HATE SPEECH DETECTION IN SOCIAL MEDIA

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science and Engineering

by

ADHYAN

(Roll No. 2017BCS0003)

to

DEPARTEMENT OF COMPUTER SCIENCE

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

KOTTAYAM-686635, INDIA

November 2020

DECLARATION

I, **ADHYAN** (**Roll No: 2017BCS0003**), hereby declare that, this report entitled

”“HATE SPEECH DETECTION IN SOCIAL MEDIA

” submitted to Indian Institute of Information Technology Kottayam towards partial requirement of Bachelor of Technology in Computer Science and Engineering is an original work carried out by me under the supervision of Dr.Manjith BC and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **ADHYAN**
November 2020

CERTIFICATE

This is to certify that the work contained in this project report entitled "HATE SPEECH DETECTION IN SOCIAL MEDIA" submitted by ADHYAN (Roll No: 2017BCS0003) to Indian Institute of Information Technology Kottayam towards partial requirement of Bachelor of Technology in IIIT kottayam has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635
November 2020

Dr.Manjith BC
Project Supervisor

ABSTRACT

With the ever-growing online content, hate speech is also spreading fast. This leads to need of systems that can detect and stop this from happening. In this project, I identify and explore the challenges facing online automated methods of finding hate-speech in the text. The difficulties are language nuance, different definitions of hate speech and restriction on data availability for these systems.In addition, many modern methods have a translation problem - that is, it can be difficult to understand why these systems make the decisions as they do. I propose a SVM multi-view approach that can achieve high-level performance, is simple and produces more easily understandable decisions than other methods.

Contents

Chapter 1

Introduction

Crimes due to hate are not new. However, social media and other online forums play a major part in hate crimes. For example, accused in various hate-related terrorist attacks had a broad history of social media of hate-related posts, suggesting that social media led to their modification. In some other cases, social media plays a direct role.

Large online forums and social media, provide users the ability to communicate freely, sometimes even hiding their identity. While this ability to freely express is a human-right to be admired, spreading hatred to other parties is a violation of that freedom. Thus, many online forums such as Facebook, YouTube, and Twitter view hate speech as harmful, and have policies to remove the content of hate speech. Given the public's concern and the widespread use of hate speech on the Internet, there is a strong encouragement to learn the detection of hate speech. With itsdetection, the

distribution of hateful content can also be reduced.

Finding hate-speech is a daunting task though. In the first place, there is the disagreement in hate speech definition. Some things may be considered hate-speech to some peoplebut not to others, depending on their ownspeculations. I begin by covering over competing explanations, focusing on various factors that add up to hate-speech. We have never been, and neither we can be perfect, as new definitions are constantly emerging. My goal is simply to show the differences that highlight the difficulties that arise from that.

Competitive explanations provide the challenges of testing programs to detect hate speech; existing data sets differ in their definition of hate speech, leading to data sets that not only come from different sources, but also capture different details. This can make it difficult to determine exactly which aspects of hate speech you might identify. I discuss the various data sets available to train and measure the effectiveness of hate speech programs in the next section. Nuance and subtlety of language present additional challenges to hate speech recognition, and depending on the meaning.

Apart from the differences, some recent approaches have found promising results in finding hate speech in text content. The proposed solutions use machine learning techniques to classify text as hate speech. One of the limitations of these methods is that the decisions they make can't be seen and it is difficult for people to interpret why the decision was made. This is a practical concern because systems that automatically process a person's speech may require a personal appeal process. To address this issue, I propose a new way of distinguishing hate speech that allows for a better understanding of decisions and shows that it can bypass existing methods in other data sets. Some of the existing methods use external sources, such as a dictionary of hate speech, in their programs. This can work, but it requires the maintenance of these resources and keeping them up to date which is a problem in itself. Here, my approach does not depend on external resources and achieves the right accuracy. I cover these topics in the next section.

In general, however, there are practical challenges left between all programs. For example, armed with the knowledge that the platforms they use are trying to silence them, those who want to distribute hateful content are desperately trying to find ways to avoid the actions taken. I cover this topic in detail in the last paragraph.

