Journal of Computer & Robotics Education Research (JOCRE)
Vol. 1, No. 1, 2024

Implementation of a Visualized Multi-Label Hate-speech Intensity model using Gradio

Bakwa Dunka Dirting
Department of Computer Science
Federal University of Technology, Owerri, Imo State, Nigeria
bakwadunka@gmail.com

Gloria A. Chukwudebe
Department of Electronic Engineering
Federal University of Technology, Owerri
Imo State, Nigeria
gloria.chukwudebe@ieee.co.ng

Euphemia C. Nwakorie
Department of Computer Science
Federal University of Technology, Owerri, Imo State, Nigeria
cenwokorie@gmail.com

Okpe Josephine O.
Department of Computer and Robotics Education
University of Nigeria, Nsukka
josephine.okpe@unn.edu.ng

Ikechukwu Ignatius Ayogu
Department of Computer Science
Federal University of Technology, Owerri. Imo State, Nigeria
ignatius.ayogu@futo.edu.ng

***Abstract*:** Past studies have contributed quite significantly to the detection of hate speech on social media platforms. However, most of such models stop at evaluating the models' efficacy. This has to be taken further with the provisions of the new ML UI tools to visually design an explainable predictive model with the plausibility of real-world usefulness beyond just accuracy. With this, classifying, scoring, and visualizing hate speeches from its mildest to aggressive form would be essential for the regulators of social networks and moderators of social platforms to be well-informed about the consequential scale of societal hate polarization flowing in the social space. UI visualization of hate speech provides high level warning signal that could go miles for actions that assuages the risk of potential violence. This study implements a user interface for determing the intensity of hate speech using Gradio and the well-known BERT model. BERT leverages the power of word contextualization and parallelization to reveal the most useful information from its key predictions. The proposed model was contextually efficient in differentiating between related hate comments through a simple interface. This study demonstrates a practically usable system to reliably rank the intensity of Hate-speech by their statistical probabilities.

Keywords: *Explainable Hate Speech, Hate Speech User Interface, Hate Speech Intensity, Multi-label Classification, BERT, Gradio.*

I. INTRODUCTION

Hate speech often starts with communications that go against the normal rules of communication in the social space and are capable of instigating conflict and disunity without well-grounded evidence. This may eventually build up to more harm. Researchers have perceived the vagaries of hating other people or groups based on some characteristics such as their nationality, political interest, religion, sexual orientation, color, or gender (Founta et al., 2018). For this reason, there is a dire need to develop a practically visualized model that could be used by social networks to visually analyze the context behind a social media comment and provide its ranked probabilities. This is helpful for moderation and decision making. This study is centered on the multilabel classification of hate speech to serve as a proxy for regulating comments that could aggravate and further as a warning signal rather than relying on manual word filters (Bahador, 2020; Idris, et al, 2020).

The ranking aspect in this work is a percentage-graded learning preference that is expected to reveal the most useful information from key predictions to map the training instances to the predefined sets of classes.

The key contributions of this paper are as follows:

1. development of a model for a contextual understanding of a piece of text using the profound BERT model.
2. development of a ranking algorithm to determine the intensities or severity of any arbitrary hate comment
3. implementation of a generalized and simplified user interface that shows the intensity of hate speech from any social media platform using Gradio platform.

According to Wulczyn et al.,(2017), more than half the number of Internet users witness one form of online harassment and anti-social behavior or other on social platforms. These negative behaviors include cyberbullying, gender discrimination, sexism, racism, etc. (Poletto et al., 2021). This could lead to the rise of violence, discrimination, communal clashes, suicide, and other forms of social injustice. The idea of model ranking, visualization and explainability is essential because the Artificial Intelligence that is provided to remove hate speech from social spaces has always been a blurry NLP task when trying to make the

