Language Agnostic Model - Detecting Islamophobic Content on Social Media

Heena Khan
Middle Tennessee State University
Murfreesboro, TN, USA
kheena05@gmail.com

Joshua L. Phillips
Middle Tennessee State University
Murfreesboro, TN, USA
Joshua.Phillips@mtsu.edu

ABSTRACT

Social media platforms can struggle to enforce rules preventing online abuse and hate speech due to the large amount of content that must be manually reviewed. Machine learning approaches have been proposed in the literature as a way to automate much of these labors, but social content in multiple languages further complicates this issue. Past work has focused on first building word embeddings in the target language which limits the application of such embeddings to other languages. We use the Google Neural Machine Translator (NMT) to identify and translate Non-English text to English to make the system language agnostic. We can therefore use already available pre-trained word embeddings, instead of training our models and word embeddings in different languages. We have experimented with different word-embedding and classifier pairs as we aimed to assess whether translated English data gives us accuracy comparable to an untranslated English dataset. Our best performing model, SVM with TF-IDF, gave us a 10-fold accuracy of 95.56 percent followed by the BERT model with a 10-fold accuracy of 94.66 percent on the translated data. This accuracy is close to the accuracy of the untranslated English dataset and far better than the accuracy of the untranslated Hindi dataset.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Applied computing** → **Sociology**.

KEYWORDS

Natural Language Processing, Sentiment Analysis, Text Classification, Islamophobia, Social Media, Dataset

ACM Reference Format:
Heena Khan and Joshua L. Phillips. 2021. Language Agnostic Model - Detecting Islamophobic Content on Social Media. In *2021 ACM Southeast Conference (ACMSE 2021), April 15–17, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3409334.3452077

1 INTRODUCTION

Most social media platforms have certain established rules to prevent online abuse and hate speech. Enforcing these rules, however, requires copious manual labor to review every report. Automatic tools and approaches can accelerate the reviewing process [11]. Researchers in the field of natural language processing (NLP) have come up with different algorithms and techniques to automate hate speech and abuse detection on social media. These tools are now used by many social media platforms to efficiently eliminate such content.

One major challenge faced in the application of these models is identifying such content posted in languages and/or dialects with which the models have not been explicitly trained. For the most widely-used languages, specialized models have been trained to help with this task, but such models are commonly not available for less-common languages or dialects. It is not currently known whether a model specifically developed for inappropriate content in a target language is practically necessary for ML models to achieve an adequate level of performance. Since training requires extensive manual curation of adequate training/validation content as well as subsequent model training and tuning, it would be preferable if this step could be avoided.

Even though pretrained word embedding and/or translation models specifically aimed at inappropriate content may not exist, a general word embedding and/or translation model is more often available. However, the use of general word embedding and/or translation models instead of specialized models has garnered little attention in the literature. While one might anticipate some loss in categorization accuracy from using a general model, but the potential practical savings in terms of manual time and effort may outweigh those costs. This is especially true for languages and dialects where sufficient data may not currently exist for the development of a specialized model. We consider such an approach to be *language agnostic* since it may be applied to any language for which only a general word embedding and/or translation model currently exists. In this paper, we explore the effectiveness of the language agnostic approach by exploring data sets in English and Hindi using both general translation and specific word embedding approaches in order to better understand the limitations and advantages of the language agnostic approach.

2 BACKGROUND

Recent years have seen an increasing number of studies on hate speech detection for different targeted groups concerning gender, race, and communities [11]. Researchers have used various classification methodologies to identify social abuse. Davidson et al. [3] used a traditional feature-based classification model that incorporates distributional term frequency-inverse document frequency (TF-IDF) and other linguistic features using Support Vector Machines (SVM). They used three labels: hate speech, offensive, and

ACMSE 2021, April 15–17, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8068-3/21/04…$15.00
https://doi.org/10.1145/3409334.3452077

neither hate speech nor offensive. Waseem et al. [16] worked on their dataset from twitter consisting of 16,914 tweets labeled as racist, sexist, or neither. For classification they used the traditional n-gram based method with Logistic Regression. Mulki et al. [12] introduced a dataset L-HSAB combining 5,846 Syrian/Lebanese political tweets labeled as normal, abusive or hate . They used traditional n-gram BOW and TF vectorization methods with Naive Bayes (NB) and SVM classifiers. Most of the time, n-gram vectorization with machine learning classifiers performs well with text categorization and sometimes they even outperform Neural Networks, but they are highly domain-specific and may not work well with unseen out-of-context data. They can also suffer when negative words are used positively. For example, "Calling Muslims terrorist is a stereotype", is a sentence that can be misunderstood by such classifiers as Hateful/Islamophobic as it contains negative words [11]. The traditional n-gram method can perform equally well with multilingual data but only when trained in the same language.