In summary, I discuss the challenges and methods of automatic detection of hate speech, including competing explanations, data acquisition and structure, and existing methods. I also suggest a new way to in some cases surpass the state of the art and discuss the remaining errors. Finally, I conclude the following:

1. **Without a public context, programs cannot do enough.**
2. **Automatic detection of hate speech is legally difficult.**
3. **Other methods achieve optimal performance.**
4. **Certain challenges remain in all solutions.**

Chapter 2

HATE SPEECH DEFINITION

The definition of hate speech is universally accepted and a few parts of the definition are completely inconsistent. Ross, et al. believe that a transparent definition of hate speech can help within the study of finding hate speech by making hate speech a straightforward task, and thus, making the annotations more reliable. However, the road between hate speech and appropriate speech is blurred, making some people reluctant to grant hate speech a transparent meaning. I summarize the prominent definitions of hate-speech in an exceedingly form of sources, furthermore as other aspects of the definition that make the discovery of hate-speech difficult.

The Encyclopaedia of the American Constitution: “Hate speech is an expression of hatred for someone or group on the idea of such characteristics as race, religion, ethnic origin, ethnic origin, gender, disability, sexual orientation, or sexual identity.”

Davidson .: "Language accustomed express hatred towards a bunch intended or intended to denigrate, humiliate, or insult members of a gaggle."

Facebook: “We define hate speech as an instantaneous attack on people supported what we call protected factors - race, nationality, ethnic origin, religious affiliation, sexual orientation, nationality, gender, gender, sexual orientation, and heavy illness or disability. We also provide immigration protection. We define harassment as violent or demeaning, demeaning statements, or to hunt exclusion or partiality.”

Fortuna: “Hate speech is offensive or derogatory language, which creates violence or hatred in groups, looking on factors like physical appearance, religion, demeanour, nationality or ethnicity, physical attraction, sexual orientation or otherwise, and will vary in language styles, even in subtle ways or when humour.” This definition is predicated on their analysis of varied meanings.

Twitter: "Hateful behaviour: you can not promote violence against or directly or indirectly threatens others on the idea of race, nationality, ethnic origin, sexual orientation, gender, sexual orientation, religious affiliation, age, disability or serious illness.”

de Gilbert .: "Hate speech is that the intentional attack on a selected group of individuals motivated by the characteristics of a group’s identity.”

It is noteworthy that in a number of the above descriptions, things has to be addressed to the group. this is often in contrast to the definition of the

Encyclopaedia of the American Constitution, where human aggression may be considered hate speech. a standard theme among definitions is that an attack relies on a specific group feature or personality. While de Gilbert's definition of ownership itself remains unclear, some definitions provide certain patent features. Particularly, these saved features are the characteristics of Davidson et al. and Facebook descriptions. the outline of Fortuna et al. It needs special diversity in language style and cunning. this could be challenging, and it goes beyond the quality text editing method you'll be able to handle.

The description of Fortuna et al. it's supported the analysis of the subsequent factors from other definitions:

1. That jokes are often considered hate speech
2. Hate speech to incite violence or hatred
3. Hate speech has certain purposes
4. Hate speech to attack or reduce

A problem that's not covered by many explanations is expounded to facts. As an example, "pig Jews" could be a hate speech with many definitions (it could be a statement of contempt), but "many Jewish lawyers" don't seem to be. within the latter case, to work out whether each statement could be a hate speech, we'll have to examine whether the statement is true or not using external sources. this type of hate speech is difficult because it's associated with the verification of real truth — another difficult task. additionally, to check the technicality, we'll first need to explain the precise wording of the word, that is, “many” within the total number or percentage of individuals, which further complicates the matter.

Another problem arising within the definition of hate-speech is that the possible praise of a hateful group. During this case it's important to grasp which groups are considered hate groups and what's actually recommended by the group as some undoubtedly recommend it, and unfortunately it's true.

Chapter 3

DATA USED

Gathering and processing data for automated classifiers training to detect hate speech is a challenge. Moreover, identifying and acknowledging that a text is hate-speech is difficult, as previously stated, hate speech cannot be universally defined. Ross, et al. studied the validity of the annotations of hate speech and suggested that the annotations were unreliable. Agreement between annotations, measured using Krippendorff's , was very low (up to 0.29). However, annotations were compared based on the description of Twitter, by comparing the annotations based on their ideas and found a solid connection.

In addition, social media is a source of hate speech, but many have strong terms for using and disseminating data. This leads to a small number of datasets available for the public, most of which come from Twitter. While Twitter services are important, their normal use is limited due to the unique nature of Twitter posts;

character limitations result in a straightforward, short-form text. Conversely, posts from other platforms are often remote and can be part of a larger discussion on a particular topic. This provides additional context that can affect the meaning of the text.

Another challenge is that there is simply not much information available in public, identified by texts that identify hate, aggression, and profanity.

