Language Agnostic Model: Detecting Islamophobic Content on Social Media

By

Heena Khan

A thesis submitted in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

Middle Tennessee State University

May 2021

Thesis Committee:

Dr. Joshua Phillips

Dr. Cen Li

Dr. Sal Barbosa

ACKNOWLEDGEMENTS

I would like to express my gratitude to my primary supervisor, Dr. Joshua Phillips, who guided me throughout this project. I would also like to thank my friends Luis Chunga, who supported me and offered deep insight into the study and Rituraj Pandey, who helped me during the process of retrieving data from Twitter. I would like to thank my annotators for their effort and time.

ABSTRACT

Islamophobia or anti-Muslim racism is one dominant yet neglected form of racism in our current day. The last few years have seen a tremendous increase in Islamophobic hate speech on social media throughout the world. This kind of hate speech promotes violence and discrimination against the Muslim community. Despite an abundance of literature on hate speech detection on social media, there are very few papers on Islamophobia detection. To encourage more studies on identifying online Islamophobia we are introducing the first public dataset for the classification of Islamophobic content on social media. Past work has focused on first building word embeddings in the target language which limits its application to new 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 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.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I.

INTRODUCTION

Islamophobia constitutes a major racist discourse today [11]. According to an online report submitted by Twist-Islamophobia, a large percentage of total online abuse is Islamophobic [15][23]. In 2018-19, Islamophobic content was the biggest source of hate speech on Facebook in India, accounting for 37 percent of the total content, reported by Equality Labs [23]. In 2020, Equality Labs reported on how Muslims are made an easy scapegoat for the corona-virus in India [28]. In 2019, NPR reported on growing Islamophobia in the U.S [32], a similar report was published by Aljazeera in 2018 [1]. Reuters and BBC in 2018 reported on China's atrocities on the Uighur Muslim community [31][47]. Hate crimes against Muslims are at an all-time high in the American [36], European [26] [39], Asian [41], and Indian Subcontinents [2][14][17]. Study on Islamophobia has gained some traction in the western research communities with articles like 'Unwanted Identities: The 'Religion Line' and Global Islamophobia' by Hafez [11]. 'Detecting weak and strong Islamophobic hate speech on social media' by Vidgen and Yasseri [45]. 'Islamophobia, White Supremacy, and the Far-Right' by Huzaifa [40]. 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 [25]. Researchers in the field of Natural Language Processing 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 [25]. Similarly, a robust computational tool can help identify and eliminate Islamophobic content on social media.

Islamophobia

Building on previous academic work, the term Islamophobia is used 'to refer to an irrational distrust, fear or rejection of the Muslim religion and those who are (perceived as)

Muslim' [19]. In this paper, both Muslims and Islam are included within our definition as targets of Islamophobia. Any negative reference to Islam, Muslims, their place of worship, festivals, and practices means that we are potentially looking at Islamophobia [45]. Anything overtly or covertly expressing indiscriminate negativity against Islam or Muslims is marked as Islamophobic, because even subtle racism/Islamophobia can impact the community equally [8][12].

CHAPTER II.

BACKGROUND

Recent years have seen an increasing number of studies on hate speech detection for different targeted groups concerning gender, race, and communities [25]. Researchers have used various classification methodologies to identify social abuse. **Davidson et al.** used a traditional feature-based classification model that incorporates distributional TF-IDF and other linguistic features using Support Vector Machines (SVM). They used three labels: hate speech, offensive, and neither hate speech nor offensive [6]. **Waseem et al.** worked on their dataset from twitter consisting of 16,914 tweets labeled as racist, sexist, or neither [46]. For classification they used the traditional n-gram based method with Logistic Regression. **Mulki et al.** introduced a dataset L-HSAB combining 5,846 Syrian/Lebanese political tweets labeled as normal, abusive or hate [30]. 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 as hateful/Islamophobic as it contains negative words [25]. The traditional n-gram method can perform equally well with multilingual data but only when trained in the same language.

De GilBerT et al. introduced their data consisting of posts from a white supremacist forum labeled as categories: Hate, No-Hate, Relation, or Skip. They used three classifiers: Support Vector Machine (SVM) with Bag Of Words (BOW), and Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) with randomly initialized word embeddings [7]. 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 [22]. 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 [16] 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.

To identify Islamophobia on twitter **Vidgen et al.** introduced 'Detecting weak and strong Islamophobic hate speech on social media' [45]. 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. addressed growing hate crimes in India and the importance of studying hate speech in the Indian language [38]. They used the HASOC 2019 public dataset with three languages Hindi, German and English. They 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. The mBERT is a multilingual BERT model, being trained in 104 languages. It would have been interesting to see the performance

on the mBERT model alone on the same data.