De GilBert et al. [4] introduced their data consisting of posts from a white supremacist forum labeled as categories: Hate, No-Hate, Relation, or Skip. They used three classifiers: SVM with Bag Of Words (BOW), Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) with randomly initialized word embeddings. Since the authors used randomly initialized word embeddings, the word embeddings would contain most of the words from the dataset, because the embedding layer is trained using the words in the dataset, but training word embeddings is a time-resource consuming task.

A 2018 Workshop on Trolling, Aggression, and Cyberbully (TRAC) hosted a shared task force focused on detecting aggressive text in both English and Hindi [10]. Their data is labeled as overtly aggressive, covertly aggressive, or non-aggressive. The teams used different methodologies, from simple machine learning classifiers to deep learning neural networks. It was observed that classifiers like SVM, random forest and logistic regression performed as good as and sometimes better than neural networks. Some of the teams using neural networks used pre-trained word embeddings with both English and Hindi data. There are chances that out-of-vocabulary words occur frequently when pre-trained word embeddings are used with non English data. Word embeddings like FastText can embed out-of-vocabulary words by looking at subword information (character n-grams), but the model must be trained on the out-of-vocabulary word.

Darwish et al. [2] researched people's stance on Islam and Muslims before and after the "Nov '15 Paris Attack". Their data was labeled as Defending, Attacking, and Neutral. To identify Islamophobia on twitter Vidgen et al. [15] introduced 'Detecting weak and strong Islamophobic hate speech on social media'. Their dataset is labeled as Strong Islamophobia, Weak Islamophobia, and Non-Islamophobia. They created six models using simple machine learning classifiers as well as a deep learning neural network. They tested the classifiers with their newly trained GloVe (GloVe DSWI) as well as a pre-trained GloVe. Their results were promising but their data is private and hence cannot be reproduced.

Saha et al. [13] addressed growing hate crimes in India and the importance of studying hate speech in the Indian language. They used the HASOC 2019 public dataset with three languages Hindi, German and English. They have used the Gradient Boosting model, along with mBert and LASER embeddings, to make the system language agnostic. Their model performed well with Hindi data but did not perform equally well with English and German which they report is due to data imbalance issues.

3 METHODS

Taking inspiration from prior research, we focus here on detection of Islamophobic content on social media. Previous work on hate speech detection, and detection of Islamophobia in particular, demonstrates the challenges of – but also the potential for – creating a classification system that can work for multiple languages. To our knowledge, no previous research has focused specifically on Islamophobia for multilingual data and so there is a need to generate a multilingual dataset for the classification of Islamophobia which will hopefully be of future benefit to the research community.

To make the system *language-agnostic* we use the Google Neural Machine Translator (NMT) to identify and translate Non-English text to English. Our dataset is classified into three categories; Islamophobic, About Islam but not Islamophobic and, Neither about Islam nor Islamophobic. The dataset consists of two languages: English and Hindi. To save training time and resources we aim to use already existing pre-trained word embeddings for both the Hindi and English language. This choice is motivated by the fact that general word embeddings are not trained especially on Islamophobic content but are still more readily available and abundant. We are using the newly trained embeddings GloVe DSWI from the paper "Detecting weak and strong Islamophobic hate speech on social media" for testing with our data. We also wanted to reproduce their results, but since their dataset is private, we were unable to do so. As most of the word embeddings are only trained on English data and do not contain vocabulary for non-English data, we will be translating Non-English text to English before word vectorization.

Word embeddings like Word2Vec, GloVe and Bert are pretrained in English text. Training these embeddings for different languages is a time and resource consuming task. We introduced a simple method for non-English text classification using existing pretrained word embeddings models. Rather than training word embeddings on multilingual data, we add the Google Neural Network Machine translator (NMT) to our model using Google API. By default, when you make a translation request to the Cloud Translation API, the text is translated using the NMT model. If the NMT model is not supported for the requested language translation pair, then the Phrase-Based Machine Translation (PBMT) model is used to translate Non-English text to English before passing it to the word embeddings [5].