•Hatebase Twitter

Twitter datasetwith a collection of 24,802 tweets. The most widely used subset of this dataset is available, containing 14,510 tweets.

•Waseem

This is also provided from Twitter, containing 16,914 tweets labelled discriminatory, sexual, or not. From this corpus, the authors themselves commented on (16,914 tweets) and conducted major sex studies reviewing the annotations.

•Stormfront

Provided from a post from the top white forum, Stormfront, this describe post-sentence posts that lead to 10,568 goals written as Hate, NoHate, Relation, or Skip. It also captures the context value annotation used to separate text.

•TRAC

Focused on obtaining aggressive text in both English and Hindi. Aggressive text is often part of hate speech. Dataset from this activity is publicly available and contains 15,869 Facebook views labelled as extremely aggressive, subtle, or non-aggressive.

●HatEval

It contains several sets of labels. The first shows whether the tweet expresses hatred towards women or foreigners, the second, whether the tweet is violent, and the third, whether the tweet is directed at one person or a group. Note that identifying a person is not considered hate speech by all definitions.

●Kaggle

The dataset contains 8,832 social media comments labelled offensive or non-offensive.

●GermanTwitter

Twitter dataset in German of the European refugee crisis. It has 541 tweets in German, labelled as expressing hate or not. It should be noted that these data sets vary greatly in scope, size, descriptive data features, and features of hate speech. The most common source of text is Twitter, which contains online short forms. While the Twitter datasets capture a wide range of hate speech in various languages such as attacking different groups, the construction process including filtering and sampling methods introduces uncontrolled corporate analytics.

Chapter 4

RELATED WORKS

Most social networks have developed user rules that prohibit hate speech; enforcing these rules, however, requires a lot of manual labour to review all reports. Some platforms, such as Facebook, have recently increased the number of content moderators. Default tools and methods can speed up the review process or empower employees for posts that require close personal testing. In this section, we will look at the automatic methods of detecting hate speech from text.

4.1 KEYWORD-BASED METHODS

The basic method of identifying hate speech is the use of keywords. By means of a dictionary, a text that contains hateful words can be identified. For example, Hatebase maintains a database of deprecatory terms in many languages. Well-maintained resources are important, as terms change over time. However, as I have seen in my study of the definition of hate speech, simply using a hatefulword is not enough to make hate speech. Ways based on keywords are quick and directlyunderstood. Still, they have serious shortcomings. Receiving only racist insults will lead to a more accurate system but with a lower memory where accuracy is a fair percentage from the set obtained and recalling the appropriate percentage from people around the world. To put it differently, a program that relies heavily on keywords will not identify hateful content that do not have these keywords. Conversely, including words that are not always hateful (e.g., ”trash”, ”swine”, etc.) can create too many wrong decisions, increasing precision memory. In addition, keyword-based approaches cannot identify hate speech without hate keywords.

4.2 SOURCE ANNOTATIONS

Explained information from social media can help further understand the features of the post and can lead to a better identification process. Details such as the sender’s location, location, timestamp, or public participation on the platform can all provide an ongoing understanding of the post in a different granularity. However, this information is not readily available to external researchers as the publication of sensitive user information raises privacy issues. External investigators may share or other user information. Therefore, they may solve the faulty puzzle or learn based on incorrect information from the data. For example, a system trained in this data may incorrectly choose to mark the content of certain users or groups as hate speech based on emerging dataset features. Using user information may raise certain ethical issues. Models or systems may be biased against certain users and often mark their posts as hateful even if some of them are not present. Similarly, relying too much on personal information can miss posts from unfamiliar users to post hateful content. Marking posts as hate based on user statistics can create a shocking feeling on the platform and ultimately reduce freedom of speech.

4.3 MACHINE LEARNING CLASSIFIERS

Machine learning models take labelled text samples to produce distinctions that can detect hate speech based on labels defined by content reviewers. Various solutions were proposed and proved to be successful in the past. I describe the selection of open-sourced programs that have been shown in recent research. •

