Hate Speech Detection

Devansh Mody (1130532)
Department of Computer Science
Lakehead University
dmody@lakeheadu.ca

Yidong Huang (1116935)
Department of Computer Science
Lakehead University
yhuang36@lakeheadu.ca

Abstract—**A remarkable development in the areas of deep learning and natural language processing algorithms in the last few decades have changed the way we browse on internet or interact with a machine. Natural language processing is important because it helps to resolve ambiguity in the language and adds useful numeric structure to the data for many downstream applications, such as speech recognition, text analytics, text summarization, question answering system, chat bots and many other similar tasks have become easier to implement. As the number of users are increasing on world wide web the amount of data generated by them is also increasing daily.**

Presently in the era of internet and electronic devices, hate speech has proven consequential to youth internet users and previous methods relied heavily on the use of manually developed dictionaries. Due to the massive rise of user generated web content on social media, the amount of hate speech is also steadily increasing. Hate speech is a challenging issue plaguing the online social media. While better models for hate speech detection are continuously being developed, there is little research on the bias and interpretability aspects of hate speech. Therefore detecting hate speech on the internet nowadays is a state of the art topic in the machine learning or deep learning and natural language processing field.

This paper will focus on building a system to detect whether a given text is a hate speech or not. Further in this paper we will describe the most effective methods proposed by few of the authors in their research work on hate speech, our proposed approach or solution for hate speech detection with system overview diagram, comparison of proposed solutions with the literature review, finally we would like to discuss a few common challenges or problems faced by authors and how to overcome those challenges or problems and conclusion.

Keywords: Convolutional Neural Network (CNN), Unigram (UI), Bigram (BI), Trigram (TI), Long Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Recurrent Neural Network (RNN), Support Vector Machine (SVM), Bidirectional Encoder Representations from Transformers (BERT), Hate Speech (HS), Parts of speech (POS), Named entity recognition (NER), Term frequency inverse document frequency (TFIDF), Gated Recurrent Unit (GRU), Generative Pretrained Transformer (GPT2), Naive Bayes (NB), Random Forests (RF), Decision Trees (DT), K-nearest neighbours (KNN), Bilingual Evaluation Understudy Score (BLEU), Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Local Interpretable Model-agnostic Explanations (LIME), Graphics Interface Format (GIF).

I. Introduction

In today's digital world also known as the era of the internet, people are living around social media like youtube, twitter etc., nowadays offensive content has become pervasive in social media and thus pose a serious concern for government organizations, online communities, and social media platforms, because of the hate speech in our world wide web. Hate speech also used to exists before digital media but now in the age of internet it spreads faster on social media platforms and other kinds of media and its difficult to track the source of the hate speech or the one who is spreading hate and provoking anger.

More recently the NLP community witnesses a growing interest in tasks related to social and ethical issues, also encouraged by the global commitment to fighting extremism, violence, fake news and other plagues affecting the online environment. One such phenomenon is hate speech, a toxic discourse which stems from prejudices and intolerance and which can lead to episodes, and even structured policies, of violence, discrimination and persecution. In many countries, including the United Kingdom, Canada, and France, there are laws prohibiting hate speech [21]. According to the paper from Michel, the meaning of hate speech is ”speech designed to promote hatred on the basis of race, religion, ethnicity or national origin poses vexing and complex problems for contemporary constitutional rights to freedom of expression” [19]. With the time change, the hate speech can extend to personal attacks like body shame, or other bullying behaviour on the world wide web. This is connected even with concrete public health issues, since recent studies show that victims are more likely to suffer from psycho-social difficulties and affective disorders [20].

One of the most common strategies to tackle the problem is to train systems capable of recognizing offensive content, which can then be deleted or set aside for human moderation. In the last few years, there have been several studies on the application of computational methods to deal with this problem. In this paper, we are going to present a system for detecting hate speech on the internet. The system aims to support the people who have the right to supervise social media(e.g. the manager of Twitter). This paper would explain a system based on two different approach to detect hate speech from textual data. Therefore plan A would be to use a pretrained model like BERT/GPT2, add our own embedding layers, adding encoding and decoding stages, different algorithms for decoding based on temperature, nucleus and topk sampling. Plan B would be to create a voting based system for hate speech detection which includes combination of models like KNN, CNN, LSTM, Naive Bayes and the pretrained model BERT/GPT2. At the same time, this paper will take the text similarity into account, as a part of checking hate speech and different evaluation metrics like ROUGE-L, BLEU score will also be used along with the LIME methods to interpret the performance of the model. Our first aim is to build a one of the proper hate speech model using pretrained models like BERT/GPT2 by changing the model according to our requirements and perform a proper context and subjective analysis to achieve the excellent performance in detecting hate speech, then we intend to check the interpretability of the model using LIME methods and then we would like to develop a voting mechanism to effectively predict the output of different models and make the final decision based on the output.

II. Literature Review

Following is the summary of the research papers surveyed for literature review some of their methods will be incorporated in our research work with additional functionalities and additional analysis based on our extended research work.

