Hate Speech Identification in Tweet-Antisemitism perspective

Sunday Cosmos Ngwobia
ngwobias1@udayton.edu

Shubham Kokul
kokuls1@udayton.edu

Abstract

In recent years, the increasing propagation of hate speech on social media and the urgent need for effective counter-measures have drawn significant investment from governments, companies, as well as empirical research. Despite a large number of emerging scientific studies to address the problem, existing methods are limited in several ways, such as the lack of comparative evaluations which makes it difficult to assess the contribution of individual works. And detecting hate speech against one community or a religion is even more difficult, as we need to first detect hate speech from the text and then we need to determine if it is against the particular community of our choosing. Here we are going to demonstrate the method for detecting hate speech against Jew community as known as antisemitic sentiments. We have used corpus from various sources, such as we have used datasets used in previous empirical works and we have used API provided by www.twitter.com and www.gab.com to gather sentiments of the masses as well, as it is important to keep up with the modern slang used represent and particular community or religion, to gather the required seed words we have used www.hatebase.org which has the largest database for all the seed words required to scrape the twitter and gab feeds. We are using 4 annotations namely Hate, Offensive, Anti-Semitic and Neither. We are using Random forest Classifier over two vectoring methods Bag-of-words and TF-IDF, and we are using LSTM as well. Our results show Random forest classifier over TF-IDF give better classification with 88% accuracy.

Keywords

Anti-Semitic, Random Forest Classifier, LSTM, SVM, feature selection,machine learning, classification, NLP, Bag-of-Words, TF-IDF.

Introduction

What constitutes hate speech and when does it differ from offensive language? No formal definition exists but there is a consensus that it is speech that targets disadvantaged social groups in a manner that is potentially harmful to them (Jacobs and Potter 2000; Walker 1994). In the United States, hate speech is protected under the free speech provisions of the First Amendment, but it has been extensively debated in the legal sphere and with regards to speech codes on college campuses. Formal definition of hate speech can be given as following.

Hate speech is commonly defined as any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic (Nockleby, 2000).

Illustrated below are examples of hate, offensive, antisemitic texts and those that are neither of the trio.

Hate and Offensive

- Go fucking kill yourself and die already useless ugly pile of shit scumbag.
- The Jew Faggot Behind The Financial Collapse
- Hope one of those bitches falls over and breaks her leg

Anti-Semitic

- How many wars are you guys going to get in to because of jews fuck it lol ..nobody is talking about this,but in my humble opinion.. the biggest threat against Europe is Israel nuclear bombsyup I am thinking that.
- Fuck off you daft jews

Neither

- The fruit of the modern Yankees farm system. Joba the Nut.
- Traded in my nikes for a new mic.. I guess it's safe to say he sold soles for his new life …
- A Muslim woman covers the yellow star of Jewish neighbor with her veil to protect her, Sarajevo,1941.

Online social networks (OSN) and microblogging websites are attracting internet users more than any other kind of website. Services such those offered by Twitter, Facebook and Instagram are more and more popular among people from different backgrounds, cultures and interests. Their contents are rapidly growing, constituting a very interesting example of the so-called big data. Big data have been attracting the attention of researcher, who have been interested in the automatic analysis of people's opinions and the structure/distribution of users in the networks, etc. While these websites offer an open space for people to discuss and share thoughts and opinions, their nature and the huge number of posts, comments and messages exchanged makes it almost impossible to control their content. Furthermore, given the different backgrounds, cultures and believes, many people tend to use and aggressive and hateful language when discussing with people who do not share the same backgrounds. However, nowadays, with the rapid growth of OSN, more conflicts are taking place, following each big event or other. Nevertheless, while the censorship of content remains a controversial topic with people divided into two groups, one supporting it and one opposing it [2], in OSN, such language still exists. It is even easier to spread among young people as well as older ones than other ‘‘cleaner’’ speeches. For these reasons, Burnap and Williams [3] claimed that collecting and analyzing temporal data allows decision makers to study the escalation of hate crimes following ‘‘trigger’’ events. However, ‘‘official’’ information regarding such events are scarce given that hate crimes are often unreported to the police. Social networks in this context present a better and more rich, yet less reliable and full of noise, source of information. To overcome this noise and the non-reliability of data, we propose in this work an efficient way to detect offensive posts, hate speeches and Anti-Semitic post in Twitter. Our approach relies on Bag-of-Words, TF-IDF along with LSTM for feature extraction to perform the detection.