3.1 Experiment

We have created several models using different word embeddings with different classifiers, but the main architecture is explained in Figure 1. Our translator remains the same for all the models. The model architecture represents different layers in our model. Input is provided in the form of data frames containing labeled tweets. Our NMT translator will identify the text language and translate it into English language. This translated data is then pre-processed and tokenized.

Our data pre-processing involves converting all text data to lowercase, removing Stopwords (for Hindi text we have used the Hindi

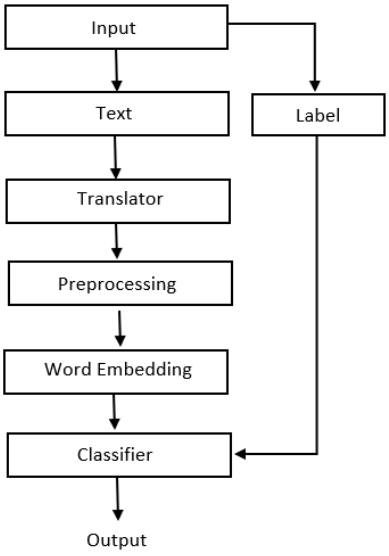

Figure 1: The Model Architecture

stopwords list), word lemmatization, removing hyperlinks, removing improper full stop and sentence continuation, and word tokenization. Since we have a slightly imbalanced dataset, our model could produce sub-optimal results [1]. So we took care of this imbalance issue by repeated sampling. We took the class with largest number of tweets and randomly duplicated the tweets from other two classes to provide examples from all classes with equal frequency.

The vocabulary size for untranslated English data after preprocessing is 17861 unigrams, for translated English data it is 16035, and for Hindi data, it is 20262 unigrams. The distribution of tweet-length for untranslated English dataset after data pre-processing is 14 words/tweet, for translated English dataset it is 12 words/tweet and for the Hindi dataset, it is 13 words/tweet.

We perform experiment using our model architecture with 3 different word embeddings models namely Word2Vec, GloVe and Bert Model. We are also using traditional n-gram method known as TF-IDF and BOW. We use different classifiers with each word embedding. Traditional n-grams embeddings are tested using Machine learning models SVM and RFM (Random Forest Method). GloVe and Word2Vec word embeddings are implemented with deep neural network models like CNN and LSTM.we are using the Bert embedding within the Bert and mBert Model. We are also using the newly trained GloVe (GloVe DSWI) word embeddings provided by the author Vidgen et al. [15]. To estimate the potential of all our models on the new data and for fine tuning the hyperparameters we used 10-fold cross validation as our metric. We also calculated the f1 score. For the train-test data split, 90 percent of the data were allocated to the training set and 10 percent were allocated to the test set.

3.1.1 hyperparameters. Our LSTM model has 3 layers: the embedding layer, the LSTM layer, and the softmax layer. We fined tuned the hyperparameters; embedding dimension to 300 and neuron count to 256 neurons (LSTM block) in the hidden layer. The CNN model has 4 layers: the embedding layer, the convolutional layer, the max pooling layer, and the softmax layer [8]. We fine-tuned the embedding dimension to 300, the neurons count in the hidden layer to 512 neurons, and the kernal (window) size to 2,3,4,5. In Bert and mBert, to tokenize our text into tokens that correspond to Bert's vocabulary we use Bert tokenizer. We fine-tuned the pre-trained Bert model using our inputs. We also flatten the output and add Dropout with two Fully-Connected layers. The last layer has a softmax activation function [14]. For the SVM we set the kernel to linear, max iteration to 10000, and Tolerance for stopping criteria to 1e-5. In the RFM model we used 200 trees with a maximum depth of 20 nodes.

3.2 Dataset

We collected tweets from the Twitter social media platform for constructing our dataset. We did not focus on a particular country or region for our English dataset. The data for the Hindi language comes mostly from the Indian subcontinent, but we did not focus on any particular region within India for our data. We extracted our data using the lexicon from a crowd-sourced online database for hate speech, called Hatebase [6], as well as some trending Islamophobic hashtags on Twitter. Data was retrieved in the span of 3 to 4 months from around January 2020 to August 2020. The dataset is heterogeneous with a diverse range of user data as we did not focus our search targeting certain user's accounts. After retrieving our data we removed all the metadata related to user identities like tweet-Id, user-Id, user-Geo-location, etc., to make sure that the data does not contain the identity of the user who posted it.