- The aim of this paper [1] is to build a detection system for hate speech, we can conclude this paper by three parts: A) Data, the data is from Twitter, using the Twitter API, they build a speech lexicon with words and phrases, and then the sample will manually coded by CrowdFlower (CF) workers, workers were asked to label each tweet as one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech. B) Features, firstly, they lowcased the sample, and then the data were stemmed by using the Porter stemmer, then they created bigram, unigram, and trigram features, each one was done by the TFIDF. C) Model, they run many different algorithms on this paper, which include, logistic regression, naive bayes, decision trees, random forests, and linear SVMs. On those above models, Logistic Regression and Linear SVM are much better than other algorithms. In this process, they use a one versus rest framework on the SVM, I think we also can use one versus one in our approach. The figure 1 describes the approach used by the author.
- This paper [2] uses the machine learning method to detect hate speech with the context. The models of this paper used are logistic regression and neural networks. Firstly, they consider many different features, but we need to clear that the maximum length of sequence they consider is 125. This paper used Word level and Character level Ngram Features, LIWC Feature, NRC Emotion Lexicon Feature. And then based on those features, they run the two algorithms, Logistic regression and neural networks.
- The objective of this paper [3] can be concluded as three things: a)Offensive Language Detection; B)Categorization of Offensive Language; C)Offensive Language Target Identification. To achieve that objectives, this paper mainly do two things, firstly, create dataset with annotation, they name dataset as ”Offensive Language Identification Dataset ”(OLID), which is a new large scale dataset of tweets with annotations, the annotations in include keywords(for example, ”medical marijuana”, ”you are”) and so on. Step 2, Run 3 different algorithms based on the annotation dataset. The algorithms are: SVM, BiLSTM, CNN. The figure 2 describes the approach used by the author.

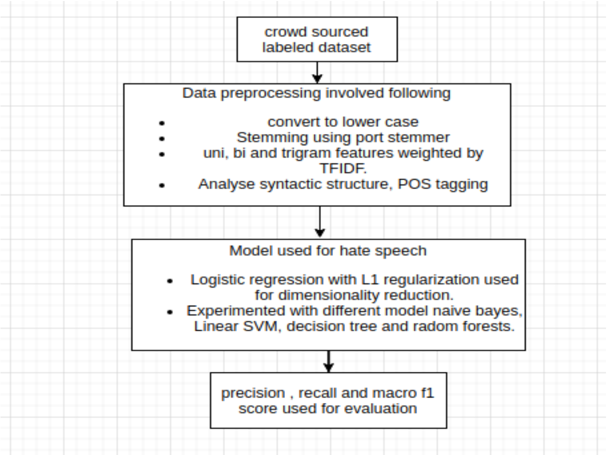

crowd sourced
labeled dataset
Data preprocessing involved following
• convert to lower case
• Stemming using port stemmer
• uni, bi and trigram features weighted by TFIDF.
• Analyse syntactic structure, POS tagging
Model used for hate speech
• Logistic regression with L1 regularization used for dimensionality reduction.
• Experimented with different model naive bayes, Linear SVM, decision tree and radom forests.
precision , recall and macro f1 score used for evaluation

Fig. 1. Flow chart of research approach.

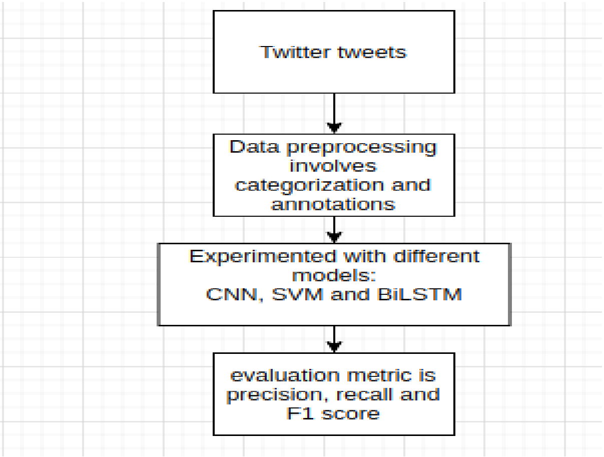

Twitter tweets
Data preprocessing involves categorization and annotations
Experimented with different models:
CNN, SVM and BiLSTM
evaluation metric is precision, recall and F1 score

Fig. 2. Flow chart of research approach.

- The idea of this paper [5] is to design a system which can prevent hate speech, the way how they did this can be concluded as 3 steps. Firstly, they collect the data from Reddit and Gab, which is different from most papers, the data will be a full conversation, so the researcher will have a context for samples. Step 2, the samples are manually labeled as hate or non hate speech by Mechanical Turk workers, and then all samples will be replied, the reply also will be recorded. Step 3 Generative Intervention, the method shown on our figure where c is the conversation, r is the corresponding intervention response, and D is the dataset.(Obj is the objective of generating response) Then based on the samples and replies,

$$Obj = \max \sum_{(c,r)\in D} \log p(r|c)$$

Fig. 3. Generative intervention method

they run four different machine learning algorithms to learn how to pick responses, the best performance is from CNN algorithm.