4.3.1 Data pre-processing and feature extraction

To identify or classify user-generated content, hateful text features should be removed. The obvious features of individual words or phrases (n-grams, i.e., sequence of consecutive words). To improve the similarity of features, words can be limited to finding only the root that removes morphological differences. Metaphor analysis, e.g., Neuman, et. al. it can similarly produce features. The bag-of-words concept is often used in text separation. Under this assumption, posts are simply represented as a set of words or n-grams without order. This view certainly omits an important aspect of language, yet proved to be powerful in many works. In this setting, there are various ways to allocate metals to less important terms, such as TF-IDF. For a review of retrieving general information, see. Apart from the distribution features, word embedding, i.e., giving a vector to a word, such as word2vec, is common when using in-depth learning methods in natural language processing and text capture. Other in-depth learning techniques, such as duplicate networks and neural transformer, challenge word-of-word thinking by modelling word order by processing word order embedded. •

4.3.2 Various Models and their performances

Naïve Bayes, Support Vector Machine and Logistic Regression

These are often used in text classification. The Naïve Bayes model label the possibilities of direct occurrence by assuming that the features do not match. Support Vector Machines (SVM) and Logistic Regression are linearclassifiers that predict classes based on a combination of points for each feature. An open source implementation of these types exists, for example in the well-known Python learning package for sci-kit learn.

FastText

FastText is an effective isolation method proposed by researchers on Facebook. The model generates the n-gr character embedding and provides model speculation based on embedding. Over time, this model has become a solid foundation for many text-splitting operations

Neural Ensemble

Zimmerman, et al. proposes a composite approach, which combines the conclusions of ten convolutional neural networks with various weighted beginnings. The structure of their network is similar to that suggested by, with a combination of 3 lengths integrated across the length of the document. The results of each model are summarized by measuring points.

C-GRU

C-GRU, the Convolution-GRU Based Deep Neural Network proposed by Zhang, et al., combines convolutional neural networks (CNN) and gated recurrent networks (GRU) to detect hate speech on Twitter. They did a number of experiments on Twitter publicly available data sets that demonstrated their ability to capture word order and order in a short text. Note, on the Hatebase Twitter database , they treat both Hate and Offensiveas Hate that leads to the binary label instead of its original multi-class label. In my testing, I use original labels with a variety of types where different model test results are expected.

BERT

BERT is a pre-training embedded training model based on a transformer-expanded model that is a separate layer with an additional output layer. It accomplishes artistic work through the separation of texts, answering questions, and the adoption of language without corrective action. When testing BERT, I add an equal layer above the separation token and check all the suggested tuning hyperparameters.

Chapter 5

PROPOSED METHOD

I propose a SVM multi-view model for hate speech classification. It uses a multiple-view stacked Support Vector Machine (mSVM). Each feature type is inserted into the Linear SVM classifier (inverse regularization constant C = 0.1), creating a view-classifiers for those features. Iwill also incorporate these view-classifiers with another Linear SVM (C = 0.1) to produce a meta-classifier. The features used in the meta-classifierwill be possibilities for each label with each view-separator. Integrating machine learning phases is not a new concept. Previous efforts have shown that combining SVM with various fragments provides the development of various data mining and text-sharing operations. Combining multiple SVMs (mSVMs) has also been proven to be an effective means of image processing operations to reduce the size problem. However, using multiple SVMs to target hate speech expands the scope of use of this category beyond what was previously tested. Multi-view readings are known for capturing different data views. In the context of the discovery of hate speech, incorporating different perspectives captures the various aspects of hate speech within the classification process. Instead of combining all the elements into a vector of one element, each view-classifier learns to separate a sentence according to only one type of feature. This allows the viewing categories to take different aspects of the pattern individually. Combining all sorts of features into a single model, in doing so, puts the vulnerability of weak but important signals at risk. In this case, my proposed model is able to pick up this feature in one of the view-classifiers, where there are a few parameters. In addition, this model offers the opportunity to translate the model to identify which visual divisions are most contributing using a meta-classifier that provides human understanding of the category. The visual separator that contributes most to the final decision identifies key words (features) that lead to a hate speech label. This compares with well-functioning neural models, which are often opaque and difficult to understand. Even modern methods of self-care suffer from loud noise that greatly reduces interpretation.

5.1 Setup

Using various hate-speech datasets, I will evaluate the accuracy of existing as well as my hate speech detection approaches.

5.1.1 Data pre-processing and features

For simplicity and normal operation, pre-processing and feature identification are less intentional. In pre-processing, I use case folding, word lemmatizationand remove punctuation marks. For features, I simply extract Word TF-IDF from unigram to 5-gram and N-gram character counts from unigram to 5-gram.