Motivation and Related Works

There are plethora of research works on hate speech detection on online social networks such facebook, twitter, etc. Most of the work are majorly on how to classification of hate speech from other speech which could be offensive, profane and abusive. Most of the research works were able to identify hate speech with following feature:

- Targets individual or groups on the basis of their characteristics (targeting characteristics);
- Demonstrates a clear intention to incite harm, or to promote hatred;
- May or may not use offensive or profane words. For example: *All you perverts (other than me) who posted today, needs to leave the O Board. Dfasdfdasfadfs'* is an example of abusive

language, which often bears the purpose of insulting individuals or groups, and can include hate speech, derogatory and offensive language. *'i spend my money how i want bitch it's my business'* is an example of offensive or profane language, which is typically characterized by the use of swearing or curse words

According to Hajime,Mondher Bouazizi and Tomoaki(2017) said that task of hate speech detection is quite different and more challenging than sentiment analysis: not only is it context-dependent, but also, we should not rely on simple words or even n-grams to detect it. In their research they used word vectorization and deep learning approach in classifying tweets into hate, offensive and clean.

Anna S. and Michael W. (2017). Conducted a survey of the various methods and approaches used in recent years to detect hate speeches and their associated limitations. They recommend the use of surface-level features, such as bag of words, unigrams and larger n-grams for feature engineering. They opined that these feature engineering approached has been reported to highly predictive. They also lay emphasis on consideration on the use other features techniques such as linguistic features, lexical resources, sentiment analysis, knowledge-based features and met-information for enhancing detection and classification of hate speech. They also suggested that hate speech classifiers are predominantly of supervised approach such as Support Vector Machine (SVM) as well as deep learning approaches (especially Recurrent Neural Network Language Models) are most recent methods.

Shanita B. (2018) explained how natural language processing (NLP) can be applied in detecting hate speech via the use of current techniques (i.e. classical and deep learning methods) in this field on a dataset. She delineated that major task that NLP application is useful are namely classification, matching, translation, structured prediction, and sequential decision proces. They established the approaches in NLP according to Liddy (2001) which are Symbolic approaches, Statistical approaches, Connectionist approaches and hybrid approaches. She stated that the existing methods for text classification can be divided into two categories: classical methods and deep learning methods which corroborate the general view of other authors.

Thomas D., Dana W., Michael M and Ingmar W. (2017). Expressed how to separate hate speech from offensive language using logistic regression with L1 regularization, Naïve Bayes, decision trees, random forests, and linear SVMs. In feature extraction and construction, utilized unigram, bigram and unigram each weighted by its TF-IDF after lemmatization and used NLTK to capture the syntactic structure of information in the dataset.

Hajime W., Mondher B. and Tomoaki O. (2017). Delineated how to separate a hateful, offensive and clean speeches from a text using ternary classification with an accuracy of 78.4%. They employ a combination of sentiment-based features, semantic features, unigram features and pattern features engineering techniques for detecting hate speech in tweets

Ziqi Z. and Lei L. (2018). Deduced that the performance of existing automated methods or approaches in identifying hate speech is unsatisfactory which was as result of hate speeches residing in the "long-tail" of dataset. This long-tail situation is a function of extremely unbalanced presence of real hateful content in the typical datasets, and the lack of unique, discriminative features in such content. Therefore, they suggested the use of concatenated deep learning networks (CNN and GRU) structure to address the challenge

Resham A. and et'al (2018). In their research paper, presents how to use machine learning models for detection of misogyny, i.e. hate speech against women, in English tweets. They use used bag of word (BOW) and lexical features extraction techniques on their dataset corpus and which they used in training their models which includes Random Forest classifier (RF), a Logistic Regression classifier (LR), a Stochastic Gradient Descent (SGD) classifier, a Gradient Boosting (GB) classifier and LSTM.

After going through some of the research works as mentioned above, we discovered that their hate speech detection approach is more generic and no attention was paid to any given ethnicity. Hence, we proposed implement a system that will detect form the pool of hate speeches those that are antisemitic. Our proposed will integrate different sets of features including language patterns and hate speech extracted using TF-IDF, BOW and word2vec. We use these features together to perform the classification based on Random forest classifier of text collected over the internet into 4 classes we refer to as 'Neither', 'Offensive', 'Hate' and 'Anti-Semitic'.

The main contribution of this paper are as follows:

- We are able to detect hate speech(anti-Semitic) for a particular community (i.e. Jewish) we can use this approach to detect hate speech a specific community.
- We classify text in 4 classes creating a clear difference between offensive and hate and hate against one religion.