- The idea of this paper [6] is combine the text in image and text to rich our samples, and then the author will convert image and text information to different features, the image will generate 2048 features, text will generate 150 feature, but there two dataset for text, one is from image samples, another one is from Tweet text. Therefore, the system will get 2348 feature, the author plan to run a CNN model for those features, the model of design show as figure below.

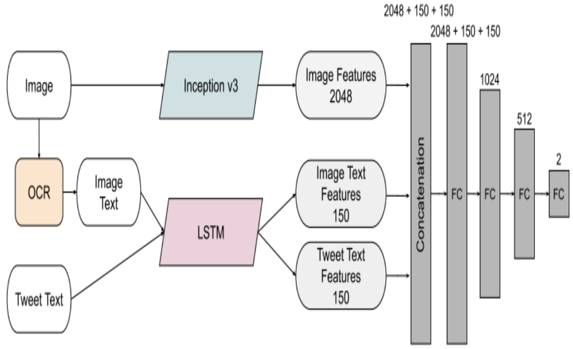


Fig. 4. Multimodel

- This paper [8] aims to detect hate speech and the type of hate speech from different languages, it has three different sub tasks on this paper. The first one is to simply check whether it is a hate speech content or non-offensive content, the second task is based on the task 1, if the content is belong to hate speech, the task 2 will identify the type of hate speech, and then this paper want to target(mark) the post of hate speech. The algorithm they used is LSTM, but on the method of evaluations, this paper introduces two ways: weighted F1 and Macro F1. The weighted F1 score calculates the F1 score for each class independently. When it adds them, it uses a weight based on the number of true labels of each class.The ’macro’ calculates the F1 separately for each class but does not use weights for the aggregation.
- This paper [12] is a kind survey report, it takes the data from Twitter, and throws out a comparative research topic: what is the difference between hate speech instigators and hate speech targets? To figure out the answer to this question, they extract some information from data: account characteristics(Gender, TimeZone, Invalid image), Personality Traits (Agreeableness, Conscientiousness, Extraversion, Emotional range, and Openness). The measure for account characteristics is simply count, but for the second one, the author of this paper used a pre trained model which is IBM Watson Personality API, this API describes scores [0,1] that reflect the normalized percentile score for the characteristic.
- This paper [14] aims to build a hate speech detection based on the lexicon sentiment. To get the lexicon sentiment, this paper describes this as three main steps, the first step is subjectivity detection, which will be done by isolating sentences that have subjective expressions from those normal sentences. In the second step, this paper builds a lexicon of hate related words with a rule based method using subjective features identified from the sentences and semantic features learned directly from the corpus. In the end this paper will build a classifier that uses the features from lexicon and use it to test the new document.
- This paper [15] aims to detect hate speech in social media like Twitter, the approach of this paper is about structure of sentence. Most existing efforts to measure hate speech require knowing the hate words or hate targets apriori. Compared to other measures, this paper provides a new way based on logic structure of sentences, which will consider a sentence from I, ¡intensity¿, ¡userintent¿, ¡hate target¿, for example I really hate Asian people. After this design they count the all used hate word from Twitter(this is because Twitter provides some related API). Finally, they will analyze the targets of hate speech on the internet.
- The core idea of this paper [16] is about how to build features from text for detecting hate speech content. Specially, this paper annotates samples by two methods: firstly, they ask Amazon Mechanical Turk (MTurk) workers to annotate these posts to cover three facets. In addition to classifying each post into hate, offensive, or normal speech, and they also select the target’s communities(like victims of speech); secondly, they use “rationales” to highlight parts of the text that could justify their classification decision. Then the author named this kind of sample as HateXplain. For algorithms, this paper runs the CNNGRU, BiRNN, BiRnnAttention (the difference with BiRNN is that BiRnnAttention includes the attention layer) and BERT pre trained model. The figure 5 describes the approach used by the author.

III. Problem Description and Formulation

Our key problem is to recognizing the textual content which includes hate speech. This paper is going to describe the problem and formulate based on three aspects: data(preprocessing problems), feature extraction (method to extract features) and

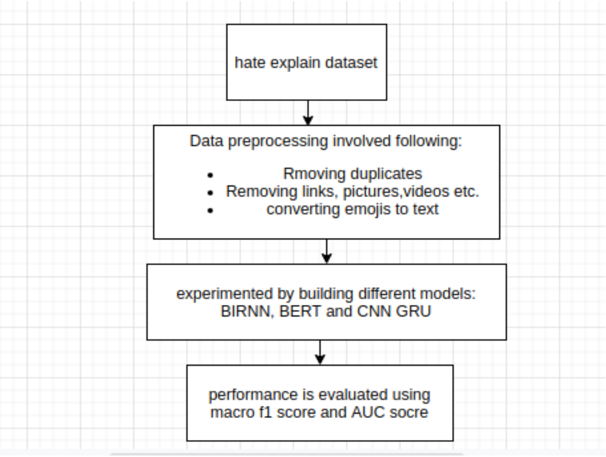

hate explain dataset
Data preprocessing involved following:
Rmoving duplicates
Removing links, pictures,videos etc.
converting emojis to text
experimented by building different models:
BIRNN, BERT and CNN GRU
performance is evaluated using
macro f1 score and AUC socre