5.1.2 Datasets

I will be examining the TRAC, HatEval, and Hatebase Twitter described earlier. These data sets provide a variety of meanings and expressions for hate speech (including many forms of harassment), as well as many forms of online content (including online forums, Facebook content, and Twitter). In the TRAC dataset, I will use English Facebook language training, validation, and classification tests. At HatEval, I will use a set of training set to validate and use a valid verification dataset for testing because the official test set is not publicly available. Finally, in the Hatebase Twitter dataset ,I will use the standard separation of the train test provided by .

5.1.3 Evaluation

I will test the effectiveness of each method using the accuracy and macro-averagedf1-score results. There is no rule in the literature on which test metrics to use. However, I believe that focusing on both accuracy and macro-f1 provides a good understanding of the strengths and weaknesses of each method.

Chapter 6

RESULTS AND DISCUSSION

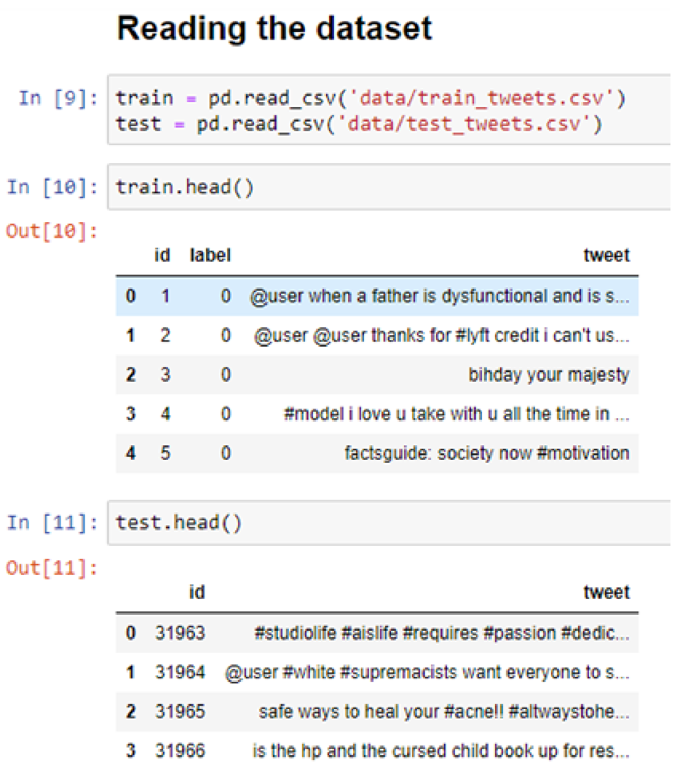
6.1 Work Completed Until Now:

At the time of writing this report, I have performed the above-mentioned data preprocessing and feature extraction operations on HateBase Twitter dataset.

The vectvariable contains the vectorized inputs to be given to the view-classifier. After this, I trained various models that are already available and following are the results obtained

6.2 Work to be done in next phase

From the obtained results, we can see that Random Forest Classifier performs as the best view-classifier. I will further apply all these classifiers once again on the features obtained to create a meta-classifier and observe which method proves to be the best.



Reading the dataset

```
In [9]:  train = pd.read_csv('data/train_tweets.csv')
         test = pd.read_csv('data/test_tweets.csv')
```

```
In [10]: train.head()
```

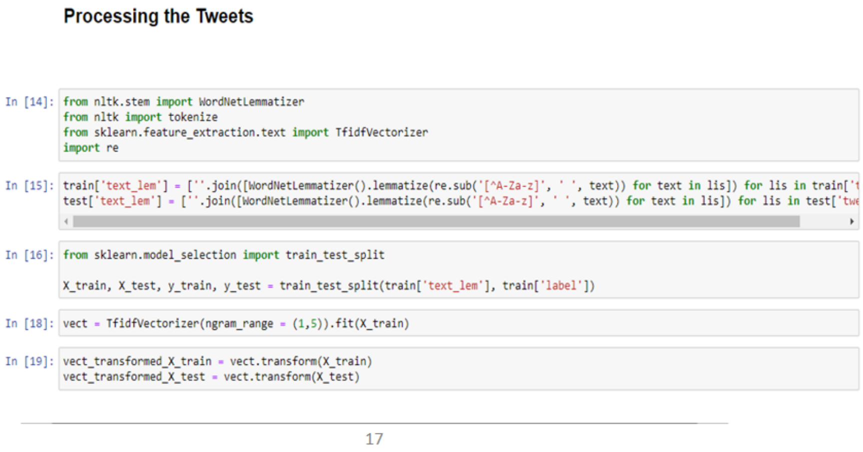
Out[10]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

```
In [11]: test.head()
```

Out[11]:

	id	tweet
0	31963	#studiolife #aislife #requires #passion #dedic...
1	31964	@user #white #supremacists want everyone to s...
2	31965	safe ways to heal your #acne!! #altwaystohe...
3	31966	is the hp and the cursed child book up for res...