computer understand the real meaning and context behind piecec of comments from social media. To take it further, the nature of informal communication introduces another level of complexity in text processing considering misspellings, linguistics diversities, emoticons, tones, stances, nuance, verbal and para-verbal attributes, and emotionality in addition to the subversive tactics employed by commenters. On the other hand, a limitation with traditional context-free representations assumes that the meaning of a word is rather constant across a document. For example BoWs, TF-IDF, or word2Vec embedding does not take cognizance of context-sensitive representations. These traditional word vector representations do not tend to capture higher-level textual information that might be more useful in a given piece of a sentence. Word embedding's capability of capturing semantic meanings of words is important. However, more understanding of higher-level concepts in addition to semantics such as polysemy, long-term dependencies, ambiguity, and negation needs to be considered in this context.

Most Deep learning algorithms are not designed for or have little capability to reliably learn deeper context (citation needed). Meanwhile, context is a significant consideration for effectively identifying hate speech. This paper relies on the original BERT model (Delvin et al., 2018) to explore contextual understanding of comments.

It is difficult to censor users' posts due to freedom of speech and the subjective nature of the English language. It remains significant to preserve the rights of free speech by individuals at the same time prevent a situation where others can leverage hate speech for nefarious purposes. Sometimes, what is referred to as hate speech could be perceived as freedom of speech by certain persons and vice-versa. This shows that context matters in the overall task of hate speech detection to clear doubts or bias.

It is important to develop an automated system with a sense of intelligence to understand what makes a piece of social media text hateful (Presant et al, 2019). Some comments come with ambiguous tweets on a topic that satisfies the writers' innate intentions. An automated system is an appropriate technique but they are yet to attain the human-level capability to logically estimate, discern or distill the real meaning behind individual words, phrases, and sentences. However, due to the huge volume of data, time overhead, and high cognitive load that the task of manual censorship imposes on human censors, the natural human ability to make errors increases significantly (citation needed). This makes the research presented in this paper important.

To illustrate how significant this work is, a distribution of the rate at which hate speech is removed online by Facebook is shown in Figure 1 (Statistica, 2022)

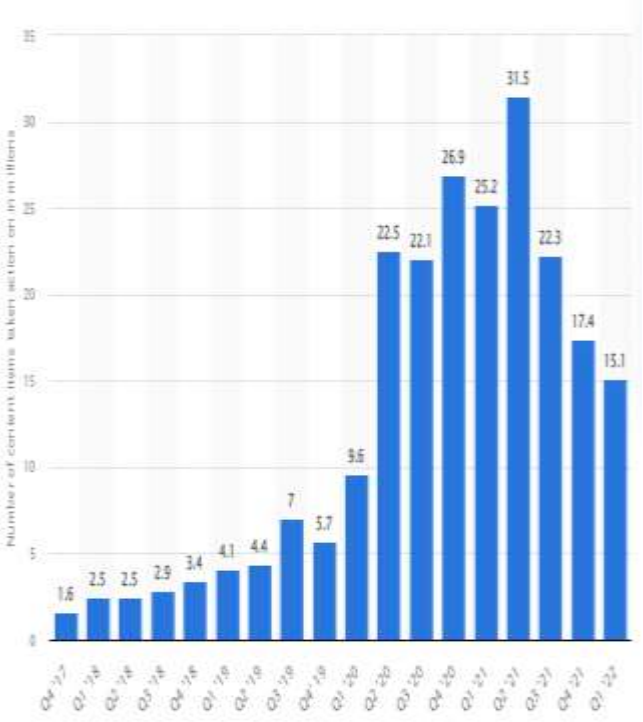


Figure 1: Hate speech containing contents removed by Facebook from 4th quarter 2017 to 1st quarter 2022 adapted from Statistica, 2022

The need for a robust dataset that work on various context was anchored on the expansive work of Bose et al., 2022). A huge number of people socialize on various social media platforms Figure 2. This generates a humongous amount of data from various platforms that has continued to incease and also prove challenging for manual approaches to filter hate speech.

