTM layer (s).

• Trying the n-gram level embedding hoping to find a new representation track.

REFERENCES

[1] Tracking racism on Twitter. https://www.folio.ca/tracking-racism-on-twitter/, accessed on 2 January 2020.

[2] Nockleby, J.T. (2000). Hate speech. Encyclopedia of the American Constitution (2nd ed., edited by Leonard W. Levy, Kenneth L. Karst et al., New York: Macmillan, 2000), pp. 1277-1279.

[3] German cabinet agrees to fine social media over hate speech. http://uk.reuters.com/article/idUKKBN1771FK/, accessed on 2 January 2020.

[4] Zhang, Z.Q., Robinson, D., TepperHate, J. (2018). Hate speech detection using a convolution-LSTM based deep neural network. Proceedings of WWW' 2018, Lyon. https://doi.org/10.475/123_4

[5] Leveraging Deep Learning for Multilingual Sentiment Analysis. http://blog.aylien.com/leveraging-deep-learning-for-multilingual/, accessed on 2 January 2020.

[6] Most-used languages on Twitter as of September 2013. https://www.statista.com/statistics/267129/most-used-languageson-twitter/, accessed on 2 January 2020.

[7] Becker, W., Wehrmann, J., Cagnini, H.E.L., Barros, R.C. (2017). An efficient deep neural architecture for multilingual sentiment analysis in twitter. Proceedings of FLAIRS Conference, pp. 246-251.

[8] Dos Santos, C., Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 69-78.

[9] Araque, O., Barbado, R., Sánchez-Rada, J.F., Iglesias, C.A. (2017). Applying recurrent neural networks to sentiment analysis of Spanish tweets. Proceedings of Tass-Sepln, pp. 71-76.

[10] Lai, S., Xu, L., Liu, K., Zhao, J. (2015). Recurrent convolutional neural networks for text classification. Proceedings of the Twenty-Ninth AAAI Conference on