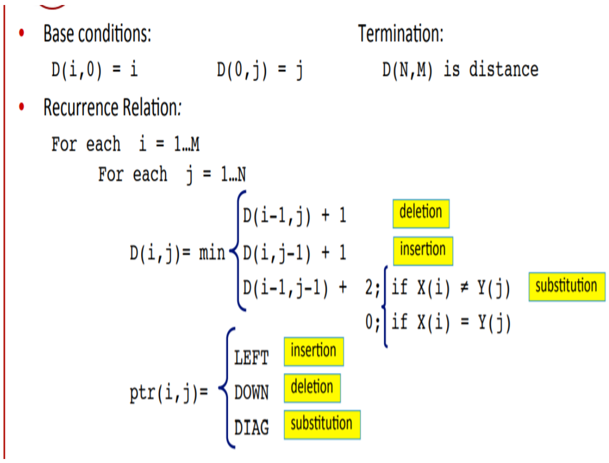
Fig. 5. Flow chart of research approach.

model(Design Problems).

- Data Problems: Firstly, our data is a huge dataset from the Internet, it may have following problems.
  - It may have some spelling mistakes, for example, this is normal for people to miss a word when people is typing, like “I like aple products, but there are some new… “;
  - There are many abbreviation words in our data set. This is a common thing for people, on the Internet, people would like to write some phrases like ASAP to represent as soon as possible.
  - Similarly to hate words, people also may use some phrases to rewrite the hate word.
  - Another problem we need to consider is the context of text. Sometimes people use the hate word, but the meaning behind the text is not going to attack someone, it could be their way to express their emotions.
  - Converting emojis, GIF, emoticons, images, numbers to text as people often reply or comment in emojis, emoticons or GIF instead of text.
- Data problems possible solutions: following are the solutions to the data problems.
  - We want to use the cosine similarity to fix the spelling problem. This is common for people generating the spelling mistakes while typing, but we can compare the wrong word with the right word (which will be a dictionary), if the similarity between two words are higher than 0.75, we will correct that word.
  - For the abbreviation problem, we want to build a dictionary of contractions to extend raw word to the original form, for example,if we found a word which is ASAP, then the program is going to check the dictionary or dataset, if a word like asap is found, then the program will replace asap with “as soon as possible” in the data.
  - For the context problem, we plan to build a word bag which is based on 1-gram, 2-gram, 3-gram. Then we can take part context into our word, but also we can build a vector which is based on the whole text, but we need to set the max length of the vector as well.
  - In real world there are nowadays emojis, emoticons, GIF, Images, Audio so we plan to use libraries like NLP pretext to extract text from emojis, pytesseract library to extract text from GIF and images, mapping of emoticons to word dictionary this dictionary will be used to convert emoticons to text, and a pretrained ready to use speech to text model for generating text from audio.
- Feature extraction problems: Its possible there can be few words which are not in the word embeddings additionally the length of the sentences will not be same.
- Solution to feature extraction problems: Individual character based encoding can be used to solve the out of vocabulary word problem, also a limit can be decided so length of all sentences are near to same.
- Model design problems: These are related to each step occupying the amount of time to execute and memory, design of encoding and decoding stages.
- Solution to model design problems: Optimization of model, parallel execution of process for this we will be using transformer models like GPT2 [28]/BERT [27] which are parallel and efficient. Design of decoding stage based on Top-k sampling, nucleus or temperature based sampling.

Given below we would like to explain few terminologies or methods or model that will be used in our planned approach.

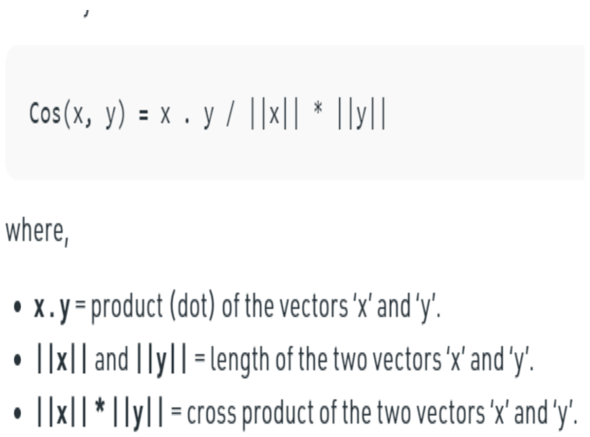
- Edit Distance: Yuan and matin mentioned this method in their paper [24], This algorithm is also called the Levenshtein distance, which is a number that explains how different two strings are. There are three technical operations used in this algorithm, Insertion, Deletion, Replacement(substitution), each time when we use the operations, the distance will increase by one.



```
• Base conditions:                              Termination:
   D(i,0) = i          D(0,j) = j                 D(N,M) is distance
• Recurrence Relation:
   For each  i = 1…M
         For each  j = 1…N
                       D(i-1,j) + 1           deletion
          D(i,j)= min  D(i,j-1) + 1           insertion
                       D(i-1,j-1) +   2; if X(i) ≠ Y(j)   substitution
                                      0; if X(i) = Y(j)
                      LEFT   insertion
          ptr(i,j)=   DOWN   deletion
                      DIAG   substitution
```