Taking inspiration from these works, we focused on Islamophobic content on social media. Previous work on Hate Speech and Islamophobia detection demonstrates the challenges of – but also potential for – creating a classification system that can work for multiple languages. **To our best knowledge, no previous research has focused specifically on Islamophobia for multilingual data.** There is also a need for a public dataset for the classification of Islamophobia, which is currently lacking and may benefit the research community. **We are introducing the first public dataset on the classification of Islamophobia. So that more researchers can now work on building tools to better identify online Islamophobia**. 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, Not 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 and existing GloVe embeddings by Stanford [34]. 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 Embedding

To train a machine learning model, text data needs to be converted into a vector representation. There are different ways of converting words to vectors, the traditional n-grams

methods are Bag of words (BOW), term frequency–inverse document frequency (TF-IDF) [33] and there are more advance methods like word embedding [27] [34]. In 2013, Google created Word2vec, a word embedding toolkit which can train vector space models faster than previous traditional approaches [27]. The generated vectors are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems [27]. GloVe is another word embedding technique introduced by Pennington et al. [34]. It is an extension to the Word2vec method for efficiently learning word vectors. Most modern word embedding techniques rely on a neural network architecture instead of the more traditional n-gram methods. In 2017, a new type of deep contextualized word representation called ELMo was introduced [35]. The ELMo offers some advantages over models like Word2vec and GloVe, because while each word has a fixed representation under Word2vec and GloVe, regardless of the context within which the word appears, ELMo produces word representations that are dynamically informed by the words around them [35]. In 2018, BERT (Bidirectional Encoder Representations for Transformers) model was introduced by researchers at Google AI Language. BERT uses ELMo like contextualized word embedding [9]. These embeddings are trained on large datasets, saved, and then used for solving other tasks. This makes pretrained word embeddings a form of transfer learning. We run model experiments with traditional n-grams methods, Word2vec and GloVe word embedding, and advance BERT and m-BERT contextual word embedding models for comparison.

Classifiers

Human communication isn't just syntax and semantics. It is much more complex as it involves emotions. The choice of words, writing style, and sentence structure play a large part in determining the sentiment behind a written message [10][29]. Earlier approaches to sentiment analysis were based on tokenizing the written sentences and trying to determine out the sentiment based on rules of grammar using machine learning classifiers like SVM, RFM,

NB, etc [29]. With the rise of social media platforms, most of the content we find online is grammatically incorrect and contains slang and abbreviations that keep changing with time. With the introduction of Deep Neural Network in Natural Language Processing (NLP) more sophisticated models were developed to overcome these issues. Using a technology called sequence-to-sequence learning, programmers could solve some of the most complex NLP problems of the time [42]. The sequence-to-sequence architectures based on Recurrent Neural Networks were widely used and were particularly useful if the prediction has to be at word-level, for instance, Named-Entity Recognition (NER) or Part of Speech (POS) tagging [3]. The RNN can easily map sequences to sequences whenever the alignment between the inputs the outputs is known ahead of time. However, it is not clear how to apply an RNN to problems whose input and the output sequences have different lengths with complicated and non-monotonic relationships [42]. Researchers also started to apply CNN to problems in NLP and obtained interesting results [20]. Since CNNs, unlike RNNs, can output only fixed sized vectors, the natural fit for them seem to be in the classification tasks such as Sentiment Analysis, Spam Detection or Topic Categorization [49]. Long Short Term Memory (LSTM) was also a big development over RNNs, Although RNNs can learn long-range dependencies in theory, in practice they're better at short distance dependencies. LSTMs help solve this problem by understanding context along with recent dependencies. Hence, LSTM are a special kind of RNNs where contextual understanding can be useful [37]. Bi-directional LSTMs were also famously used in NLP. As the name suggests, these networks are bidirectional, that is, it has access to both past and future input features for a given time. LSTMs and Bi-directional LSTMs were the most progressive model for NLP until the end of 2018. While using RNN and LSTM models, it was harder for the context vector to capture all the information contained in a sentence for long sentences with complicated dependencies between words, due to their sequential order of word processing. To address this bottleneck issue, researchers created a technique for paying *attention* to

specific words. Attention was a revolutionary idea in sequence-to-sequence systems such as translation models. The release of the Transformer paper and code, and the results it achieved on tasks such as machine translation, started to make some in the field think of them as a replacement to LSTMs. This was compounded by the fact that Transformers deal with long-term dependencies better than LSTMs [48].

On December 2018, Google introduced a transformer model based on the attention technique, ELMO contextual word-embedding model and ULMFIT. This new transfer learning technique called BERT (Bidirectional Encoder Representations for Transformers) made big waves in the NLP research space [9]. Later, Google introduced their pre-trained multilingual BERT (mBERT) model and by the end of 2019 Facebook improved on BERT by introducing RoBERTa [24]. We will be training and testing our data using the machine learning SVM and RFM method, as well as improved deep learning models like CNN and LSTM and the advance transformers like BERT and mBERT.