3.3 Data Annotation

Our data consist of 8438 English tweets and 8790 Hindi tweets. Our English-Hindi data is annotated by three annotators proficient in English and Hindi language. To ensure anonymity and to prevent bias we provided our annotators with raw tweets without any user-id or tweet-id attached to them (see Dataset above). The annotation was done based on a set of guidelines provided along with a few examples for each class. In the case of annotators' disagreement, tweets were assigned to the class with the majority vote. Our dataset is classified into three categories; Islamophobic, About Islam but not Islamophobic and, Neither about Islam nor Islamophobic. Table 1 represents the tweet counts for each label in both of the datasets.

Table 1: Total Count of Tweets

Label	English Dataset	Hindi Dataset
2 - Islamophobic	2485	3373
1 - About Islam but not Islamophobic	2398	2172
0 - Not about Islam nor Islamophobic	3555	3245

The code and dataset developed during this research are available online in a GitHub repository: https://github.com/hk-mtsu/Language-agnostic-model-Detecting-Islamophobic-content-on-Social-Media.git.

4 RESULTS

All the models that we have created are trained and tested on the English language. All Non-English text is first identified and then translated to English by the Google Neural Machine Translation (GNMT) model. Table 2 represents 10-fold mean accuracy and F1 score obtained using different classifier and word embeddings pairs on English data and translated English data (from Hindi). We also evaluated our model performance on the original Hindi data for comparison.

Table 2: 10-Fold Cross-validation Accuracy in Percent and F1 Score

	English		Translated English		Hindi	
	10-Fold	F1 Score	10-Fold	F1 Score	10-Fold	F1 Score
BERT	96.21	96	94.66	94	94.17	93
CNN-Word2vec	96.58	98	93.50	93	92.10	90
CNN-GloVe	95.96	97	87.37	86	91.69	90
CNN-DSWI	93.07	90	88.53	72	89.74	83
LSTM-Word2vec	65.42	66	91.76	90	87.78	85
LSTM-GloVe	93.67	94	92.13	92	43.28	43
LSTM-DSWI	33.31	31	37.61	38	32.61	30
SVM-TFIDF	97.18	97	95.56	96	89.83	90
SVM-BOW	97.44	97	94.44	95	87.49	88
RFM-TFIDF	91.90	95	84.47	86	80.86	81
RFM-BOW	94.20	96	87.79	89	81.98	83
mBERT	95.55	96	93.72	94	95.15	94

The BERT model being trained on English data, performed well with the English dataset. The mBert model being trained on multilingual data, performed well with the English and Hindi datasets. On comparing Bert and mBert model, Bert model performed better on English and translated English data and mBERT performed better with Hindi data, but the difference in accuracy is very small.

The accuracy for Word2Vec word embeddings for both LSTM and CNN model remained quite similar for both translated and original Hindi data, but a considerable difference in accuracy was observed for both LSTM and CNN models while using standard GloVe word embeddings. The GloVe DSWI worked well with the CNN model with accuracy of 88.53% on translated English data, but when used with LSTM it showed no improvement.

SVM gave us the highest accuracy on English and translated English data in spite of being such a simple model because the n-grams word embeddings are domain specific having no pre-training on external data. SVM performance on Hindi dataset was also not bad. The highest accuracy on English data is by SVM–BOW 97.44% and the highest accuracy on translated English data is by SVM–TF-IDF 95.56 %.

Table 3 represents the top 5 models that performed well with non-English data following our model architecture (translated English data).

Table 3: Top 5 Accuracy's and F1 Score

	SVM-TFIDF	BERT	SVM-BOW	mBERT	CNN-Word2vec
10-Fold	95.56	94.66	94.44	93.72	93.50
F1 Score	96	94	95	94	93

