Comparative Analysis of Detecting Offensive Speech in Multiple Languages on Social Media

Rehan Ahmed*§, Prerak Tusharkumar Pradhan*, Mauli Amrishbhai Trivedi*,
Jimi Cao*, and Romina Mir*
Viterbi School of Engineering, University of Southern California, Los Angeles, California, USA
{rehanahm, preraktu, mauliamr, jimicao, rmir}@usc.edu
*equal contribution, §Corresponding author

Detecting offensive language in social media is a complex problem to resolve because this problem can not be resolved just by using word matching. This results in offensive language becoming a massive problem in online social platforms. Online communities, social media platforms, and technology companies have been investing heavily in ways to cope with offensive language to prevent abusive behavior in social media. In this project, we will conduct a study on various learning models on Offensive Speech on Twitter and discuss the possibility of using additional features and context data for improvements.

I. Introduction

With the increase of social media usage, the focus on improving the social space for the community members has been expanded. The proliferation of social media enables people to express their opinions widely online [1, 8, 9]. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Nowadays, many people on the internet publish content containing offensive and/or language on social media such as Facebook, Twitter, etc. In online comments, offensive and abusive language can lead to a host of different problems, including cyber-bullying that targets individuals (celebrity, politician, and product) and a group of people (specific country, age, and religion). Offensive language refers to any type of insult, vulgarity, or profanity that debases the target [5, 9]. Meanwhile, offensive language can be anything that causes offense, arousing a visceral reaction of disgust, anger, or hatred. It is vital to detect and analyze offensive language from online comments automatically. Detecting such language in social media is a complex problem to resolve because this problem can not be resolved just by using word matching as abuse and offensive language can take different forms. Thus, it is fast becoming a massive problem in online social platforms, something that people are still struggling to tackle to this day [11]. Online communities, social media platforms, and technology companies have been investing heavily in ways to deal with offensive language and to prevent offensive behavior on social media, but given the scale of the problem, where billions of users use social media and post every day, it is a very hard task that requires robust and scalable systems.

Most existing offensive language detection techniques that are in place rely on manual human intervention and are particularly ill-suited for large datasets. In this project, we will conduct a study on various learning models on Offensive Speech detection on Social Media Platforms and discuss the possibility of using additional features like a combination of multiple models to improve them. We will try to account for heterogeneity in this dataset by separately annotating both the comment as a whole and the individual sentences that comprise each comment and evaluate the best system for abuse and offensive language detection for large-scale datasets. We will also work on evaluating the performance of different abuse detection models in different languages.

II. Related Work

A number of studies have been done on modern techniques for conducting offensive language detection on large-scale data, collected from social media forums. We will be taking a look at the different systems proposed in such papers and their caveats and aim to evaluate

their performance on multilingual large datasets. **Lee et al. (2018)** looked at various learning models discussed the possibility of using additional features and context data for improvements, finally settling on bidirectional GRU networks trained on word-level features, with Latent Topic Clustering modules, as the most accurate model with an F-1 score of 0.805 [2].

Dai et al. (2020) made an offensive language detection system that combines multi-task learning with BERT-based models to learn noisy text in social media systems along with leveraging super-vision signals from other related tasks to achieve an F-1 score of 0.915 [3].

Nobata et al. (2016) experimented with several new features for this task: different syntactic features as well as different types of embedding features, and found them to be very powerful when combined with the standard NLP features. Char- acter n-grams alone fare very well in these noisy data sets. Our model also outperforms a deep learning-based model while avoiding the problem of having to retrain features on every iteration [4].

Park et al. (2018) found abusive language detection models tend to have a problem of being biased toward the identity words of a certain group of people because of imbalanced training datasets. The experiments with three bias mitigation methods: (1) debiased word embeddings, (2) gender swap data augmentation, and (3) fine-tuning with a larger corpus. These methods can effectively re- duce gender bias by 90-98% and can be extended to correct model bias in other scenarios [5].

Wiegand et al. (2019) studied the impact of data bias on abusive language detection and showed that this problem is closely related to how data have been sampled, states under a realistic evaluation classification performance is actually quite poor, particularly on implicit abuse [6].

Pitsilis et al. (2018) address the important problem of discerning hateful content in social media. They proposed a detection scheme that is an ensemble of Recurrent Neural Network (RNN) classifiers, and it incorporates various features associated with user-related information, such as the users' tendency towards racism or sexism [7].

Davidson et al (2017) examined racial basis in Twitter data. They found systematic racial bias in all datasets. These biases discriminate against the groups who are often the target of abuse that they try to identify [9].

Our work. A number of studies have been done on modern techniques for conducting offensive language detection on large-scale data, collected from social media forums. We will be taking a look at the different systems proposed in such papers and their caveats aiming to evaluate their performance on multilingual large datasets.