Processing the Tweets

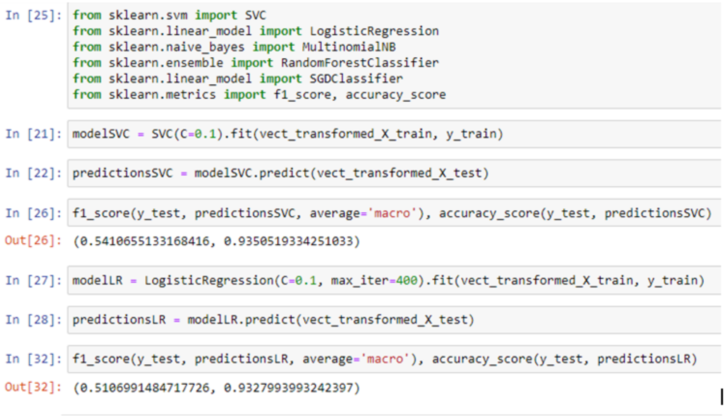
```
In [14]: from nltk.stem import WordNetLemmatizer
         from nltk import tokenize
         from sklearn.feature_extraction.text import TfidfVectorizer
         import re
```

```
In [15]: train['text_lem'] = [''.join([WordNetLemmatizer().lemmatize(re.sub('[^A-Za-z]', ' ', text)) for text in lis]) for lis in train['t
         test['text_lem'] = [''.join([WordNetLemmatizer().lemmatize(re.sub('[^A-Za-z]', ' ', text)) for text in lis]) for lis in test['twe
```

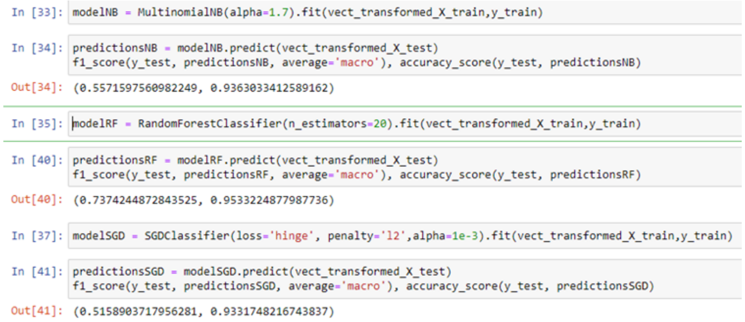
```
In [16]: from sklearn.model_selection import train_test_split

         X_train, X_test, y_train, y_test = train_test_split(train['text_lem'], train['label'])
```

```
In [18]: vect = TfidfVectorizer(ngram_range = (1,5)).fit(X_train)
```

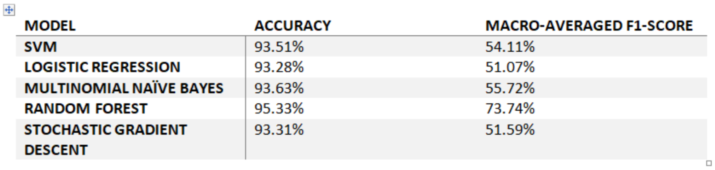
```
In [19]: vect_transformed_X_train = vect.transform(X_train)
         vect_transformed_X_test = vect.transform(X_test)
```



```
In [25]: from sklearn.svm import SVC
         from sklearn.linear_model import LogisticRegression
         from sklearn.naive_bayes import MultinomialNB
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.linear_model import SGDClassifier
         from sklearn.metrics import f1_score, accuracy_score

In [21]: modelSVC = SVC(C=0.1).fit(vect_transformed_X_train, y_train)

In [22]: predictionsSVC = modelSVC.predict(vect_transformed_X_test)

In [26]: f1_score(y_test, predictionsSVC, average='macro'), accuracy_score(y_test, predictionsSVC)

Out[26]: (0.5410655133168416, 0.9350519334251033)

In [27]: modelLR = LogisticRegression(C=0.1, max_iter=400).fit(vect_transformed_X_train, y_train)

In [28]: predictionsLR = modelLR.predict(vect_transformed_X_test)

In [32]: f1_score(y_test, predictionsLR, average='macro'), accuracy_score(y_test, predictionsLR)

Out[32]: (0.5106991484717726, 0.9327993993242397)
```



```
In [33]: modelNB = MultinomialNB(alpha=1.7).fit(vect_transformed_X_train,y_train)

In [34]: predictionsNB = modelNB.predict(vect_transformed_X_test)
         f1_score(y_test, predictionsNB, average='macro'), accuracy_score(y_test, predictionsNB)

Out[34]: (0.5571597560982249, 0.9363033412589162)

In [35]: modelRF = RandomForestClassifier(n_estimators=20).fit(vect_transformed_X_train,y_train)

In [40]: predictionsRF = modelRF.predict(vect_transformed_X_test)
         f1_score(y_test, predictionsRF, average='macro'), accuracy_score(y_test, predictionsRF)

Out[40]: (0.7374244872843525, 0.9533224877987736)

In [37]: modelSGD = SGDClassifier(loss='hinge', penalty='l2',alpha=1e-3).fit(vect_transformed_X_train,y_train)

In [41]: predictionsSGD = modelSGD.predict(vect_transformed_X_test)
         f1_score(y_test, predictionsSGD, average='macro'), accuracy_score(y_test, predictionsSGD)

Out[41]: (0.5158903717956281, 0.9331748216743837)
```