Fig. 6. Formula for edit distance [25].

- Cosine Similarity: According to the paper from Doctor Li [26], Cosine Similarity is a measurement that quantifies the similarity between two or more vectors. The cosine similarity is the cosine of the angle between vectors. The vectors are typically non-zero and are within an inner product space.



Cos(x, y) = x . y / ||x|| * ||y||

where,

- **x . y** = product (dot) of the vectors 'x' and 'y'.
- **||x||** and **||y||** = length of the two vectors 'x' and 'y'.
- **||x|| * ||y||** = cross product of the two vectors 'x' and 'y'.

Fig. 7. Formula for cosine similarity [25].

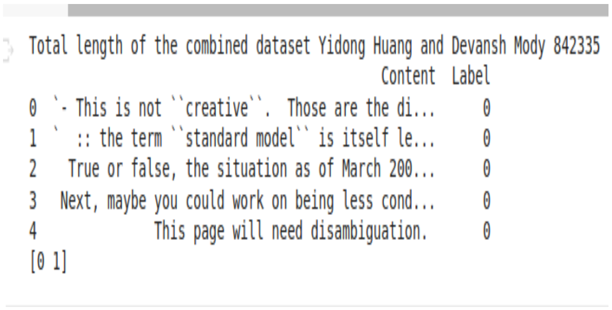
- Conceptual number batch embedding: The conceptual numberbatch word embeddings based on ConceptNet [22] are going to be used for the project. ConceptNet Numberbatch is a set of word direct forms, which can express word input in a formal way. Compared with other word vectors (word2vec, glove), the advantage of ConceptNet Numberbatch is that it uses both text and semi-structured information in ConceptNet for learning, so it can learn some semantics that may not be directly learned from general corpus. This method can help us to handle OOV(out of vocabulary) problem, therefore, we can improve our features of the dataset better.
- Thirdly, the design of our model, we plan to build a transformer model. To get a good performance, we read some papers, and we decide on many technologies. There are three key technologies we want to introduce and use in this paper.
  - BERT MODEL: The first one key technology is the BERT [27] model. The two tasks of the model training are to predict the words that are covered in the sentence and determine whether the two input sentences are upper and lower sentences. After the pre-trained BERT model is added to the corresponding network according to the specific task, the downstream tasks of NLP can be completed, such as text classification, machine translation, etc. Although BERT is based on the transformer, it only uses the encoder part of the transformer, and its overall frame is made up of multiple layers of transformer encoders. The encoder of each layer is composed of a layer of multi-head-attention and a layer of feed-forward. The large model has 24 layers with 16 attentions in each layer, and the small model has 12 layers with 12 attentions in each layer. The main function of each attention is to re-encode the target word through the correlation between the target word and all vocabulary in the sentence. Therefore, the calculation of each attention includes three steps: calculating the correlation between words, normalizing the correlation, and performing a weighted summation of the correlation and the encoding of all words to obtain the encoding of the target word. When calculating the correlation between words through attention, first, the input sequence vector (512*768) is linearly transformed through three weight matrices to generate three new sequence vectors of query, key and value, using each word The query vector is multiplied with the key vectors of all words in the sequence to obtain the correlation between words, and then this correlation is normalized by softmax, and the normalized weight and value are weighted and summed. Get the new code for each word.
  - Embedding Layer: According to the document of keras, the embedding layer must be set as the first layer of the net. The function of the Embedding layer is mainly to learn the distributed representation of words and reduce the dimensionality of extremely sparse one-hot encoded words. The sparsity of one-hot encoding determines that it is difficult to grasp the similarity between words, including part of speech, semantic information, etc. For example, 'car' and 'bus' are more or less similar, but one-hot encoding is difficult to measure. After embedding, a sparse vector of hundreds of thousands is mapped to a dense vector of hundreds of dimensions. Each feature of this dense vector can be considered to have practical meaning, such as singular and plural numbers, noun verbs, and so on. In the language model, words with similar characteristics and meanings often have similar surrounding words. Through the training of the language model, similar words often have similar dense vector representations. Therefore, the similarity between words can be simply used by the vector.
  - Encode and Decode: Encoder-Decoder is a very common model framework in deep learning. To be precise, Encoder-Decoder is not a specific model, but a kind of framework. The Encoder and Decoder part can be any text, voice, image, video data, and the model can use CNN, RNN, BiRNN, LSTM, GRU, etc. So based on Encoder-Decoder, we can design a variety of application algorithms. One of the most notable features of the Encoder-Decoder framework is that it is an End-to-End learning algorithm; this paper will take text-text examples as an introduction. Such models are often used in machine translation, such as translating French into English. Such a model is also called Sequence to Sequence learning.The