Our main contributions are i) an experimental evaluation of multiple different machine learning models on social media datasets, demonstrating the top performance achieved on the classification task, and Compare Model Performances to evaluate the best models for each language. ii) Implement a Voting-ensemble predictor to try to improve overall accuracy for each language

III. Methods

Machine Learning techniques. We are conducting different Machine Learning algorithms on 4 main Natural languages: English, Filipino, Chinese and Korean. We conducted both traditional machine learning models and Neural Network solutions (table-1), using both word and character features. The datasets had multiple labels, from sexism, abuse, to offensive, hate, and others, and suffered from imbalanced ratios. Keeping our goal of detecting all kinds of offensive language in mind, we decided to merge abuse, sexism, hate, offensive labels under the umbrella of "offensive" language, and all other terms as "non-offensive".

Traditional Algorithms	1. Naive Bayes 2. Logistic Regression 3. SVM 4. Random Forests 5. Gradient Boosting
Neural Networks Algorithms	1. CNN 2. RNN/BiLSTM

Table-1: Machine Learning algorithms that have been covered in this study

We took inspiration from **Lee et al. (2018)** for data preprocessing. For languages like Chinese and Korean that don't use Roman script, we used different tokenizing techniques [2]. For Chinese, we used a word segmentation technique using Jieba, which uses a combination of dynamic programming and HMM for Chinese text segmentation and has been proposed as having a good performance by **Zhang et al. (2019)**[12]. For Korean, we used a technique proposed by **Eunjeong L. ParkO et al.(2014)** [13].

We implemented traditional using both character and word-based features, converting text sequences are converted into Bag Of Words (BOW) representations and normalized with Term Frequency-Inverse Document Frequency (TF-IDF) values. Due to processing power constraints, we used n-grams ranging from 1 to 3, for both word and character, and 14000 features for word, and 53000 features for char models respectively.

For Convolutional Neural Networks (CNN), also implemented by **Lee et al. (2018)** [2], we have a word-level, char-level, and hybrid model, as proposed by Park and Fung (2017). The CNN models have 3 convolutional filters of different sizes [1,2,3] with ReLU activation and a Train Size:1674max-pooling layer. The CNN models use cross-entropy with softmax as their loss function and Adam (Kingma and Ba, 2014) as the optimizer. We ran each model for 10 epochs with a 0.01 learning rate and a batch size of 16.

To implement Recurrent Neural Network (RNN), we refer to the implementation provided in the paper by Georgios et al (2018). Models have an embedding layer, a single LSTM layer (Sepp, 1997), a fully connected layer, and an output layer. The dropout layer (Srivastava et al., 2014) is used after the LSTM layer and the fully connected layer for regularization. embeddings are learned throughout the training by embedding layers [11]. The final hidden state from the LSTM is passed to the fully connected layer. We also implement bidirectional LSTM with similar architecture. For character-based models, we use an LSTM layer with more hidden units than in word-based models to prevent under-fitting. Similar to CNN, RNN models use the Adam optimization algorithm (Kingma and Ba, 2014) and cross-entropy as loss function.

Embedding techniques. In the effort of conducting this classification, two types of word embedding are used as a representation for text where words that have the same meaning have a similar representation. The Global Vectors for Word Representation, or GloVe, the algorithm is an extension to the word2vec method for efficiently learning word vectors, developed by Pennington, et al.[1] at Stanford for English Natural Language. For Chinese and Korean language, we used pre-trained word vectors for 157 languages provided by Facebook Research's FastText platform [16], trained on Common Crawl and Wikipedia using fastText, enabling it to be applied to a range of downstream tasks, such as text classification in our case. For the Filipino language, we weren't able to find a suitable embedding dataset at the time.

Ensemble. We implemented an ensemble model using a voting-based ensemble technique that uses every model's prediction to make a better prediction as well as a probabilistic approach that tries to maximize the probability of the predicted class among all the model predictions.

Datasets. Annotated English tweets from **Davidson et al (2017)** have been collected using the Twitter API with tweets containing terms from the lexicon, resulting in a sample of tweets from 33,458 Twitter users. It contains 19,190 offensive tweets, 4,163 normal, and 1430 hateful tweets [9].

The Filipino dataset used in this study was a subset of the corpus 1,696,613 tweets crawled by Andrade et al. and posted from November 2015 to May 2016 during the campaign period for the Philippine presidential election. The original dataset for the Filipino natural language is primarily in Filipino, with the addition of some English words commonly used in Filipino vernacular. It contains 24,232 labeled tweets, with 12,979 labeled as normal, and 11253 as hate [15].

Chinese language dataset containing sexism-related posts collected from Sina Weibo, as well as the Chinese lexicon SexHateLex. There are 8,969 human-labeled comments in total, with 5876 normal and 3093 as sexist [17].