MODEL	ACCURACY	MACRO-AVERAGED F1-SCORE
SVM	93.51%	54.11%
LOGISTIC REGRESSION	93.28%	51.07%
MULTINOMIAL NAÏVE BAYES	93.63%	55.72%
RANDOM FOREST	95.33%	73.74%
STOCHASTIC GRADIENT DESCENT	93.31%	51.59%

Chapter 7

CONCLUSION

As hate speech continues to be a social problem, the requirement for systems to detect hate speech is natural. I have introduced the current methods of this work and propose a new system that will find the right accuracy. I have also suggested a new way to surpass existing programs in this work, with the benefit of better-interpretation. Given all the remaining challenges, there is a need of further research into this problem, including both practical and technical issues.

Chapter 8

ACKNOWLEDGEMENTS

I thank Dr. Manjith BC for reviewing the early versions of this project and helpful feedback which made me find various perspectives of doing this work. I also thank my friends Shubham and Somesh for their insightful comments on the topic.

Chapter 9

Bibliography

1. Zimmerman S, Kruschwitz U, Fox C. Improving Hate Speech Detection with Deep Learning Ensembles. In: LREC; 2018.

2. Ross B, Rist M, Carbonell G, Cabrera B, Kurowsky N, Wojatzki M. Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. In: The 3rd Workshop on Natural Language Processing for Computer-Mediated Communication @ Conference on Natural Language Processing; 2016.

3. Wermiel SJ. The Ongoing Challenge to Define Free Speech. Human Rights Magazine. 2018;43(4):1–4. o View Article o Google Scholar

4. Nockleby JT. Hate Speech. Encyclopedia of the American Constitution. 2000;3:1277–79. o View Article o Google Scholar

5. de Gibert O, Perez N, Garc'ia-Pablos A, Cuadros M. Hate Speech Dataset from a White Supremacy Forum. In: 2nd Workshop on Abusive Language Online @ EMNLP; 2018.

6. Popat K, Mukherjee S, Yates A, Weikum G. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In: EMNLP; 2018.

7. Hatebase;. Available from: https://hatebase.org/.

8. Waseem Z, Hovy D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: SRW@HLT-NAACL; 2016.

9. Waseem Z. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In: Proceedings of the first workshop on NLP and computational social science; 2016. p. 138–142.

10. Kumar R, Ojha AK, Malmasi S, Zampieri M. Benchmarking Aggression Identification in Social Media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). ACL; 2018. p. 1–11.

11. Robertson C, Mele C, Tavernise S. 11 Killed in Synagogue Massacre; Suspect Charged With 29 Counts. 2018;.

12. The New York Times. New Zealand Shooting Live Updates: 49 Are Dead After 2 Mosques Are Hit. 2019;.

13. Hate Speech—ABA Legal Fact Check—American Bar Association;. Available from: https://abalegalfactcheck.com/articles/hate-speech.html.

14. Community Standards;. Available from:

https://www.facebook.com/communitystandards/objectionable$_{c}ontent.$

15. Hate speech policy—YouTube Help;. Available from:

https://support.google.com/youtube/answer/2801939.

16. Hateful conduct policy;. Available from:

https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy.