Corpus Acquisition

One of the most important tasks in our work is corpus acquisition as that would help us get better results if we have good annotated data at our disposal, we can change the hyper parameters in the training task and try to find the perfect settings for hate speech detection. We have used the seed words provided by www.hatebase.org for Anti-Semitic to mine tweets containing those words. There are around 40 Anti-Semitic words, which can be used to collect the corpus. After collecting the tweets and cleaning the text we have manually annotated each of the tweet, and categorized them in the following categories Hate, Neither, Offensive and Anti-Semitic. The other datasets which we found was from the following blog *https://www.groundai.com/project/hate-speech-dataset-from-a-white-supremacy-forum/*, we also collected some post from *https://gab.com* it is a prominent social media platform for right wing ideology, and it is marketed as Free Speech over internet, it is quite openly used in Europe. It has been really good inspiration for data as well. Apart from these two data sources we have used data corpus found on kaggle *https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data* which is annotated and has been used in previous works as well, the next source has the similar background *https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data*. The most difficult task was to merge the data and give all them the common annotation. So as to make the training data rich enough and generic enough to weed out outliners while running the detection over the model created by it. We also used human annotator to annotate for Anti-Semitic tweets. And merged the same with the already acquired datasets. Our dataset after combining the previous works dataset and our text scrapped over twitter and gab has over 27000 unique post which are segregated as given below. We tried to create balance data as it is important to have balanced data for better and effective results. We have plotted the table and as well as the pie chart for the same. We have also used word-cloud to get most used words in each of the labels this gives us the insight for the pattern of words used in the classification.

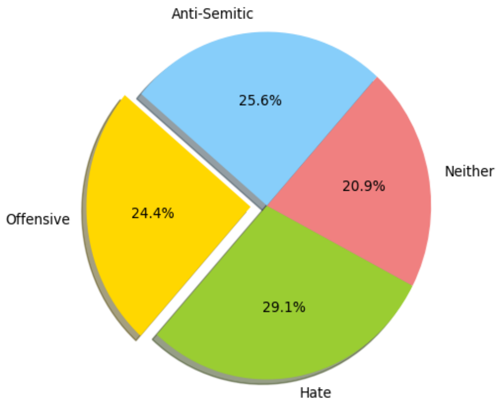


Fig 1: *Percentage distribution of dataset across different labels*

Table 1: *Dataset size for each label*

Labels	**Size**
Hate	7857
Offensive	6588
Neither	5643
Anti-Semitic	6912

Proposed Approach

In approach we have used three different type of feature extraction namely Bag-of-words, LSTM and TFIDF. And over one classifier random forest classifier, we have used different hyper-parameters to train the data. And we have tested it on different setting to understand the proper setting for each of the feature extraction and as well as for classification. We have simple data pre-processing, after cleaning the data we curated the data into single Data-Frame for further process. As we are taking supervised learning approach, we have carefully taken into the consideration the labels used in the study. We are using confusion matrix to represent our results, we have also done matric evaluation for accuracy, precision and F1 score as well. Apart from that we have used different methodologies for data extraction. We have used twitter API to get the required tweets which have the seed words used to describe antisemitic sentiment along with that we have also used Gab API as well to collect the required post for hate and Anti-Semitic to have different region perspective, as slang changes with the change in the region. Apart from that we have also used a proper annotated dataset which was used in previous work so that we get good combination of current slang and also the data which is termed as Hate from experts as well.

Given a set of Tweets, the aim of this work is to classify each of them into one of three classes which are:

- Neither: This class consists of tweets which are neutral, non-offensive and present no hate speech.
- Offensive: This class contains tweets that are offensive, but do not present any hate or a segregative/racist speech
- Hateful: This class includes tweets which are offensive, and present hate, racist and segregative words and expressions.
- Antisemitic: This class includes tweets which are offensive, and present hate, racist and segregative words and expressions specifically against Jew and Jew community.

We use machine learning to perform the classification: we extract a set of features from each tweet, we refer to a training set and perform the classification. We have divided the dataset into 70% and training and 30% testing data for our system.

To perform the task of classification, the data set is split into three subsets as follows:

- A training set: This set contains 18900 tweets, distributed evenly among the three classes (i.e. ‘Neither,’ ‘Offensive’, ‘Hateful’ and ‘Anti-Semitic’): each class has 4725 tweets. This set will be referred to as the “training set” in the rest of this work.
- A test set: this set contains 4050 tweets: each class has 1012 tweets. This set will be referred to as the “test set” and will be used to optimize our proposed approach.
- A validation set: this set contains 4050 tweets: each class has 1012 tweets. This set will be referred to as the “validation set” and will be used to evaluate our proposed approach.

