CSE556_NLP_22_Project_H2

Sarthak Maini sarthak20576@iiitd.ac.in
Abhit Rana abhit20421@iiitd.ac.in
Saharsh Dev saharsh20572@iiitd.ac.in
Ashwin Tomer ashwin20289@iiitd.ac.in

1. Problem Definition

The task is a binary classification problem to classify the given dataset into two classes namely Hate Offensive tweets (HOF) and Non-Hate Offensive tweets (NOT).

Hate and Offensive (HOF) - This post contains Hate speech, profane, offensive content
Non-Hate-Offensive (NOT) - This post does not contain Hate speech, profane or, offensive content.
The task appeared as Subtask-A in HASOC 2021. The dataset taken is sampled from Twitter. It consists of twitter posts in Hindi and Hinglish language .it was curated based on the Offensive Language Identification Dataset (OLID) which contained posts in the English language

.

2. Related work

Initial works in the domain were based on a dictionary-based approach to identify hate speech. Several Researchers like Peter Burnap et al. (2015) [1] used an N-gram technique to create profane vectors from a list of profane phrases which were used in combination with a classifier like SVM to obtain a vector indicating the probability of hate.
After the development of non-contextual encoding, several researchers explored the possibility of applying the same in hate speech detection task.
Sreelakshmi K et al. (2020) [2] compared the performance of fastText with word2vec and doc2vec features for the Binary Classification task (hate/non-hate) whereas others like Jadhav et al. (2022) [3] applied a logistic regression base binary classification model which was later improved upon to be able to classify tweets into 3 desired classes.
With the rapid advancement of Deep learning architectures with respect to NLP tasks, more and more researchers started applying DL for almost every NLP task. Kamble et al. (2018) [4] trained typical DL Models (CNN, LSTM, BiLSTM etc.) for generating domain-specific word embeddings. Chopra et al. (2020) [5] used social network-based features (like author profiling) and targeted hate embeddings to train the model.
After the emergence of transformers, researchers started combining 2 tasks to achieve better performance.

Jahan et al. (2021) [6] performed Hate speech identification on text containing Hindi and Hinglish language. Hindi text was first translated into English while Hinglish was phonetically converted to Hindi and then synonyms were replaced into English. The model was trained on LR, CNN and BERT and gave significantly better results.

Sreelakshmi K et al. (2020) [2], observed that fastText features gave better feature representation with the Support Vector Machine (SVM)-Radial Basis Function (RBF) classifier. Jadhav et al. (2022) [3], observed that different models perform differently for different features for e.g., LR performs similarly for both BOW and TFIDF features but NB performs poorly when using Word2Vec embeddings. Kamble et al. (2018) [4] observed 12% increment in F1 scores when statistical classification was used as these embeddings incorporate the targeted groups while Chopra et al. (2020) [5] observed that a particular configuration of CNN, BiLSTM and Attention Layer together yielded the best results.

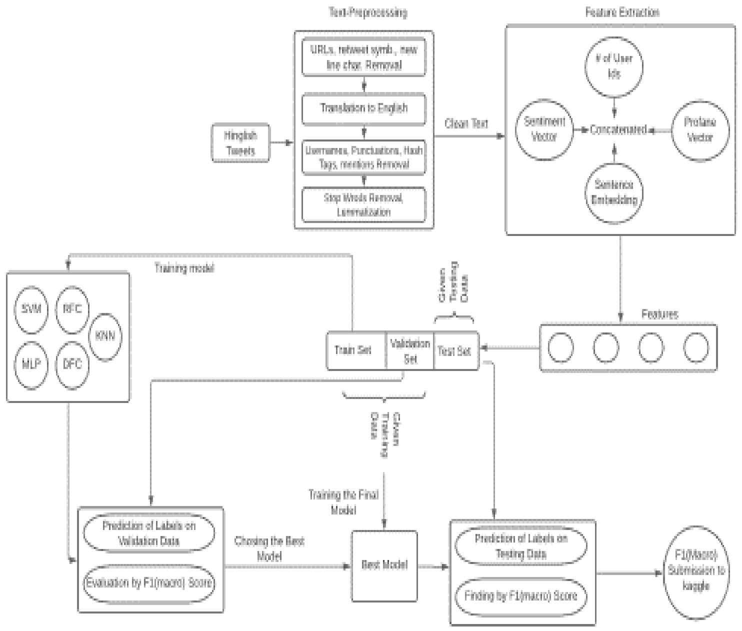
Jahan et al. (2021) [6] made an increment of almost 3% in accuracy and F1 score when compared to the baseline models.