CHAPTER III.

DATASET

Given the spread of Islamophobia across the world we did not focus on a particular country or region for our English dataset, but to test our model on multilingual data we chose Hindi as our non-English language. Data for the Hindi language comes mostly from the Indian subcontinent. 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 [13]. We also used some trending Islamophobic hashtags like #fuckIslam, #Jihadi, #Coronajihad, #Tablighijamat, #TablighiJamaatVirus on Twitter to retrieve our data. Our data was retrieved in the span of 3 to 4 months from around January 2020 to August 2020. Our 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 the user identity like tweet-Id, user-Id, user-Geo-location, etc., to make sure that the data we are sharing with our annotators and the public does not contain the identity of the user who posted it.

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 it. The annotation was done based on a set of provided guidelines 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 count of tweet for each label in both the dataset.

Table 1: Total count of tweets for each label

Label	English Dataset	Hindi Dataset
2 - Islamophobic	2485	3373
1 - About Islam Not Islamophobic	2398	2172
0 - Neither Islamophobic Nor About Islam	3555	3245

CHAPTER IV.

METHODS

We are introducing a practical method for Non-English text classification using existing pretrained word embedding models. Word embeddings like Word2vec, GloVe and BERT are pretrained on the English language. Training these embedding for different languages is a time and resource consuming task. So rather than training a word embedding on multilingual data, We add the Google Neural Network Machine Translator (NMT) to our model using the Google API. By default, when you make a translation request to the Cloud Translation API, your 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 embedding [5].

Our main focus is to use existing pre-trained word embeddings. We experiment our model with 3 different word embedding models namely Word2vec, GloVe and BERT. We are also going to use traditional n-gram method known as TF-IDF and BOW.

We are using different classifiers with each word embedding. Traditional n-grams embeddings are tested using Machine learning models SVM and RFM. GloVe and Word2vec word embedding are implemented with Deep Learning models like CNN and LSTM. We are using BERT Embedding within our BERT and mBERT model.

Data Pre-processing

Preprocessing is one of the key components in a typical text classification framework [43]. Our data pre-processing involves lower-casing all text data, removing stopwords (for the Hindi text we have used the Hindi stopwords list), word lemmatization, removing hyperlinks, removing improper full stop and sentence continuation, and word tokenization.

Data Balancing

Since we have a slightly imbalanced dataset, our model could produce sub-optimal results [4]. So 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. In our results we will be providing our models performance with and without data balancing.

Table 2 represents count of tweet for each label in both the dataset after data balancing.

Table 2: Total count of tweets for each label after data balancing

Label	Balanced English Dataset	Balanced Hindi Dataset
2 - Islamophobic	3554	3244
1 - About Islam Not Islamophobic	3554	3244
0 - Neither Islamophobic Nor About Islam	3555	3245

Model Architecture

We have created several models using different word embedding with different classifiers, but the main architecture is explained in Figure 1. Our translator remains the same for all of the models and the architecture supports multi-lingual and multi-class classification.

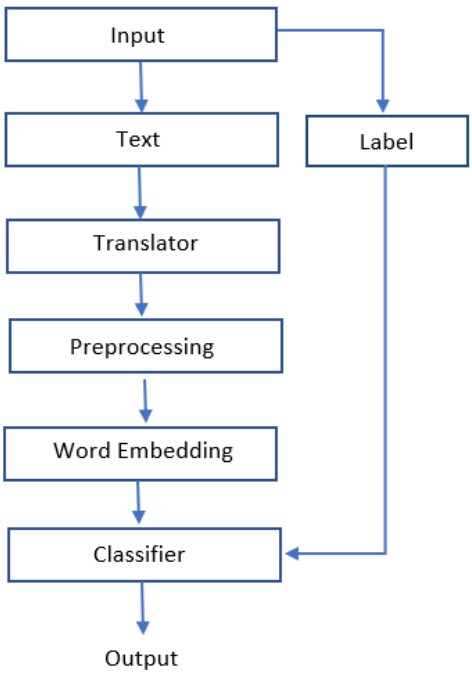

Figure 1: Model Architecture.

The model architecture represents different layers in our model. Input is provided in the form of data frames containing labeled tweets. The NMT translator will identify the text language and translate it into English language. This translated data is then pre-processed and tokenized. Our next layer is the Word Embedding layer which calculates the vector representation of the words in a sentence. The vectorized data is then passed to the Classifier for classification.

Experiments

Depending on the word embedding and the classifier used we divided our experiments into 8 tasks. We used 10-fold accuracy and f1 score metric to fine tune our hyperparameters.