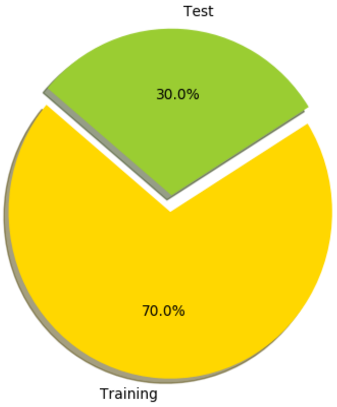
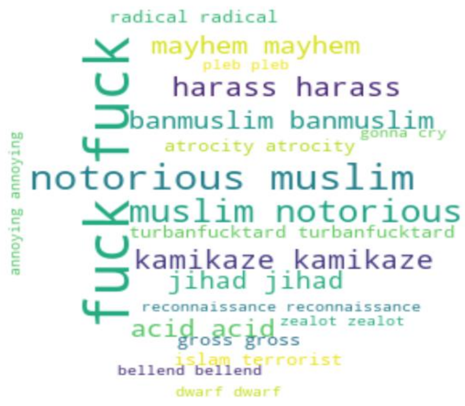
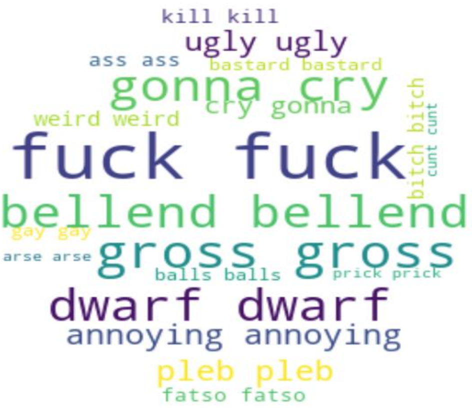


Fig. 2: *Percentage split of total datasets into training and testing datasets*

To get fair result, we use the same number of tweets for each set. Given that the number of tweets in “Neither’ class was 5643 and it is the least among the three classes, we set the number of training tweets for each class to 5500 tweets, that of the test tweets to 1012 tweets and that of the validation tweets to 1012.

We even created and plotted a Word Cloud to mention and get the words used most frequently hate, offensive and Anti-Semitic and the result were very insightful. These words can be used as features as well and can be termed as discriminating features used in different vectoring methods.

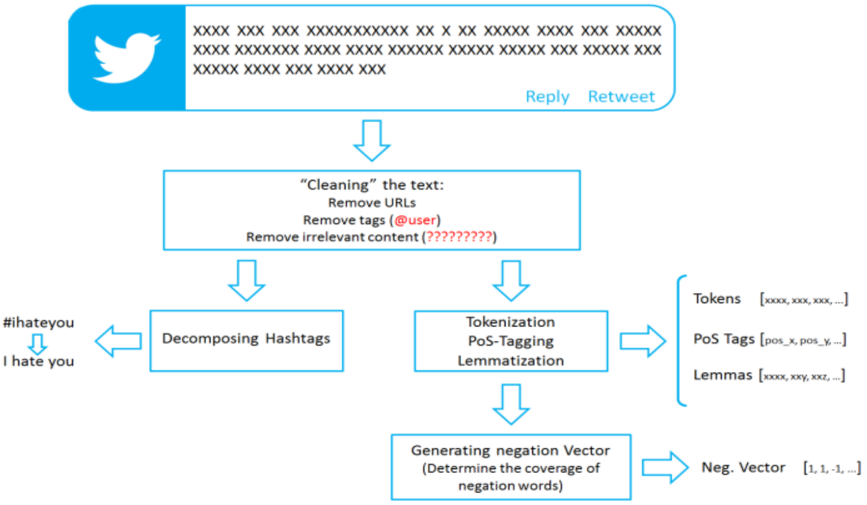
Discriminate Features

Image	**Feature Description**
 **Hate Word Cloud**	We can term a word as hate word if is used to hurt the sentiment of a person based on his community, religion or faith. These words are also used to segregate the community and agitate a particular community against each other. There is very fine line between Anti-Semitic and Hate Speech based on our observation.
 **Offensive Word Cloud**	The offensive words can be termed as the curse words used in different languages. These words are meant to hurt people but cannot be termed as hate words as they are not target to a particular community or religion but in general. This words are basically meant for all the people and can be transferred to different languages. And can be used with translations as well. These words also fall into the category of bullying.

 Anti-Semitic Word Cloud	The main take away from this word cloud is to get the idea of the seed words or slur used against the Jew community which can be termed as Anti-Semitic as well. So, the seed words gathered from www.hatebase.org closely resemble this word cloud which reinforces our assumption that we need to search the OSN with modern seed words.

Data Pre-Processing

In this section, we briefly describe how the tweets were preprocessed. The below figure shows the different steps done during this phase.



XXXX XXX XXX XXXXXXXXXXX XX X XX XXXXX XXXX XXX XXXXX XXXX XXXXXXX XXXX XXXX XXXXXX XXXXX XXXXX XXX XXXXX XXX XXXXX XXXX XXX XXXX XXX
Reply
Retweet
"Cleaning" the text:
Remove URLs
Remove tags (@user)
Remove irrelevant content (?????????)
Decomposing Hashtags
#ihateyou
I hate you
Tokenization
PoS-Tagging
Lemmatization
Tokens [xxxx, xxx, xxx, ...]
PoS Tags [pos_x, pos_y, ...]
Lemmas [xxxx, xxy, xxz, ...]
Generating negation Vector
(Determine the coverage of negation words)
Neg. Vector [1, 1, -1, ...]