encoding is to convert the input sequence into a fixed-length vector; decoding is to convert the previously generated fixed vector into an output sequence. encode and decode can work well on the long sequence, but it also have some disadvantages.First one is that the semantic vector cannot fully represent the information of the entire sequence, and the other is that the information carried by the first input will be diluted by the information input later, or overwritten. The longer the input sequence, the more serious this phenomenon is.

IV. System Overview

To Solve the problem of hate speech detection we have proposed two approaches out of which plan A is what we intend to build it first then our plan B is to design a voting system so based on voting majority it can be decided weather the given text is hate speech or not.

- The data for hate speech detection is downloaded from the link https://hatespeechdata.com/, the given link contains links to data sets on different websites. Given below is the snapshot of the dataset. It can be seen in data set snapshot image it contains only two classes 0 and 1 where 0=not an hate speech and 1=hate speech. The size of the data set is 451709 samples.



```
Total length of the combined dataset Yidong Huang and Devansh Mody 842335
                                              Content  Label
0  `- This is not ``creative``.  Those are the di...      0
1  `  :: the term ``standard model`` is itself le...      0
2    True or false, the situation as of March 200...      0
3   Next, maybe you could work on being less cond...      0
4              This page will need disambiguation.      0
[0 1]
```

Fig. 8. Data set snapshot.

- After downloading the data following are the data preprocessing steps that will be applied.
  - converting text to lower case.
  - removing stop words.
  - reducing the number of repeated words and punctuation's.
  - expanding contractions.
  - tokenization
  - standardize and remove URLs and @usermentions
  - remove accented characters
  - correcting misspelled words using edit distance and cosine similarity
  - remove date time information.
  - using NLP pretext library to extract text from emojis
  - converting emoticons, emoji, GIF, images, numbers to text.
  - removing html tags and hyperlinks
  - removing space and unwanted special characters
  - correcting the grammar of the sentence using language correction tool.
  - Also we will incorporate hate speech detection on speech or audio by converting speech to text using pretrained model and for GIF or images we will use pytesseract library to convert GIF or images to text.
- After the data is preprocessed next step is to create a vocabulary of unique words with their index position.
- Additionally to pad the sentence with start and stop tokens.
- Make lengths of all the sentences of equal length.
- After the vocabulary is generated, the word embeddings for the given words are fetched from word embedding matrix. The conceptual numberbatch word embeddings based on ConceptNet are going to be used for the project. Even after completing data preprocessing there might be a case when there is an out of vocabulary word therefore to handle the problem of OOV we will use character based embeddings.
- Plan A in this approach we will replace the embedding layer with custom embedding layer, tune the model BERT/GPT2 model by adding the encoding and our own decoding stage based on the different decoding techniques like nucleus or temperature or topk based. Then classify the given sentence as hatespeech or not.
- Plan B approach employs a voting strategy by deciding based on the majority of the models, that is if three out of five models says its hate speech then classify the text as hate speech and vice versa. The five models that will be used are CNN, SVM, NB, BERT/GPT2, RF/DT/KNN.
- Once the model is build using approach A or B model will be evaluated using performance evaluators like ROUGE-L, BLEU, Recall and Precision.
- Also LIME [23] methods will be used to interpret the model reliability in real life and evaluate the performance.

Given below is the system overview diagram of our planned approach for implementing hate speech detection system.

V. Proposed Solutions and Comparison to the Literature

Our proposed solution aims to solve the various problems mentioned above in section problem description and formulation which are different and based on real life scenarios in comparison to the previous work done by researchers.

- Our solution aims to solve the problem of context and subjective analysis. which was not considered by the author [1].
- This size of data set was limitation for few researchers, we have a dataset of more than 800k samples which is huge enough for the research.
- Also limited data preprocessing techniques were used to clean and pre process data , we are going to work extensively on data pre processing like normalizing text which means converting everything from images, GIF,