Artificial Intelligence, pp. 2267-2273. https://doi.org/10.5555/2886521.2886636
[11] Hingmire, S., Chougule, S., Palshikar, G.K., Chakraborti, S. (2013). Document classification by topic labeling. Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 877-880. https://doi.org/10.1145/2484028.2484140
[12] Post, M., Bergsma, S. (2013). Explicit and implicit syntactic features for text classification. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2(Short Papers), pp. 866-872. https://www.aclweb.org/anthology/P13-2150.pdf
[13] Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631-1642.
[14] Yuan, Y., Zhou, Y. (2015). Twitter sentiment analysis with recursive neural networks. CS224D Course Projects.
[15] Warner, W., Hirschberg, J. (2012). Detecting hate speech on the World Wide Web. Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012), pp. 19-26. https://doi.org/10.5555/2390374.2390377
[16] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In ACL-95, Cambridge, pp. 189-196. https://doi.org/10.3115/981658.981684
[17] Badjatiya, P., Gupta, S., Gupta, M., Varma, V. (2017) Deep learning for hate speech detection in tweets. Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759-760. https://doi.org/10.1145/3041021.3054223
[18] Mathur, P., Ratn Shah, R., Sawhney, R., Mahata, D. (2018). Detecting offensive tweets in Hindi-English code-switched language. Proceedings of SocialNLP@ACL, pp. 18-26. https://doi.org/10.18653/v1/W18-3504
[19] Georgios, K.P., Heri, R., Helge, L. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. Proceedings of Applied Intelligence, 48(12): 4730-4742. https://doi.org/10.1007/s10489-018-1242-y
[20] Waseem, Z., Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. Proceedings of the NAACL Student Research Workshop. Association for Computational Linguistics, San Diego, California, pp. 88-93. https://doi.org/10.18653/v1/N16-2013
[21] Waseem, Z. (2016). Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. Proceedings of EMNLP Workshop on Natural Language Processing and Computational Social Science, ACL, Austin, Texas, pp. 138-142. https://doi.org/10.18653/v1/W16-5618
[22] Gambäck, B., Sikdar, U.K. (2017). Using convolutional neural networks to classify Hate-speech. Proceedings of the First Workshop on Abusive Language Online, pp. 85-90. https://doi.org/10.18653/v1/W17-3013
[23] Park, J.H., Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter. arXiv preprint arXiv:1706.01206.
[24] Davidson, T., Warmsley, D., Macy, M., Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Eleventh International AAAI Conference on Web and Social Media.
[25] Zewdie, M., Jenq-Haur, W. (2018). Social network hate speech detection for Amharic language. Proceedings of Computer Science & Information Technology (CS & IT), pp. 41-55. https://doi.org/10.5121/csit.2018.80604
[26] Modha, S., Majumder, P., Mandl, T. (2018). Filtering aggression from the multilingual social media feed. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 199-207.
[27] Mubarak, H., Darwish, K., Magdy, W. (2017). Abusive language detection on Arabic social media. Proceedings of the First Workshop on Abusive Language Online, pp. 52-56. https://doi.org/10.18653/v1/W17-3008
[28] Religious Hate Speech Detection for Arabic Tweets. https://github.com/nuhaalbadi/Arabic_hatespeech, accessed on 2 January 2020.
[29] Sanguinetti, M., Poletto, F., Bosco, C., Patti, V., Stranisci, M. (2018). An Italian twitter corpus of hate speech against immigrants. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).
[30] Italian Twitter Corpus of Hate Speech. https://github.com/msang/hate-speech-corpus#italian-twittercorpus-of-hate-speech, accessed on 2 January 2020.
[31] Fortuna, P.C.T. (2017). Automatic detection of hate speech in text: An overview of the topic and dataset annotation with hierarchical classes. Thesis. Faculdade de Engenharia da Universidade do Porto.
[32] Hate speech dataset annotated for Portuguese. https://rdm.inesctec.pt/id/dataset/cs-2017-008#, accessed on 2 January 2020.
[33] Alfina, I., Mulia, R., Fanany, M. I., Ekanata, Y. (2017). Hate speech detection in the Indonesian language: A dataset and preliminary study. In 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 233-238. https://doi.org/10.1109/ICACSIS.2017.8355039
[34] The Dataset for Hate Speech Detection in Indonesian. https://github.com/ialfina/id-hatespeech-detection, accessed on 2 January 2020.
[35] Automated Hate Speech Detection and the Problem of Offensive Language. https://github.com/t-davidson/hate-speech-and-offensive-language, accessed on 2 January 2020.
[36] Wiegand, M., Siegel, M., Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language. Proceedings of GermEval Conference on Natural Language Processing.
[37] GermEval-2018 data repository. https://github.com/uds-lsv/GermEval-2018-Data, accessed on 2 January 2020.
[38] Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., Wojatzki, M. (2017). Measuring the reliability of hate speech annotations: The case of the European refugee crisis. arXiv preprint arXiv:1701.08118. https://doi.org/10.17185/duepublico/42132
[39] Bohra, A., Vijay, D., Singh, V., Akhtar, S.S., Shrivastava, M. (2018). A dataset of Hindi-English code-mixed social media text for hate speech detection. In Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, pp. 36-41. https://doi.org/10.18653/v1/W18-

1105
[40] https://github.com/deepanshu1995/HateSpeech-Hindi-English-Code-Mixed-Social-Media-Text, accessed on 2 January 2020.
[41] https://developer.twitter.com/, accessed on 2 January 2020.
[42] https://pypi.org/project/beautifulsoup4/, accessed on 2 January 2020).
[43] https://pypi.org/project/PyArabic/, accessed on 2 January 2020.
[44] Zhang, X., Zhao, J.B., LeCun, Y. (2015). Character-level convolutional networks for text classification. Proceedings of Neural Information Processing Systems, pp. 649-657.
[45] https://keras.io/optimizers/, accessed on 2 January 2020.
[46] Hoffer, E., Hubara, I., Soudry, D. (2017). Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. Processing of Systems 30. MIT Press, pp. 1729-1739.