Fig. 3: *Data pre-processing flow pipeline*

In this section first we start cleaning the text based upon removing the special characters mentioned in the tweet and the post the special characters include hash-tags, 'http://' or 'https://' and also the user-based tags such as '@user' and irrelevant expression as well. We have also transformed abbreviated words into their full meanings (e.g. 'BS' → Bullshit', we've → we have, etc.) and we lowercased each tweet, this step was carried out to make the dataset more uniform so as to get more effective vectors. After basic cleaning with the help of regex and beautiful soup API we moved to getting the stop words out of the text as well. The next step included stemming and lemmatization of words with the help of Stemmer and Porter library. We got the total number of token and vocabulary in our dataset as stats with the help NLTK library. On a last step, we extract all the hashtags, and usea small tool we developed to decompose it into the words that compose it (e.g., the hashtag "#ihateyou" will give the expression "I hate you") and are kept aside to be used when needed.

Features Extraction and Classifier

This is the second step in tour proposed system and it entails the extraction and classification techniques we used to vectorized words content of our textual corpus and onward classification of that. There are multiple word vectorizing methods but we choose CBOW, TF-IDF and Word2Vec for our system in order to take advantage of feature extraction capabilities. CBOW and TF-IDF helped us takes the context of each word as the input and tries to predict the word corresponding to the context while Word2vec is great for digging into documents into word embeddings that we can train our deep neural network on.

In nutshell, we used CBOW, TF-IDF and word embedding for extracting the discriminative features that we help our classifier to make accurate and less error classification on our targeted datasets. We used python's *CountVectorizer* for bag of words vectors, *TfidfVectorizer* for TF-IDF, and LSTM embedding for word2Vec

Classifiers

The classifiers used in this research were Random Forest and bidirectional Recurrent Neural Network (RNN) which we implemented using *sklearn* and *keras* libraries respectively. We chose to use these classifiers because of the robustness and they are also among the state-of-the-art classifiers.

Baseline and Model Pipeline Diagram

We developed the code based on different blogs and some of the implementation was found on GitHub. The contribution of each member is divided on the bases of the task performed previously we tried to prioritize data collection and then data annotation before we started implementing classifier as data would give us the idea on how to bifurcate the coding task based on the skillset. The implementation pipeline for running our experiment is diagrammatical shown in the figure below.

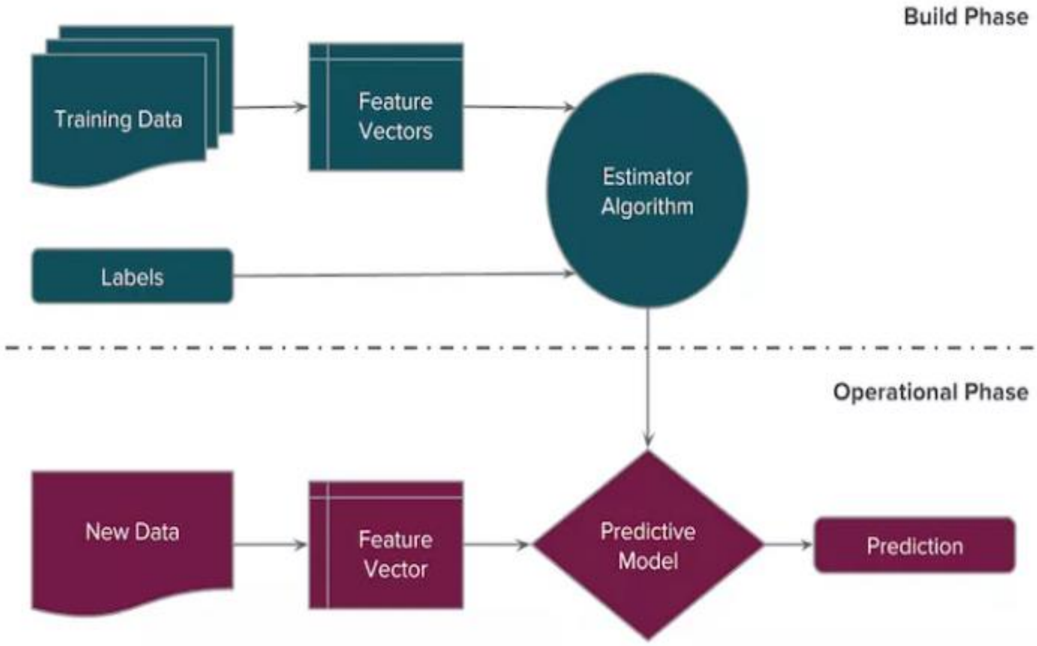


Fig. 4: *Classification Pipeline for our Proposed Models*

Implementation Environment and Libraries used

We are able to achieve implementation objective of this via python 3.6 over pip environment and libraries. Below are listed of the major libraries we utilized.