17. Mondal M, Silva LA, Benevenuto F. A Measurement Study of Hate Speech in Social Media. In: ACM HyperText; 2017.

18. Fortuna P, Nunes S. A Survey on Automatic Detection of Hate Speech in Text. ACM ComputSurv. 2018;51(4):85:1–85:30. o View Article o Google Scholar

19. Davidson T, Warmsley D, Macy MW, Weber I. Automated Hate Speech Detection and the Problem of Offensive Language. ICWSM. 2017;

20. CodaLab—Competition;. Available from:

https://competitions.codalab.org/competitions/19935.

21. Detecting Insults in Social Commentary;. Available from: https://kaggle.com/c/detecting-insults-in-social-commentary.

22. Neuman Y, Assaf D, Cohen Y, Last M, Argamon S, Howard N, et al. Metaphor Identification in Large Texts Corpora. PLoS ONE. 2013;8(4). o View Article o Google Scholar

23. Salton G, Yang CS, Wong A. A Vector-Space Model for Automatic Indexing. Communications of the ACM. 1975;18(11):613–620. o View Article o Google Scholar

24. Grossman DA, Frieder O. Information Retrieval: Algorithms and Heuristics. Berlin, Heidelberg: Springer-Verlag; 2004.

25. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Dis-

tributed Representations of Words and Phrases and their Compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in Neural Information Processing Systems 26. Curran Associates, Inc.; 2013. p. 3111–3119.

26. Arroyo-Fernández I, Forest D, Torres JM, Carrasco-Ruiz M, Legeleux T, Joannette K. Cyberbullying Detection Task: The EBSI-LIA-UNAM system (ELU) at COLING'18 TRAC-1. In: The First Workshop on Trolling, Aggression and Cyberbullying @ COLING; 2018

27. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805 [cs]. 2018;.

28. Yang Z, Chen W, Wang F, Xu B. Unsupervised Neural Machine Translation with Weight Sharing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics; 2018. p. 46–55. Available from: http://aclweb.org/anthology/P18-1005.

29. Kuncoro A, Dyer C, Hale J, Yogatama D, Clark S, Blunsom P. LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 1426–1436. Available from: http://aclweb.org/anthology/P18-1132.

30. Aroyehun ST, Gelbukh A. Aggression Detection in Social Media: Using Deep Neural Networks, Data Augmentation, and Pseudo Labeling. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). Santa Fe, New Mexico, USA: Association for Computational Linguistics; 2018. p. 90–97. Available from:

https://www.aclweb.org/anthology/W18-4411

31. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. JMLR. 2011;12:2825–2830. o View Article o Google Scholar

32. Kim Y. Convolutional Neural Networks for Sentence Classification. In: EMNLP; 2014.

33. Hagen M, Potthast M, Büchner M, Stein B. Webis: An Ensemble for Twitter Sentiment Detection. In: SemEval@NAACL-HLT; 2015.

34. Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of Tricks for Efficient Text Classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. ACL; 2017. p. 427–431.

35. Zhang Z, Robinson D, Tepper J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference. Springer; 2018. p. 745–760.

36. Zhao J, Xie X, Xu X, Sun S. Multi-view learning overview: Recent progress and new challenges. Information Fusion. 2017;.

37. Opitz D, Maclin R. Popular ensemble methods: An empirical study. Journal of artificial intelligence research. 1999;11:169–198. o View Article o Google Scholar

38. Chand N, Mishra P, Krishna CR, Pilli ES, Govil MC. A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. In: 2016 International Conference on Advances in Computing, Communication, Automation (ICACCA)(Spring). IEEE; 2016. p. 1–6.

39. Waseem Z. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In: NLP+CSS @ EMNLP; 2016

40. Dong YS, Han KS. Boosting SVM classifiers by ensemble. In: Special interest tracks and posters of the 14th international conference on World Wide Web. ACM; 2005. p. 1072–1073.

41. Abdullah A, Veltkamp RC, Wiering MA. Spatial pyramids and two-layer stacking SVM classifiers for image categorization: A comparative study. In: 2009 International Joint Conference on Neural Networks. IEEE; 2009. p. 5–12.

42. Jain S, Wallace BC. Attention is not Explanation. ArXiv. 2019;abs/1902.10186.

43. Serrano S, Smith NA. Is Attention Interpretable? In: ACL; 2019.

44. Vig J. Visualizing Attention in Transformer-Based Language Representation Models. arXiv preprint arXiv:190402679. 2019