In **Task-1** we are using GloVe word embedding with the LSTM model. There are two GloVe embeddings, standard GloVe word embedding provided by Stanford NLP and the newly trained GloVe embeddings from the paper Detecting Strong and Weak Islamophobia on social media (GloVe DSWI) [45]. The LSTM model has 3 layers: the Embedding Layer, the LSTM Layer, and the Softmax Layer. We fined tuned the hyperparameter Embedding dimension to 300 and neuron count to 256 neurons (LSTM block) in the hidden layer. In **Task-2** we used GloVe word embeddings with the CNN model. We are using the standard GloVe word embeddings provided by Stanford NLP and the newly trained GloVe embeddings from the paper Detecting Strong and Weak Islamophobia on social media (GloVe DSWI) [45]. The CNN model has 4 layers: the Embedding layer, the Convolutional layer, the Max Pooling layer, and the Softmax layer. We fine-tuned the Embedding dimension to 300, the neurons count in the hidden layer to 512 neurons and the kernel (window) size to 2,3,4,5. In **Task-3** we used the Word2vec word embedding with the LSTM model. We used the same hyperparameters for the LSTM model as Task-1. In **Task-4** we used the Word2vec word embeddings with the CNN model. We used the same hyperparameters for the CNN model in this task as Task-2. In **Task-5** we used the BERT embedding within the BERT Model. In **Task-6** we used the BERT embedding within the mBERT Model. In Task-5 and Task-6, to tokenize our text into tokens that correspond to BERT's vocabulary we use BERT tokenizer. We fine-tuned the pre-trained BERT-base-uncased 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 [44]. In **Task-7** we used the SVM model with the TF-IDF and the BOW vectorization method. In **Task-8** we used the RFM model with the TF-IDF and the BOW. We used 200 trees with a maximum depth of 20 nodes. To estimate the potential of all our models on the new data, we use 10-fold cross validation. We also calculated f1 score. For the train-test data split 90 percent of data were allocated to the training set and 10 percent were allocated to the test set.

Vocabulary

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

Similarly, the vocabulary size for untranslated imbalanced English data after preprocessing is 16580 unigrams, for translated imbalanced English data it is 14947 unigrams, and for balanced untranslated Hindi data, it is 20755 unigrams. The distribution of tweet length for untranslated imbalanced English dataset after data pre-processing is 8 words/tweet, for translated imbalanced English dataset it is 8 words/tweet and for the imbalanced untranslated Hindi dataset, it is 13 words/tweet.

Figure 2, 3 and 4 represents the most frequent words after pre-processing in the untranslated English data, translated English data and untranslated Hindi data respectively.

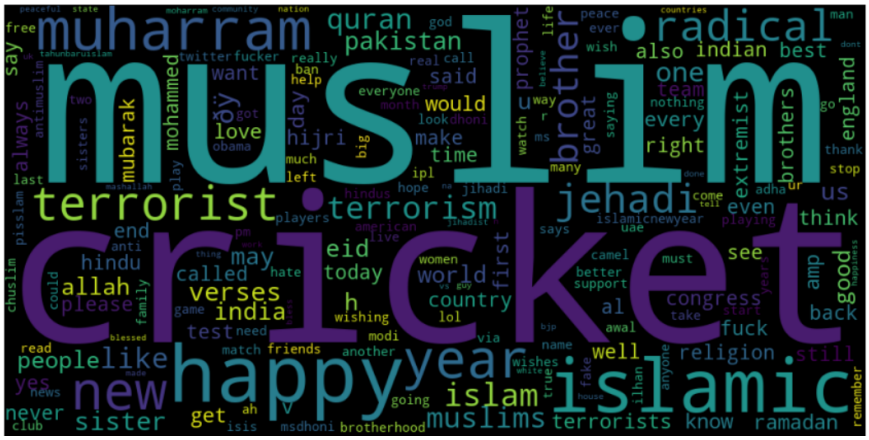

Figure 2: The most frequent words in the untranslated English dataset

The code and dataset developed during this research are available online in a GitHub

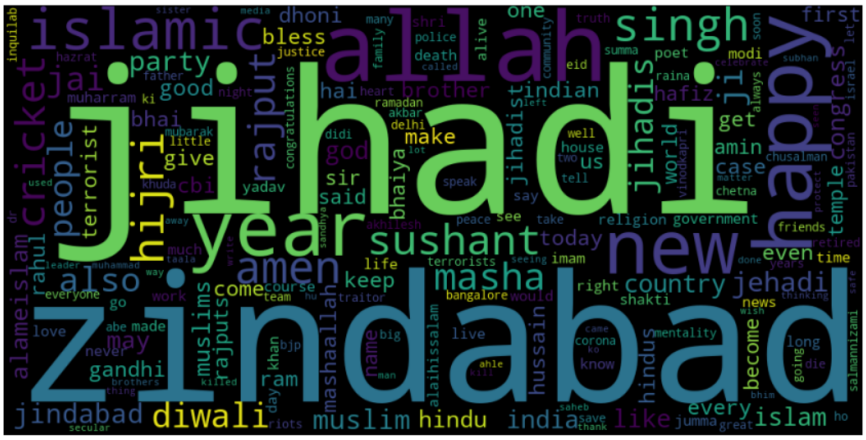

Figure 3: The most frequent words in the translated English dataset

Figure 4: The most frequent words in the untranslated Hindi dataset

repository: https://github.com/hk-mtsu/ Language-agnostic-model-Detecting-Islamophobic-content-on-Social- Media.git.