SN	Library name	Description
1	Gensim	A free Python library designed to automatically extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) as possible
2.	NLTK	A suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language
3.	Pandas	A software library for data manipulation and analysis, it offers data structures and operations for manipulating numerical tables and time series.
4.	Sklearn	A library in Python that provides many unsupervised and supervised learning algorithms.
5.	matlibplot	A Python 2D plotting library which produces publication quality figures in a variety of formats
6.	tweepy	an open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API
7	Numpy	A fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrice), etc.

Experimental Setup and Results

After the extraction of features using classical python libraries, we proceed to experimenting with various code algorithms for the various classifiers. The classification experiment was first done using Random Forest Classifier by feeding the classifier with trained dataset which has already undergone pre-processing and feature extraction stage. We used TF-IDF and CBOW feature extraction techniques at this stage and metric results obtained while training our model as well using it in classifying our test dataset were recorded.

We also carried out another classification experiment with RNN model, whereby with trained the model on our train dataset that is pre-processed and feature engineered using word embedding vectorization techniques (Word2Vec). The result obtained during and a testing of this model was equally recorded.

The basic metric evaluation techniques used in representing our experimental results are precision, recall, accuracy and loss scores of our model training and predictive performance. The mathematical definitions of these metrics are:

$$F1 = 2 \times \frac{Precision \cdot Recall}{Precision + Recall}$$

$$\text{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

CBOW over Random Forest Classifier

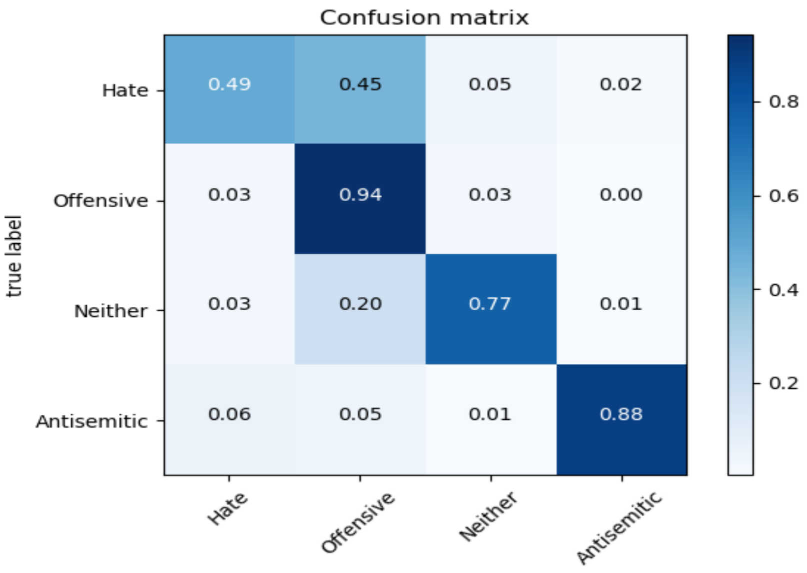
As mentioned in proposed approach above, we have used 18900 tweets to train the classifier and we got vector for each of the word. Converted those vectors to array and pushed it to the Random forest (RF) classifier. Below are the metric tables of the results obtained for the RF classifier when trained on our dataset that we engineered the feature using CBOW feature extraction techniques.

Labels	Precision	Recall	F1-score	Support
Hate	0.49	0.53	0.60	693
Offensive	0.94	0.93	0.92	928
Neither	0.77	0.85	0.83	826
Antisemitic	0.88	0.83	0.80	897

Micro-Macro Score
The below table mentions the micro and macro score of the CBOW results

Labels	Precision	Recall	F1-score	Support
Micro avg	0.87	0.87	0.87	887
Macro avg	0.79	0.79	0.79	805
Weighted avg	0.87	0.87	0.87	887

Further we have plotted confusion matrix of the same results to identify the proper classification and the misclassification of the text.



Error Analysis

- The precision for Anti-Semitic is 88% which gives us 897 tweets correctly classified as anti-Semitic, the 122 tweets were not classified which gives us the percentage loss of 13% for CBOW over random forest classifier.
- The precision for Hate is 49% which gives us 693 tweets correctly classified as Hate the 327 tweets were not classified which gives us the percentage loss of 13% for CBOW over random forest classifier.
- The precision for Offensive is 94% which gives us 928 tweets correctly classified as Offensive, the 92 tweets were not classified which gives us the percentage loss of 13% for CBOW over random forest classifier.
- The precision for Neither is 77% which gives us 826 tweets correctly classified as Neither, the 194 tweets were not classified which gives us the percentage loss of 13% for CBOW over random forest classifier.