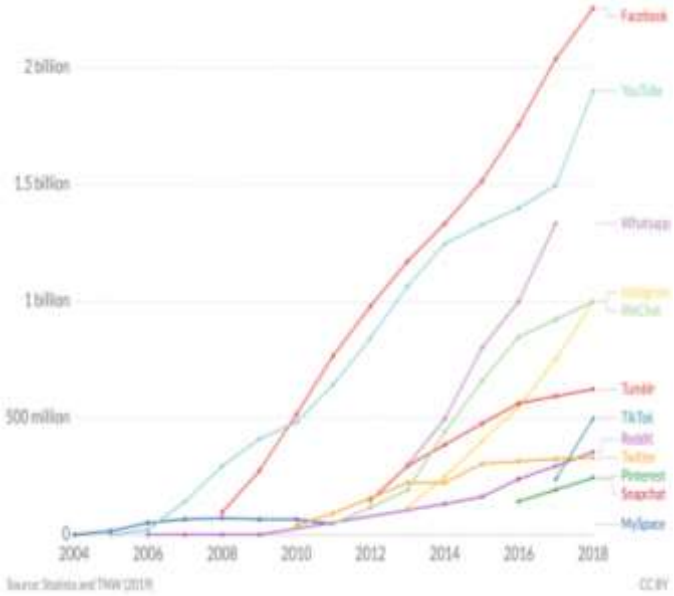


Figure 2: Social media growth trend over the years adapted from Statista and TNW, (2019)

The traditional problem of single-label classification is concerned with learning from examples, each associated with a single label λ_i from a finite set of disjoint labels (L) defined by $L = \{\lambda_1, \lambda_2, \ldots, \lambda_Q\}$, with $Q > 1$. For Q = 2, the learning problem is referred to as a binary classification problem, if $Q > 2$, the learning problem is referred to as a multi-class classification problem.

In the case of a multi-label classification task, a predictive model that correctly assigns a list of relevant labels to a given, previously unseen example is constructed. The goal is to find a function : $f: X \rightarrow (0,1)^L$, such that f maximizes the label space condition that returns a high predictive accuracy. If L is too large e.g. 1000, it is seen as an extreme multi-label classification problem due to the huge feature space (citation needed).

Deploying ML models is significant to provide a simplified user view for real-world understanding. This research has built a sharable web interface tool for the determination and presentation of the intensity of hate speech.

Current research has established that single-class classification methods are unable to adequately address the hate speech classification due to increasing rise of Big data. More fine-grained classification is needed to address the increasing complexity of hate classification. Therefore, the subjective and contextual nature of impolite occurrences in social media communication are better handled with multilabel learning. The study will also learn and provide the multi-label ranking to measure what is relevant or not.

II. LITERATURE REVIEW

Multilabel classification and ranking was pioneered by the works of Brinker and Hüllermeier, (2007), Fürnkranz, et al., (2008), Hüllermeier (2008), and Katakis, et al., (2008). Though these studies were not applied directly to hate speech detection, they laid the theoretical foundations for the research presented in this paper.

Bakwa et al, (2022) observed that there was a need to scale hate speech detection solutions to match the complexity of the growing social media data as well as correctly identify the correlations between hate labels. Research has shown that single-class classification cannot properly handle the relationships and correlation that exist among various hateful terminologies, distinctiveness due to the subjectiveness of hate, and contextual-dependent nature of impolite occurrences in social media communication.

A. Findings from the Review of Related literatures

Mozafari et al., Farahbakhsh and Crespi, (2019), proposed a transfer learning approach using the pre-trained BERT to work on a single-class classification of hate speech detection on a publicly available benchmark datasets. They proposed a new fine-tuning strategies to examine the effect of different embedding in the hate speech detection task. The experimental results of this study showed that using the pre-trained BERT model and fine-tuning it on the specific task will and by leveraging the BERT classifier's syntactical and contextual information outperforms previous works in terms of precision, recall, and F_1-score. Also, the results showed the ability of the model to detect biases. From their result, it was deduced that leveraging syntactical and contextual information, the BERT's model was able to detect the hatespeech but was not able to identify 'biases' in hate speech (Mozafari et al., 2019). This is a valuable to consider debiasing in hate speech in future studies.