CHAPTER V.

RESULTS

The result table represents the performance of our model with respect to different tasks. The table contains observation for three dataset categories: untranslated English data (English), translated English data (Translated English) and untranslated Hindi data (Hindi). It contains average of 10-Fold accuracy and F1 Score for both balanced and imbalanced data in each category.

Table 3 represents 10-fold mean accuracy of Task-1 on the untranslated English data (English), the translated English data (Translated English) and the untranslated Hindi data (Hindi). The table also include observation for both balanced and imbalanced data

Table 3: 10-fold cross-validation accuracy of Task-1

Task-1	Dataset	LSTM - balanced		LSTM - imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
GloVe	English	93.67	94	76.43	78
GloVe	Translated Hindi	92.13	92	90.20	90
GloVe	Hindi	43.28	43	42.60	40
Gl ove DSWI	English	33.31	31	42.65	40
GloVe DSWI	Translated Hindi	37.61	38	37.81	37
GloVe DSWI	Hindi	32.61	30	33.54	35

The standard GloVe embeddings from Stanford NLP, when used with LSTM model, performed better with the balanced dataset compared to the imbalanced dataset. The accuracy of translated English is quite similar to the accuracy of untranslated English dataset, where as the GloVe word embeddings along with LSTM did not performed well with untranslated Hindi dataset. On using GloVe DSWI with LSTM, we did not see any learning in our model and hence we have quite low accuracy with both balanced and imbalanced data. As GloVe DSWI is trained on English twitter data it also didn't performed well with the untranslated Hindi dataset.

Table 4 represents 10-fold mean accuracy of Task-2 on the untranslated English data, the

translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

Table 4: 10-fold cross-validation accuracy of Task-2

Task-2	Dataset	CNN - balanced		CNN - imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
GloVe	English	95.96	97	93.31	97
GloVe	Translated Hindi	87.37	86	92.87	91
GloVe	Hindi	91.69	89	89.88	90
GloVe DSWI	English	93.07	90	91.69	91
GloVe DSWI	Translated Hindi	88.53	72	79.82	75
GloVe DSWI	Hindi	89.74	83	89.59	82

The standard GloVe embeddings from Stanford NLP, when used with CNN model, performed better with the balanced data compared to the imbalanced data for untranslated English and untranslated Hindi dataset. Translated English imbalanced data showed slightly better accuracy. The GloVe embeddings with CNN performed best with English data. The difference in the accuracy of the untranslated English dataset and the translated English dataset is 8.5 percent for balanced data and point 4 percent for imbalance data. Whereas the performance of GloVe embeddings along with CNN on untranslated Hindi data was better than translated English data while using balanced dataset, the difference between them is just 3 percent for imbalance data. While the performance of translated data was better by 2 percent on imbalanced data.

On using GloVe from the paper DSWI with CNN model, the accuracy was close to standard GloVe model on English dataset, but on untranslated Hindi dataset the accuracy is 2 percent lower than the standard glove on untranslated data and 1 percent greater than standard GloVe on translated data. GloVe DSWI performed equally well with translated English and untranslated Hindi dataset.

Table 5 represents 10-fold mean accuracy of Task-3 on the untranslated English data, the translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

Table 5: 10-fold cross-validation accuracy of Task-3

Task-3	Dataset	LSTM - balanced		LSTM - imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
Word2vec	English	65.42	66	61.42	60
Word2vec	Translated Hindi	91.76	90	91.45	93
Word2vec	Hindi	87.78	85	86.17	86

The Word2vec word embedding, when used with LSTM model performed better with the balanced data compared to the imbalanced data for the untranslated English, the translated English data and the untranslated Hindi data, but the accuracy is very close with the difference of just 1 percent for untranslated Hindi data and 4 percent for the untranslated English data, but the accuracy is almost same for the translated English data. The accuracy of translated English dataset when is better than the accuracy of English dataset. The Word2vec word embedding along with LSTM performed well with both translated English and untranslated Hindi dataset. The difference in the accuracy of translated English data and untranslated Hindi data is 4 percent for the balanced dataset and 5 percent for imbalanced dataset. The untranslated English dataset performed poorly with the LSTM model.