TF-IDF over Random Forest Classifier

As mentioned in the mentioned in proposed approach we have used 18900 tweets to train the classifier and we got vector for each of the word. Converted those vectors to array and pushed it to the Random forest classifier. Below are the metric tables of the results obtained for the RF classifier when trained on our dataset that we engineered the feature using TF-IDF feature extraction techniques.

Labels	Precision	Recall	F1-score	Support
Hate	0.40	0.69	0.51	408
Offensive	0.96	0.75	0.84	979
Neither	0.59	0.88	0.70	601
Antisemitic	0.67	0.80	0.73	683

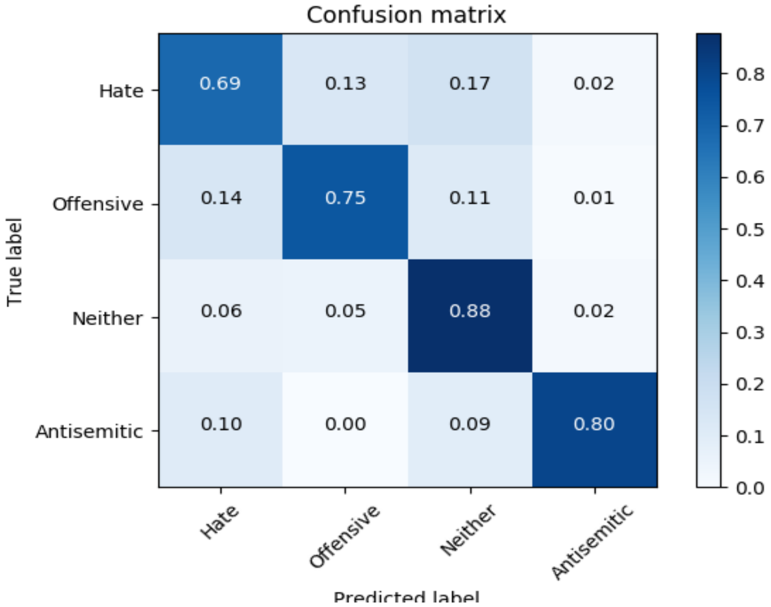
Train accuracy: 0.79
Test accuracy: 0.76

Micro-Macro Score

The below table mentions the micro and macro score of the TFIDF results

Labels	Precision	Recall	F1-score	support
Micro avg	0.76	0.76	0.76	775
Macro avg	0.65	0.78	0.69	663
Weighted avg	0.83	0.76	0.78	846

Further we have plotted confusion matrix of the same results to identify the proper classification and the misclassification of the text.



Analysis Error

- The precision for Anti-Semitic is 67% which gives us 785 tweets correctly classified as antisemitic, the 337 tweets were not classified which gives us the percentage loss of 33% for TFIDF over random forest classifier.
- The precision for Hate is 40% which gives us 408 tweets correctly classified as Hate, the 615 tweets were not classified which gives us the percentage loss of 60% for TFIDF over random forest classifier.
- The precision for Offensive is 96% which gives us 979 tweets correctly classified as Offensive, the 61 tweets were not classified which gives us the percentage loss of 4% for TFIDF over random forest classifier.
- The precision for Neither is 77% which gives us 826 tweets correctly classified as Neither, the 194 tweets were not classified which gives us the percentage loss of 13% for TFIDF over random forest classifier.

Word2Vec over LSTM

As mentioned in the mentioned in proposed approach we have used 18900 tweets to train the classifier and we got vector for each of the word. In LSTM as it internally uses Word2Vec with the word length of 130 and the embedding dimension is 128 and the batch size is 256 drop out of 0.7 activation function is SoftMax and optimizer is Adam optimizer. Below are the metric tables of the results obtained for the RNN classifier when trained on our dataset that we engineered the feature using LSTM word embedding feature extraction techniques.