We gathered the Korean natural language dataset from an OpenSourceAgenda published by **Cho et al**. There are 8,367 human-labeled comments in total, with 3646 as normal, 2688 as offensive, and 2033 as hate [14].

To handle some severe imbalance between normal and other non-normal classes, and make it a fair comparison between languages, for all non-normal comments/tweets, like hate, offensive, sexist, we clubbed them into a new binary class called "offensive".

Measurement. We are going to use the weighted F-1 score as the performance measure. The F1 metric is pretty good at capturing recall and precision performance, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.

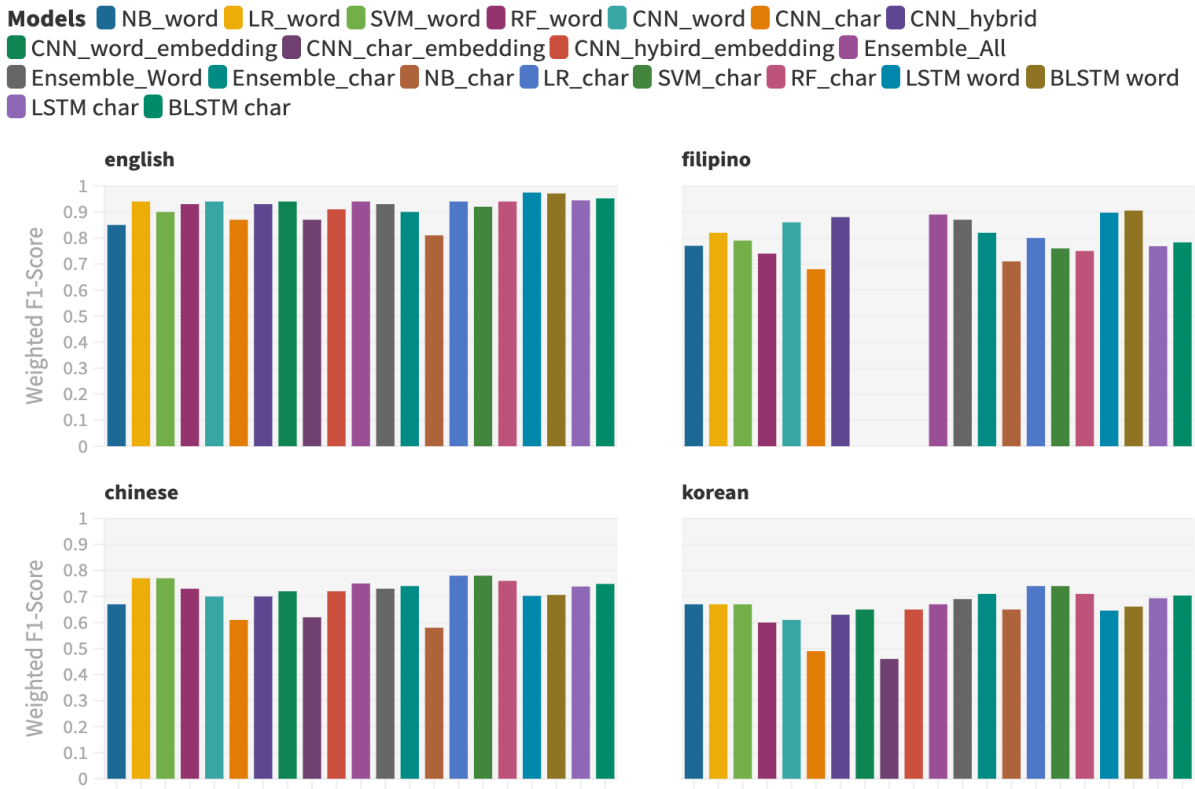

Fig 1. Weighted F1 performance of all machine learning models performed on 4 languages with different embeddings.

IV. Results & Discussion

Figure 1 shows the performance of different models on the four selected languages. In total 21 models with respect to two different embeddings of character-based and word-based have been performed. Overall we got better results for the English and Filipino than Chinese and Korean. Neural networks like word-based LSTM and BiLSTM models outpaced other techniques, getting the highest F1-scores of 0.97 and 0.91 for English and Filipino respectively, both using the Roman script and both word-driven languages. Meanwhile, traditional models, based on character features, such as SVM and Logistic Regression were equally good with weighted F1-scores of 0.78 at classifying Chinese, which uses symbols as words, unlike English. It is interesting to note that LSTM and BiLSTM performed similarly for character-based classification. We observed

similar results for Korean as well with char-based SVM and Logistic Regression besting other models with scores of 0.74. Compared to ensemble models we were not able to pass the score of our best model in each language independently. As mentioned above, since we were not able to find an embedding dataset for the Filipino language we are missing CNN models.