Table 6 represents 10-fold mean accuracy of Task-4 on the untranslated English data, the translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

Table 6: 10-fold cross-validation accuracy of Task-4

Task-4	Dataset	CNN - balanced		CNN - imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
Word2vec	English	96.58	98	94.90	95
Word2vec	Translated Hindi	93.50	93	75.39	74
Word2vec	Hindi	92.10	90	85.56	86

The Word2vec word embedding, when used with CNN model performed better with balanced data compared to imbalanced data for both English and untranslated Hindi dataset.

The accuracy translated English dataset is quite close to the accuracy of untranslated Hindi dataset. The difference in their accuracy for the balanced data is 1.04 percent. English Dataset shows the highest accuracy with the difference of 3.07 percent between English and translated Hindi dataset for balanced data.

Table 7 represents 10-fold mean accuracy of Task-5 on the untranslated English data, the translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

Table 7: 10-fold cross-validation accuracy of Task-5

Task-5	Dataset	balanced		imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
BERT	English	96.21	96	95.92	98
BERT	Translated Hindi	94.66	94	93.82	95
BERT	Hindi	94.17	93	93.50	93

The BERT model performed better with balanced data compared to imbalanced data. It performed equally well with the translated English data and the untranslated Hindi data. The BERT has been studied as a potentially promising way to further improve neural machine translation (NMT). In this paper we are using NMT as our translator for translating Non-English text to English. According to the performance of the BERT model shown in table 7 we can say that BERT being a transformer model does not need a NMT translator on top of it. The BERT model gave us the second highest accuracy on our untranslated English, translated English and untranslated Hindi data. BERT model performed equally well for translated Hindi data and untranslated Hindi data, with a difference of 0.5 percent.

Table 8 represents 10-fold mean accuracy of Task-6 on the untranslated English data, the translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

The mBERT model performed better with our balanced data compared to imbalanced data for both English and Hindi dataset. It performed equally well with the translated

Table 8: 10-fold cross-validation accuracy of Task-6

Task-6	Dataset	balanced		imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
mBERT	English	95.55	96	92.41	97
mBERT	Translated Hindi	94.08	94	80.43	94
mBERT	Hindi	95.15	94	86.94	94

English and the untranslated Hindi data. Since mBERT model is trained on multilingual data it performed exceptionally well with untranslated Hindi data compared to all other models. BERT model gave us the highest accuracy on untranslated Hindi dataset. The mBERT model on comparing performed equally well with the untranslated Hindi data and untranslated English data, with a difference of 0.4 percent, and with a difference of 1.07 percent between the untranslated Hindi and the translated English data.

Table 9 represents 10-fold mean accuracy of Task-7 on the untranslated English data, the translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

Table 9: 10-fold cross-validation accuracy of Task-7

Task-7	Dataset	SVM - balanced		SVM - imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
TF-IDF	English	97.18	97	95.66	95
TF-IDF	Translated Hindi	95.56	96	93.46	93
TF-IDF	Hindi	89.83	90	86.71	86
BOW	English	97.44	97	95.36	95
BOW	Translated Hindi	94.44	95	92.74	92
BOW	Hindi	87.49	88	84.33	84

The SVM model performed better on balanced data compared to imbalanced data. SVM with BOW gave us the highest accuracy for untranslated English data. The SVM with TF-IDF gave us the highest accuracy for translated English Dataset. SVM with TF-IDF and BOW performed equally well with minute difference in the untranslated English and

translated English data. The accuracy for untranslated Hindi data is also quite close to both translated and untranslated English data. The TF-IDF and BOW evaluates how relevant a word is to a document in a collection of documents and assign vector accordingly. There is no pretrained transfer learning involved. Hence the vectors are more domain and dataset relevant.

Table 10 represents 10-fold mean accuracy of Task-8 on the untranslated English data, the translated English data and the untranslated Hindi data. The table also include observation for both balanced and imbalanced data

Table 10: 10-fold cross-validation accuracy of Task-8

Task-8	Dataset	RFM - balanced		RFM - imbalanced	
		10 - Fold	F1 Score	10 - Fold	F1 Score
TF-IDF	English	91.90	95	87.30	89
TF-IDF	Translated Hindi	84.47	86	84.43	84
TF-IDF	Hindi	80.86	81	72.89	74
BOW	English	94.20	96	87.30	89
BOW	Translated Hindi	87.79	89	87.30	83
BOW	Hindi	81.98	83	79.27	76

The RFM model performed better on balanced data compared to imbalanced data. RFM performed better with untranslated English dataset compared to translated to English with a difference of 7.43 percent accuracy using TF-IDF and 6.41 percent using BOW. The performance of RFM model with translated English data is better than untranslated Hindi data, with a difference of 3.61 percent using TF-IDF and 5.81 percent using BOW.

The top 5 models that performed well with non-English data following our model architecture (translated English data) are shown in table 11.

Table 11: Top 5 accuracy's and F1 score from Task 1-8 for our model

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

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 data and far better than the accuracy of the untranslated Hindi dataset.