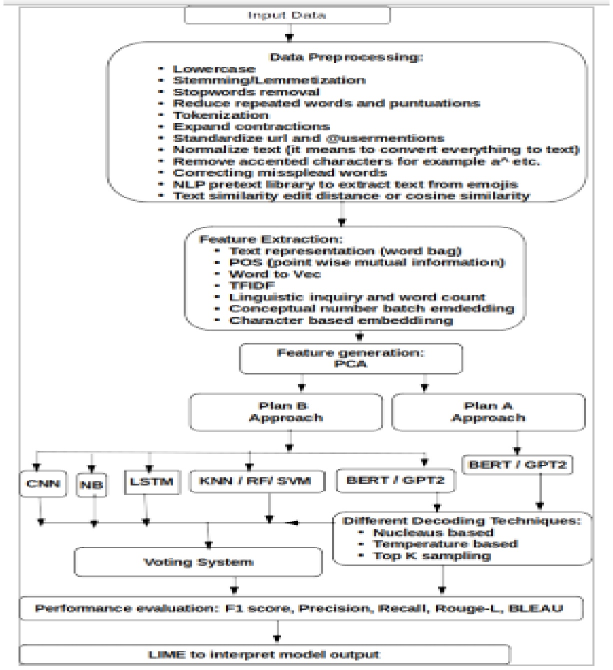

Input Data
Data Preprocessing:
Lowercase
Stemming/Lemmetization
Stopwords removal
Reduce repeated words and puntuations
Tokenization
Expand contractions
Standardize url and @usermentions
Normalize text (it means to convert everything to text)
Remove accented characters for example a^ etc.
Correcting missplead words
NLP pretext library to extract text from emojis
Text similarity edit distance or cosine similarity
Feature Extraction:
Text representation (word bag)
POS (point wise mutual information)
Word to Vec
TFIDF
Linguistic inquiry and word count
Conceptual number batch emdedding
Character based embeddinng
Feature generation:
PCA
Plan B
Approach
Plan A
Approach
CNN
NB
LSTM
KNN / RF/ SVM
BERT / GPT2
BERT / GPT2
Different Decoding Techniques:
Nucleaus based
Temperature based
Top K sampling
Voting System
Performance evaluation: F1 score, Precision, Recall, Rouge-L, BLEAU
LIME to interpret model output

Fig. 9. Overview of the system or our planned approach for the project

Audio, Emojis etc to text first, then finding misspelled words and correcting them based on similarity metrics like edit distance or cosine similarity, Expanding the contractions, converting to lowercase, removing stop words and white spaces, removing @usermentions and URLs, tokenization, remove accented characters, stemming/lemmatization, reducing repeated words and punctuation.

- The researcher have used either TF_IDF or Glove embedding but we will be using Conceptual Numberbatch embeddings.
- We will be using transformer model like BERT/GPT2 which preserves the meaning or context and its highly efficient model.
- Similar to other researcher's the models performance will be evaluated using various performance evaluation metrics like ROUGE-L, BLEU, F1, Precision and Recall.
- Moreover LIME based methods will be used to interpret and evaluate the model and to build the trust that our model is accurate and can be deployed in real time.
- The problem of out of vocabulary words is largely ignored by researchers in this area but we would like to solve the problem by character based embeddings which is a new idea to solve the OOV problem.
- So when compared to the research work done by the authors mentioned in the references below our proposed solution is advance based on our approach form preprocessing, to the use of embeddings and then using transformer model and interpreting the model performance in real time.

VI. Discussion

The most common problems faced by researchers in past were lack of training data, context analysis, building a universal model, subjective analysis, out of vocabulary words and mode architecture problems. Also the authors relied on traditional algorithms like random forests, KNN, SVM, Adaboost and decision trees for hate speech detection. Few authors tried LSTM, CNN but still they were not able to achieve a good output. Our approach is to use a more advance transformer model like BERT/GPT2 for our task of hate speech detection. A transformer is a deep learning model that adopts the mechanism of attention, by deferentially weighting the significance of each part of the input data. Transformer model have proved their mettle in language translation, text summarization and other tasks in NLP and computer vision. In our approach to hate speech detection lofty importance is given to the design of an effective preprocessing pipeline by considering all the possible real life scenarios in social media comments or on internet which can be in a form of audio, text, GIF and Image. So every thing audio, image, GIF, etc will be converted to text first, then sentences will be corrected if any spelling mistakes, expanding contractions, stemming/lemmatization, tokenization, converting to lowercase, removal of stop words and reducing repeated words and punctuation marks. Once data is cleaned exploratory analysis will be carried, features will be extracted, model will be build and then the model performance will be evaluated based on different scores like F1, Precision, Recall, ROUGE-L, BLEU and LIME methods to interpret and build trust in the model.

VII. Conclusion

Henceforth we would build an hate speech detection model on text, tweet and other short sentences. Moreover we would like to carry out an voluminous research analysis to understand the data, designing a effective data preprocessing pipeline, creating our own vocabulary, generating embeddings for given word, Architecture and developing of our Hate Speech detection model, evaluating and interpreting the real time performance using LIME methods. Our project has many real time application for social media platforms and government who are facing the problem of hate speech by anonymous user on Internet. Additionally also our project can be used as a preprocessing step to remove biases like hate speech before developing other projects like text summarization, text generation and other projects.

References

[1] Thomas Davidson, Dana Warmsley, Michael Macy and Ingmar Weber, ”Automated Hate Speech Detection and the Problem of Offensive Language”, arXiv:1703.04009v1 [cs.CL] 11 Mar 2017.