5 DISCUSSION

We have made several observations while experimenting with different models and word embeddings. Since we are using Google API to translate non-English sentences, we noticed that some of the sentences remained unchanged. So the efficacy of our model depends on the NMT model. We also observed that TF-IDF and BOW performed better with translated English data (from Hindi data). But the accuracy for untranslated Hindi data was still relatively good. This is quite possibly because TF-IDF and BOW evaluate how relevant a word is to a document and assigns the vector accordingly. We observed that the accuracy of the SVM and the BERT models is very similar. However, SVM gave us the highest accuracy on English and translated English data. The mBERT model gave us the highest accuracy of 95.15 percent on the untranslated Hindi dataset. Since the mBERT model is trained on multilingual data it performed exceptionally well with untranslated Hindi data compared to all other models, but, once translated, our results for the translated English dataset and the untranslated English dataset were very close using SVM, LSTM, RFM, BERT, and CNN models. While using LSTM with Word2Vec our accuracy for translated English dataset was better than it was for the English dataset. Some of our models like CNN, BERT, and SVM also performed well with untranslated Hindi data, but the performance of translated data was consistently good with all the models.

6 CONCLUSION

We aimed to create language-agnostic classifiers for detecting hate speech or abusive content on social media which use only general pretrained word-embeddings for multilingual data. In our paper, we provided a fairly simple solution to the multilingual data classification problem by translating non-English text to English. The results we obtained from our model demonstrated it to be a viable solution. The introduction of a public dataset can benefit future research in this area. Hate detection is an ongoing area of research which will need to be constantly revisited as the nature of online abuse changes [15]. In the future, we would like to test other lightweight text classification models like Projection Attention Neural Network (PRADO) and pQRNN. The PRADO model was introduced by Google AI in Nov 2019, and it showed promising results when compared to CNN and LSTM with much fewer parameters [9]. pQRNN is the more recent model introduced by Google AI in Sept 2020. The pQRNN model is an extension of the PRADO model. The results of the pQRNN model have been quite close to the state-of-art BERT model [7]. These lightweight models do not require any external word embedding, so we would like to test their performance on both non-English text and their translations using our approach. We also plan to include other languages in our dataset to support more studies on Islamophobia as well as to identify other kinds of hate speech and abusive content.

REFERENCES

[1] R. Batuwita and V. Palade. 2013. Class Imbalance Learning Methods for Support Vector Machines. (2013).
[2] K. Darwish, W. Magdy, A. Rahimi, T. Baldwin, and N. Abokhodair. 2018. Predicting Online Islamophopic Behavior After #parisattacks. *The Journal of Web Science* 4 (2018).
[3] T. Davidson, D. Warmsley, M. Macy, and I. Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Eleventh International AAAI*

Conference on Web and Social Media. Québec, Canada.

[4] O. de Gibert, N. Pérez, A.-G. Pablos, and M. Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. *arXiv preprint arXiv:1809.04444* (2018).

[5] Google Cloud [n.d.]. *Google Translation*. https://cloud.google.com/translate/docs/basic/translating-text.

[6] Hatebase [n.d.]. *Hatebase*. https://hatebase.org/.

[7] P. Kaliamoorthi. 2020. *Google AI - Advancing NLP with Efficient Projection based Model Architectures*. https://ai.googleblog.com/2020/09/advancing-nlp-with-efficient-projection.html.

[8] Y. Kim. 2014. Convolutional Neural Networks for Sentence Classification. *arXiv preprint arXiv:1408.5882* (2014).

[9] K. Krishnamoorthi, S. Ravi, and Z. Kozareva. 2019. PRADO: Projection Attention Networks for Document Classification On-device. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 5013–5024.

[10] R. Kumar, A. Ojha, S. Malmasi, and M. Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, 1–11.

[11] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder. 2019. Hate Speech Detection: Challenges and Solutions. *Plos One* 14, 8 (2019), e0221152.

[12] H. Mulki, H. Haddad, C. Ali, and H. Alshabani. 2019. L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language. In *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy, 111–118.

[13] P. Saha, B. Mathew, P. Goyal, and A. Mukherjee. 2019. HateMonitors: Language Agnostic Abuse Detection in Social Media. *arXiv preprint arXiv:1909.12642* (2019).

[14] V. Valkov. [n.d.]. *Intent Recognition with BERT using Keras and TensorFlow 2*. https://www.kdnuggets.com/2020/02/intent-recognition-bert-keras-tensorflow.html.

[15] B. Vidgen and T. Yasseri. 2020. Detecting Weak and Strong Islamophobic Hate Speech on Social Media. *Journal of Information Technology & Politics* 17, 1 (2020), 66–78.

[16] Z. Waseem and D. Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*. San Diego, California, 88–93.