CHAPTER VI.

DISCUSSION

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 Google Neural Machine Translation (NMT) model. We have made several observations while experimenting with different models and word embeddings. While 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 observed considerable difference in 10-fold accuracy of both the LSTM and the CNN model while using GloVe word embeddings on translated English (from Hindi) and untranslated Hindi data, but the 10-fold accuracy of Word2vec word embedding with the LSTM and the CNN remained quite similar for both translated English and the untranslated Hindi data. We also observed that TF-IDF and BOW performed better with translated English data. But the accuracy for the untranslated Hindi data was also not too bad. This is quite possible because TF-IDF and BOW evaluates how relevant a word is to a document and assign vector accordingly. We observed that SVM and BERT models accuracy is almost similar. But SVM gave us the highest accuracy on untranslated English and translated English data in spite of being such a simple model because the n-grams word embedding and SVM model are domain specific having no pre-training on external data. In our experiment mBERT model gave us the highest accuracy of 95.15 percent on untranslated Hindi dataset. Since mBERT model is trained on multilingual data it performed exceptionally well with untranslated Hindi data compared to all other models, But once translated our 10-fold accuracy and F1-Score for the translated English (from Hindi) data and the untranslated English data where very close using the SVM, the LSTM,the RFM,the BERT and the CNN models. In fact wile using the LSTM model with Word2vec our accuracy for translated English data was better than the untranslated English dataset. Some of our models like the CNN, the BERT and the SVM also performed well

with untranslated Hindi data, but the performance of translated data was consistently good with all the models.

CHAPTER VII.

CONCLUSION AND FUTURE WORK

We aimed to create a classifier for detecting Islamophobic content on social media using pretrained word-embedding 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 as shown in table 11 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 that will need to be constantly revisited as the nature of online abuse changes [45]. In the future, we plan to implement our model into an app or extension for the browser that will check potential Islamophobia on the screen. We would like to study other lightweight text classification models like the Projection Attention Neural Network (PRADO) and the 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 [21]. The 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 [18]. 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.

BIBLIOGRAPHY

[1] Aljazeera,. *Two in five Americans say Islam 'is incompatible with US values'*, 2018 (accessed on Oct 3, 2020). "https://www.aljazeera.com/news/2018/11/americans-islam-incompatible-values-181101185805274.html.

[2] Ayyub, R. *What a Rising Tide of Violence Against Muslims in India Says About Modi's Second Term, TIME*, 2019 (accessed October 3, 2020). https://time.com/5617161/india-religious-hate-crimes-modi/.

[3] Bajpai, A. *Recurrent Neural Networks: Deep Learning for NLP*, 2019 (accessed March 1, 2021). https://towardsdatascience.com/recurrent-neural-networks-deep-learning-for-nlp-37baa188aef5.

[4] Batuwita, R. and Palade, V. Class imbalance learning methods for support vector machines. 2013.

[5] Cloud, G. *Translating text*, 2020 (accessed October 3, 2020). https://cloud.google.com/translate/docs/basic/translating-text.

[6] Davidson, T., Warmsley, D., Macy, M., and Weber, I. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, 2017.

[7] de Gibert, O., Perez, N., García-Pablos, A., and Cuadros, M. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*, 2018.

[8] Deitch, E. A., Barsky, A., Butz, R. M., Chan, S., Brief, A. P., and Bradley, J. C. Subtle yet significant: The existence and impact of everyday racial discrimination in the workplace. *Human Relations*, 56(11):1299–1324, 2003.

[9] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Forman, G. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

[11] Hafez, F. Unwanted identities: The 'religion line'and global islamophobia. *Development*, 63(1):9–19, 2020.

[12] Hanson, C. *Huffpost-Subtle Racism*, 2018 (accessed October 3, 2020). `https://www.huffpost.com/entry/subtle-racism_b_14113118`.

[13] Hatebase,. *Hatebase Database*, (accessed October 3, 2020). `https://hatebase.org/`.

[14] Hussain, S., Usman, A., Habiba, U., Amjad, A., and Amjad, U. Hate crimes against muslims and increasing islamophobia in india. *Journal of Indian Studies*, 5(1):7–15, 2019.

[15] Islamophobia, T. *Islamophobia endurance in Social Media*, 2019 (accessed on October 3, 2020). `http://twistislamophobia.org/en/2019/05/20/islamophobia-endurance-in-social-media/`.

[16] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651, 2016.

[17] Kai Schultz, Suhasini Raj, J. G. and Kumar, H. *In India, Release of Hate Crime Data Depends on Who the Haters Are, The Newyork Time*, 2019 (accessed October 3, 2020). `https://www.nytimes.com/2019/10/24/world/asia/india-modi-hindu-violence.html`.