[2] Lei Gao, Ruihong Huang, ”Detecting Online Hate Speech Using Context Aware Models”, arXiv:1710.07395v2 [cs.CL] 22 May 2018.

[3] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra and Ritesh Kumar, ”Predicting the Type and Target of Offensive Posts in Social Media”, Proceedings of NAACL-HLT 2019.

[4] Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, Dit-Yan Yeung, ”Multilingual and Multi-Aspect Hate Speech Analysis”, arXiv:1908.11049v1 [cs.CL] 29 Aug 2019.

[5] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding and William Yang Wang, ”A Benchmark Dataset for Learning to Intervene in Online Hate Speech”, arXiv:1909.04251v1 [cs.CL] 10 Sep 2019.

[6] Raul Gomez, Jaume Gibert, Lluis Gomez and Dimosthenis Karatzas, ”Exploring Hate Speech Detection in Multimodal Publications”, Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV (2020) 1459-1467.

[7] Nina Bauwelinck, Gilles Jacobs, Veronique Hoste and Els Lefever, ”LT3 at SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter (hatEval)”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages (436–440).

[8] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia and Aditya Patel, ”Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages”, FIRE ’19: Proceedings of the 11th Forum for Information Retrieval Evaluation December (2019) Pages (14–17).

[9] Ona de Gibert, Naiara Perez, Aitor Garcıa-Pablos and Montse Cuadros, ”Hate Speech Dataset from a White Supremacy Forum”, arXiv:1809.04444 [cs.CL] 12 Sep 2018.

[10] Zeerak Waseem and Dirk Hovy, ”Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter”, Proceedings of NAACL-HLT (2016), pages (88–93).

[11] Elisabetta Fersini, Debora Nozza and Paolo Rosso, ”Overview of the Evalita 2018 Task on Automatic Misogyny Identification (AMI)”, CEUR Workshop Proceedings (2018).

[12] Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna and Elizabeth Belding, ”Peer to Peer Hate: Hate Speech Instigators and Their Targets”, arXiv:1804.04649v1 [cs.SI] 12 Apr 2018.

[13] Poletto F, Basile V, SanguBasileinetti M and Bosco C, ”Resources and benchmark corpora for hate speech detection: a systematic review”, Lang Resources Evaluation (2021).

[14] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien and Jun Long, ”A Lexicon-based Approach for Hate Speech Detection”, International Journal of Multimedia and Ubiquitous Engineering (2015) 10(4) 215-230.

[15] Silva, LMondal, MCorrea, DBenevenuto and FWeber, ”Analyzing the Targets of Hate in Online Social Media”, Proceedings of the 10th International Conference on Web and Social Media, ICWSM 2016 (2016) 687-690.

[16] Mathew B, Saha P, Yimam S, Biemann C, Goyal P and Mukherjee A, ”HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection”, ArXiv: 2012.10289 18 Dec 2020.

[17] Cao, RLee R and Hoang T, ”DeepHate: Hate Speech Detection via Multi-Faceted Text Representations”, WebSci 2020 - Proceedings of the 12th ACM Conference on Web Science (2020) 11-20.

[18] Marzieh Mozafari, Reza Farahbakhsh and Noël Crespi, ”Hate speech detection and racial bias mitigation in social media based on BERT model”, PLoS ONE 15(8): e0237861.

[19] Walker, Samuel. Hate speech: The history of an American controversy. U of Nebraska Press, 1994.

[20] Vedeler, Janikke Solstad, Terje Olsen, and John Eriksen. ”Hate speech harms: a social justice discussion of disabled Norwegians’ experiences.” Disability Society 34.3 (2019): 368-383.

[21] Consolidated acts, Criminal Code (R.S.C., 1985, c. C-46), Act current to 2021-09-22 and last amended on 2021-08-27, ”https://laws-lois.justice.gc.ca/eng/acts/c-46/section-319.html”.

[22] Robyn Speer and Joanna LowryDuda, ”ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge”, arXiv:1704.03560 [cs.CL].

[23] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, “Why Should I Trust You?: Explaining the Predictions of Any Classifier”, arXiv:1602.04938v3 [cs.LG] 9 Aug 2016.

[24] Yuan, Yihong, and Martin Raubal. ”Measuring similarity of mobile phone user trajectories–a Spatio-temporal Edit Distance method.” International Journal of Geographical Information Science 28.3 (2014): 496-520.

[25] https://web.stanford.edu/class/cs124/lec/med.pdf

[26] Li, Baoli, and Liping Han. ”Distance weighted cosine similarity measure for text classification.” International conference on intelligent data engineering and automated learning. Springer, Berlin, Heidelberg, 2013.

[27] Jacob Devlin, Ming Wei Chang, Kenton Lee, Kristina Toutanova, ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, Language Models are Unsupervised Multitask Learners, Open AI February 14, 2019.