Faizal, Muhammad and Indra, (2019), presented a hierarchical-multi-label-classification-API, that identifies some attributes such as the targets, groups, and levels of hate speech for Indonesian language using Twitter data. Random Forest Decision Tree (RFDT), Naïve Bayes (NB), and Support Vector Machine (SVM) classifiers were used along with term frequency features such as word and character level n-gram. They investigated five scenarios with different label hierarchies with very high accuracy. They discovered that the SVM algorithm and word unigram feature has the highest accuracy of 68.43%. The total data size was 31,634. From our analysis, the results of their studies can be replicated and improved upon using a deep learning algorithm that considers context and long-range dependencies.

Mwadime et al, (2020), utilized the BERT model to develop a model for hate speech detection on social media interaction. Their model produced an accuracy of 91.36%.

This proved to be a reliable model for automated hate speech detection with an F1 score of 0.81 recorded against 0.77 and 0.75 for logistic regression and random forests models. This represents a significant improvement in the model's performance.

Starosta, (2019) and Benk, (2019) worked separately on transfer learning and weak supervision in building new text classification models with minimum labeling cost. Their high precision classification model that combined weak supervision and transfer learnig was able to detect anti-semitic tweets based on their text, with no initial labeled data. They obtained a 95% precision and a 39% recall with less than 1,000 labeled examples. It was observed that building a large training set using weak supervision increases recall and precision by a margin of 0.27% and 0.5% respectively.

III. METHODOLOGY

A. *Methodology workflow*

This study focuses on detecting the intensity of hate speech using the BERT models by considering the context of words through a high-level features understanding (i.e. words contextualization and semantics). Figure 3 describes the methodology workflow implemented in this work.