[18] Kaliamoorthi, P. *Google AI - Advancing NLP with Efficient Projection-Based Model Architectures*, 2020 (accessed October 3, 2020). https://ai.googleblog.com/2020/09/advancing-nlp-with-efficient-projection.html.

[19] Khan, F. R., Iqbal, Z., Gazzaz, O. B., and Ahrari, S. Global media image of islam and muslims and the problematics of a response strategy. *Islamic studies*, pages 5–25, 2012.

[20] Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[21] Krishnamoorthi, K., Ravi, S., and Kozareva, Z. 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)*, pages 5013–5024, 2019.

[22] Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, 2018.

[23] Labs, E. *FACEBOOK INDIA-TOWARDS A TIPPING POINT OF VIOLENCE CASTE AND RELIGIOUS HATE SPEECH*, 2019 (accessed on Oct 3, 2020). https://www.equalitylabs.org/facebookindiareport.

[24] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[25] MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., and Frieder, O. Hate speech detection: Challenges and solutions. *PloS one*, 14(8), 2019.

[26] Marsh, S. *Record number of anti-Muslim attacks reported in UK last year, The Guardian*, 2018 (accessed October 3, 2020). https://www.theguardian.com/uk-news/2018/jul/20/record-number-anti-muslim-attacks-reported-uk-2017.

[27] Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[28] Mohta, P. *Fuelled by social media, in India Muslims are "a convenient scapegoat" for the coronavirus*, 2020 (accessed on Oct 3, 2020). "EqualityLabs-https://www.equaltimes.org/fuelled-by-social-media-in-india?lang=en#.XzzE3ehKhPZ.

[29] Montantes, J. *Deep Learning for Natural Language Processing (NLP) using RNNs CNNs*, 2019 (accessed October 3, 2020). https://www.kdnuggets.com/2019/02/deep-learning-nlp-rnn-cnn.html.

[30] Mulki, H., Haddad, H., Ali, C. B., and Alshabani, H. L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118, 2019.

[31] News, B. *China Uighurs: Xinjiang legalises 're-education' camps*, 2018 (accessed October 3, 2020). https://www.bbc.com/news/world-asia-45812419.

[32] NPR, C. *Study Shows Islamophobia Is Growing In The U.S. Some Say It's Rising In Chicago, Too*, 2019 (accessed on Oct 3, 2020). https://www.npr.org/local/309/2019/05/03/720057760/study-shows-islamophobia-is-growing-in-the-u-s-some-say-it-s-rising-in-chicago-too.

[33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[34] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[35] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[36] Post, T. W. *Hate crimes against Muslims hit highest mark since 2001*, 2016 (accessed October 3, 2020). https://www.washingtonpost.com/world/national-security/hate-crimes-against-muslims-hit-highest-mark-since-2001/2016/11/14/7d8218e2-aa95-11e6-977a-1030f822fc35_story.html.

[37] Rao, A. and Spasojevic, N. Actionable and political text classification using word embeddings and lstm. *arXiv preprint arXiv:1607.02501*, 2016.

[38] Saha, P., Mathew, B., Goyal, P., and Mukherjee, A. Hatemonitors: Language agnostic abuse detection in social media. *arXiv preprint arXiv:1909.12642*, 2019.

[39] Sasnal, P. and Menouar, Y. E. *There's a social pandemic poisoning Europe: hatred of Muslims*, 2020 (accessed October 3, 2020). https://www.theguardian.com/commentisfree/2020/sep/28/europe-social-pandemic-hatred-muslims-blm.

[40] Shahbaz, H. *Islamophobia, White Supremacy, and the Far-Right*. PhD thesis, Graduate Theological Union, 2020.

[41] Shih, G. Independent—Islamophobia in china on the rise fuelled by online hate speech, 2017 (accessed October 3, 2020). https://www.independent.co.uk/news/world/asia/islamophobia-china-rise-online-hate-speech-anti-muslim-islam-nangang-communist-party-government-xinjiang-a7676031.html.

[42] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[43] Uysal, A. K. and Gunal, S. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112, 2014.

[44] Valkov, V. *Intent Recognition with BERT using Keras and TensorFlow 2*, (accessed March 1, 2021). `https://www.kdnuggets.com/2020/02/intent-recognition-bert-keras-tensorflow.html`.

[45] Vidgen, B. and Yasseri, T. Detecting weak and strong islamophobic hate speech on social media. *Journal of Information Technology & Politics*, 17(1):66–78, 2020.

[46] Waseem, Z. and Hovy, D. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93, 2016.

[47] Wen, P. and Auyezov, O. *Turning the desert into detention camps, Reuters*, 2018 (accessed October 3, 2020). `https://www.reuters.com/investigates/special-report/muslims-camps-china/`.

[48] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and others,. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[49] Yin, W., Kann, K., Yu, M., and Schütze, H. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

APPENDIX

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.