Apparently, fine-tuning and combining different Deep Learning architectures has gained a lot of interest among the community. This has resulted in achieving even higher scores for tasks like hate speech detection in Hindi and other languages.

Study in [7] revolved around a fine-tuning approach for Hinglish language using the embedding such as ELMo(Embeddings for Language Models), FLAIR, and transformer-based BERT(Bidirectional Encoder Representations from Transformers). They demonstrated the use of BERT for English and Hinglish Embeddings using pre-trained weights and validated a new approach using ELMo and FLAIR which reached a F1 score of 0.94. A similar approach was used in [8] where they used character level embedding and then used a hybrid of Bi-LSTM and GRU with Attention Model. It came out on top when compared with more than 12 models experimented with and gave a maximum F1 score of 0.87.

Study in [9] proposed two ensemble models. First ensemble model was proposed using the basic ML models namely Logistic Regression, Random Forest, Support Vector Machine and Multinomial Naive Bayes which gave an F1 score of 0.847 using count vectorizer. Second proposed model consisted of CNN and Bi-Directional LSTM Model (DL Ensemble Stacked Model) and it gave an F1 score of 0.873.

3. Methodology



Dataset Used:
We used the dataset available for HASOC 2021 Subtask-A. The dataset contained Hindi and Hinglish posts sampled from Twitter.
It was inspired from the OLID dataset of English twitter posts.

Pre-processing:
The data then went through a series of processing steps. This is a crucial step as the raw data contains many errors and redundant information that can affect the model's training and output.

- The steps used in the pre-processing are:
- Removing all the web URLs that are in the tweets.
- Replacing the new line characters and RT/rt (retweets) with space.
- Translation to english Language
- Replacing all the @ mentions with username.
- Removing all the hashtags (#) from the tweets. Eg. #Modi → Modi.
- Removing all the numbers from the tweets.
- Converting all the tweets to lowercase.
- Stop words Removal

- A bit more cleaning by applying lemmatization on the tweets.

A profanity vector is generated for the cleaned tweet.

Removal of URLs was necessary because they do not play any role in the hate speech detection task. We used the googletrans api for translating the tweets in Hindi (Devanagari) to English. Earlier, we had used the itranslate library, but the performance of the model improved after using googletrans api. The usernames in the format @Name in tweets were replaced with "username" for the purpose of anonymity.

Feature Extraction

Usernames

The number of usernames in a tweet is taken as a feature as more no. of usernames in the tweet could mean more no. of individuals are being pointed out in the tweet which increases the probability of the tweet being hateful.

Profane Vector

A profane vector in the format [0, 0, 0…,0] is made where the length of the vector is equal to no. of profane words in Hinglish Profanity List csv. The vector for a particular tweet is generated by replacing the values in the null vector with the values from csv if a profane word is found in the tweet.

Sentiment vector

A sentiment array containing the sentiment component of the tweet (negative, positive, neutral, compound) is constructed. It is critical in determination of hateful speech since the sentiment of the tweet gives a lot of information about the tweet whether it contains hate or not. The VADER library was used for determination of the sentiment.

Contextual sentence embedding

In this type of sentence embedding there is information regarding the position of words in the sentence which makes this type of embedding as context dependent. For generating the sentence embedding, a sentence transformer having modules as base-language model, pooling layer and a dense layer is used. For base language model,'sentence-transformers/bert-base-nli-mean-tokens' model was loaded, fine-tuned and then it was used for the generation of the sentence embeddings

Model Building

A supervised Machine Learning Algorithm, Support Vector Machine (SVM) was chosen to build the final model. This model was chosen after performing several experiments with other classifiers as listed below in the experimental results.