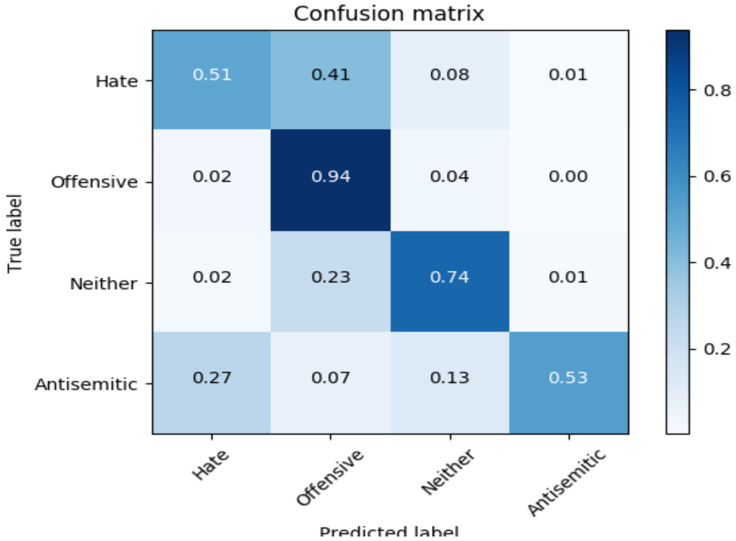
Label	Precision	Recall	F1-score	Support
Hate	0.51	0.51	0.58	520
Offensive	0.94	0.94	0.91	958
Neither	0.74	0.74	0.76	754
Antisemitic	0.53	0.53	0.63	540

Loss: 0.429
Accuracy: 0.851

Micro-Macro Score

The below table mentions the micro and macro score of the LSTM results

Label	Precision	Recall	F1-score	Support
micro avg	0.85	0.85	0.85	6790
macro avg	0.78	0.68	0.72	6790
weighted avg	0.84	0.85	0.84	6790



Error Analysis

- The precision for Anti-Semitic is 53% which gives us 520 tweets correctly classified as antisemitic, the 500 tweets were not classified which gives us the percentage loss of 57% for Word2Vec over LSTM.
- The precision for Hate is 91% which gives us 958 tweets correctly classified as Hate, the 62 tweets were not classified which gives us the percentage loss of 9% for Word2Vec over LSTM.
- The precision for Offensive is 94% which gives us 754 tweets correctly classified as Offensive, the 266 tweets were not classified which gives us the percentage loss of 6% for Word2Vec over LSTM.
- The precision for Neither is 77% which gives us 540 tweets correctly classified as Neither, the 480 tweets were not classified which gives us the percentage loss of 23% for Word2Vec over LSTM.

Conclusion

If we conflate hate speech and offensive language then we erroneously consider many people to be hate speakers. So, we needed a system which can detect hate speech, offensive language and Anti-Semitic sentiments and segregate them as per their classification. We were able to achieve this objective by running Random forest classifier over TF-IDF and CBOW and we also used LSTM over word2vec and in their comparison we observed that CBOW work better among their peers, with the precision of 88%. and the loss of 12%. From our own observation and insight, the 12% text which were misclassified due to misclassified annotation (Human Error).

The other aspect of misclassification can be attributed to very slim difference between Hate speech Offensive and Anti-Semitic which can even make a human cross-verify his annotation. Recall of our system in comparison is better for CBOW as well. The poor performance of LSTM could be attributed to the hyper parameters. We can further increase the accuracy of the system if we can find proper annotated data, and data on large scale. In the below table we can summarize the performance of the various classifiers.

Model	Label	Precision	Recall	F1-score
Word2Vec over LSTM	Hate	0.51	0.51	0.58
	Offensive	0.94	0.94	0.91
	Neither	0.74	0.74	0.76
	Antisemitic	0.53	0.53	0.63
TF-IDF over Random Forest Classifier	Hate	0.40	0.69	0.51
	Offensive	0.96	0.75	0.84
	Neither	0.59	0.88	0.70
	Antisemitic	0.67	0.80	0.73
CBOW over Random Forest Classifier	Hate	0.49	0.53	0.60
	Offensive	0.94	0.93	0.92
	Neither	0.77	0.85	0.83
	Antisemitic	0.88	0.83	0.80

References

Anna S. and Michael W. (2017). *A Survey on Hate Speech Detection using Natural Language Processing*. Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pages 1–10, Valencia, Spain, April 3-7, 2017. c 2017 Association for Computational Linguistics

Shanita B. (2018). *Hate Speech Detection Using Natural Language Processing Techniques*. https://beta.vu.nl/nl/Images/werkstuk-biere_tcm235-893877.pdf

Thomas D., Dana W., Michael M and Ingmar W. (2017). *Automated Hate Speech Detection and the Problem of Offensive Language.* Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)

Hajime W., Mondher B. and Tomoaki O. (2017). *Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection*. *IEEE Access* 6. 13825–13835. DOI: 10.1109/ACCESS.2018.2806394.

Ziqi Z. and Lei L. (2018). *Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter*. arXiv preprint arXiv:1803.03662.

Resham A., Himani S., Edward C., Anderson N. and Martine D. (2018). *Detecting Hate Speech Against Women in English Tweets*. http://ceur-ws.org/Vol-2263/paper032.pdf