Challenges. One of the crucial technical challenges while addressing abusive language detection is its subjectivity and context-dependent nature. A message classified as harmless can be classified as abusive when taking account of previous threads of messages. 1) How the hell in this world someone help this man? 2) Family helped the killer in hiding. 3) I will come to your home and help your son go. 4) Your son did right by attempting suicide, coward. Sentence 1) can be categorized as abuse because of strong language, unless the context sentence 2) is accounted to classify it as non-abusive. Similarly, Sentence 3) feels harmless, as in a man who wants to help someone's child. But, its context sentence 4) helps to understand the grim intention of the user. Because of this, abusive language detection becomes extremely laborious and arduous, even for human annotators.

Datasets have errors and Ambiguities. Therefore, it is difficult to gather a large and reliable dataset. Comments contain sentences with transliteration and emojis. A word in a language can have multiple spellings after transliteration. Usage of slang is frequent in the comments and should be taken care of to get better results. Current research on abusive language detection focuses on the English language. Metadata-based explicit feedback such as the number of likes and reports are pivotal factors in the classification. There is not sufficient literature present on feedback incorporation, and it is outside the scope of our class.

V. Conclusions

In this Study special focus is given to investigating different models' performances in text classification across different natural languages. To the best of our knowledge, there has not been any previous study on exploring features related to the users' tendency to hate content that used different deep learning models in different languages.

The massive rise in user-generated content in social media services, with manual filtering not being scalable, highlights the need for automating the process of online hate-speech detection. In addition, with a large global presence, the need for the detection to work on multiple languages is ever increasing. With these in mind, our main purpose was to compare five different traditional machine learning models and two deep learning models on different languages. The comparison was done based on the weighted F1-score of each model on each language. We concluded that neural network word-based models worked best for English and Filipino and traditional character-based models worked best for Chinese and Korean.

Additional information: This study is part of the graduate course CSCI-554 (Natural Language processing - graduate level) led by head lecturer Dr. Mohammad Rostami aiming to study the detection of offensive language on social media platforms by using NLP-related algorithms as our final project.

Supplementary GitHub information is available for this paper at: https://github.com/reallyrehan/comparative-analysis-multilingual-offensive-language

The YouTube video can be found directly at https://youtu.be/9YRKuz0-flM

Resources:

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[2] Younghun Lee, Seunghyun Yoon, Kyomin Jung. 2013. Comparative Studies of Detecting Abusive Language on Twitter

[3] Wenliang Dai, Tiezheng Yu, Zihan Liu, Pascale Fung. 2020. Kungfupanda at SemEval-2020

Task 12: BERT-Based Multi-Task Learning for Offensive Language Detection

[4] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, Yi Chang. 2016. Abusive Language Detection in Online User Content

[5] Ji Ho Park, Jamin Shin, Pascale Fung. 2018. Reducing Gender Bias in Abusive Language Detection

[6] Michael Wiegand, Josef Ruppenhofer, Thomas Kleinbauer. 2019. Detection of Abusive Language: the Problem of Biased Datasets

[7] Georgios K. Pitsilis, Heri Ramampiaro, and HelgeLangseth. 2018. Detecting Offensive Language in Tweets Using Deep Learning. Thomas Davidson, Debasmita Bhattacharya, Ingmar

[8] Weber 2019. Racial Bias in Hate Speech and Abusive Language Detection Datasets.

[9] Davidson T, Warmsley D, Macy MW, Weber I. Automated Hate Speech Detection and the Problem of Offensive Language. CoRR. 2017

[10] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1 (January 2014), 1929–1958.

[12] Zhang, Xianwei & Wu, Peng & Cai, Jiuming & Wang, Kun. (2019). A Contrastive Study of Chinese Text Segmentation Tools in Marketing Notification Texts. Journal of Physics: Conference Series. 1302. 022010. 10.1088/1742-6596/1302/2/022010.

[13] Eunjeong L. ParkO, Sungzoon Cho (2014). KoNLPy: Korean natural language processing in Python. Seoul National University, Industrial Engineering Department.

[14] Cho, Won Ik and Lee, Junbum, (2020). {K}orean Corpus of Online News Comments for Toxic Speech Detection - Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media.

[15] Neil Vicente Cabasag, Vicente Raphael Chan, Sean Christian Lim, Mark Edward Gonzales, and Charibeth Cheng (2019). Hate speech in Philippine election-related tweets: Automatic detection and classification using natural language processing.

[16]https://fasttext.cc/docs/en/crawl-vectors.html

[17] Aiqi Jiang and Xiaohan Yang and Yang Liu and Arkaitz Zubiaga (2021), SWSR: A Chinese Dataset and Lexicon for Online Sexism Detection