4. Experimental Results

Hate specific neural architectures

Transformer	**RFC**	**MLP**	**SVM**
distilroberta-hatespeech	0.642	0.671	0.668
uncased-hatexpain	0.647	0.676	0.666
facebook/roberta	**0.657**	**0.697**	**0.697**

Fine-Tuned BERT

Architechture	**RFC**	**MLP**	**SVM**
1 layer 1epoch	**0.729**	**0.75**	**0.773**
4 layer 2 epoch	0.728	0.724	0.767

TF-IDF Vectorizer

Embedding	**DT**	**RFC**	**MLP**	**KNN**
TF-IDF	0.693	**0.762**	0.693	0.565

SVM gave the highest F1 score on all of the Hate specific architectures, and gave somewhat

comparable performance on Fine-Tuned BERT as well.
SVM gave the best performance in case of Fine-Tuned BERT model but its performance decreased in case of Hate specific architectures. The best performance while using the TF-IDF embedding was given by the Random Forest classifier

5. Analysis

- Using contextual embedding resulted in a higher F1 score than non-contextual embedding because contextual embedding provides context information that could help identify Hate speech tweets, even in the absence of profane words.

- Concatenation of contextual embeddings with more information like - the number of user ids, profane vector (listing the profane words and their values in the sentence), and sentiment information (positive, harmful, compound, neutral all together concatenated) of the sentence. This is because the added information gave the model better context about the scenario.

- We got more scores when we fine-tuned the pre-trained language models. This is because the pre-trained models are trained on representative data, which learns general information to generate sentence embeddings. Still, fine-tuning them based on our data would lead to learning the information relevant to the task in hand, a classification governed by the type of loss function we take for our pretraining or task.

- To get contextual embeddings, we take BatchAllTripletLoss because we are performing a classification task, so specifying the label of the class for a sentence would help in learning the sentence under the specified class.

6. Individual Contribution

TEXT PRPROCESSING:-
Saharsh Dev, Ashwin Tomer, and Abhit Rana

FEATURE EXTRACTION :-
Sarthak Maini, Abhit Rana, and Saharsh Dev

MODEL FORMATION :-
Sarthak Maini and Abhit Rana

7. References:

[1] P. Burnap, M. L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics, in: EPJ Data Science vol. 5, Springer, 2016 (pp. 1-15).

[2] Sreelakshmi K, Premjith B, Soman KP (2020) Detection of HATE Speech text in Hindi-English code-mixed data. Procedia Comput Sci. https://doi.org/10.1016/j.procs.2020.04.080

[3] Jadhav, Ishali & Kanade, Aditi & Waghmare, Vishesh & Chaudhari, Deptii. (2022). Hate and Offensive Speech Detection in Hindi Twitter Corpus. https://ceur-ws.org/Vol-3159/T1-33.pdf

[4] Kamble, Satyajit & Joshi, Aditya. (2018). Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. https://doi.org/10.48550/arXiv.1811.05145

[5] Chopra, Shivang & Sawhney, Ramit & Mathur, Puneet & Shah, Rajiv. (2020). Hindi-English Hate Speech Detection: Author Profiling, Debiasing, and Practical Perspectives. Proceedings of the AAAI Conference on Artificial Intelligence. https://doi.org/10.1609/aaai.v34i01.5374

[6] Jahan, Md Saroar & Oussalah, Mourad & Mim, Jhuma Kabir & Islam, Mominul(2021). Offensive Language Identification Using

Hindi-English Code-Mixed Tweets, and
Code-Mixed Data Augmentation
https://ceur-ws.org/Vol-3159/T1-23.pdf

[7] Srivastava, Ananya & Hasan, Mohammed & Bhargva, Yagnik & Walambe, Rahee & Kotecha, Ketan. (2021). Role of Artificial Intelligence in Detection of Hateful Speech for Hinglish Data on Social Media.
https://doi.org/10.48550/arXiv.2105.04913

[8] Rahul, V. Gupta, V. Sehra and Y. R. Vardhan, "Ensemble Based Hinglish Hate Speech Detection," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1800-1806, doi: 10.1109/ICICCS51141.2021.9432352

[9] Rahul, V. Gupta, V. Sehra and Y. R. Vardhan, "Hindi-English Code-Mixed Hate Speech Detection using Character Level Embeddings," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1112-1118, doi: 10.1109/ICCMC51019.2021.9418261.