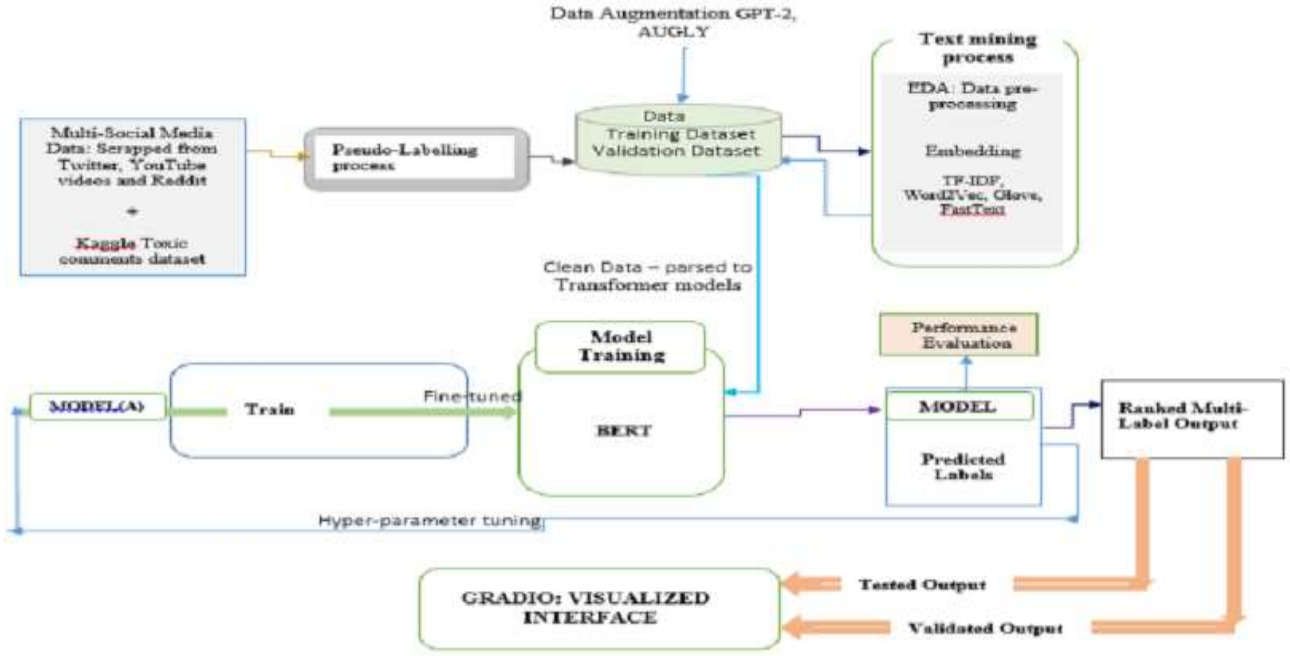


Figure 3: Methodology workflow

Experimental Setup

i. Datasets

Two datasets were utilized in this study. The first dataset, known as the Toxicity dataset, was developed specifically for this work. Additionally, a Multi-social media dataset. The Kaggle toxic dataset, which is publicly accessible, was obtained from comments and wikipedia Talk Pages. Kaggle enlisted the assistance of 5,000 crowd-workers to annotate this dataset. Each worker was tasked with rating the toxicity or neutrality of each Wikipedia comment. The toxicity labels within this dataset include toxic, severe toxic, obscene, threat, insult, and identity_hate. The Toxicity dataset encompasses 159,571 comments, consisting of over 24 million words and a vocabulary size of 495,147. It is important to note that this dataset is characterized by a high degree of imbalance, meaning that the number of comments for each class is not proportionate. Figure 4 visually represents the distribution of comments across the different classes, demonstrating that certain classes exhibit

significantly higher numbers compared to others.

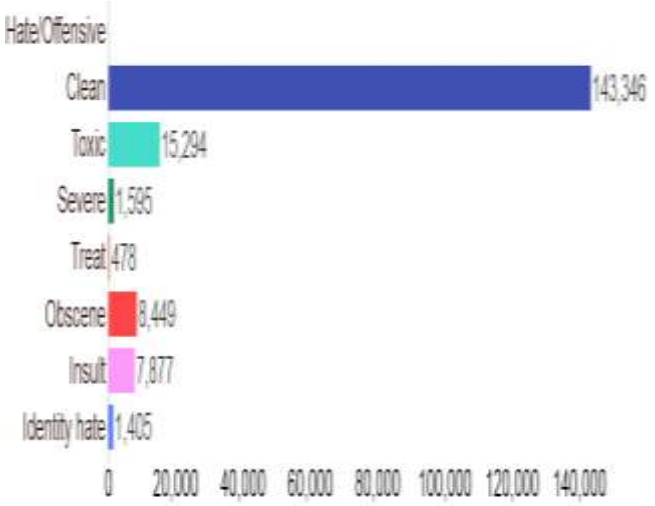


Figure 4: The distribution of the number of comments per class of the Kaggle Toxic Dataset.

The dataset provided by Kaggle comprises a total of 143,346 comments that have been thoroughly cleaned. Remarkably, the subset of comments categorized as hate speech accounts for approximately 80% of the entire dataset. Consequently, a significant issue arises due to the imbalanced distribution of classes within the dataset. To address this problem, it becomes imperative to employ data augmentation techniques in order to effectively increase the size of the dataset for the minority classes, such as Threat, severe toxic, and identity hate. As a result, synthetic hate speech data was generated to address this concern. In a broader sense, the process of data augmentation was meticulously applied, ensuring that the entire dataset was appropriately labeled and balanced prior to training, thus preventing any bias in the final evaluation.

In addition to the Kaggle dataset, a second dataset was incorporated in the experimental phase, which was derived from our previous research conducted by Bakwa et al. in 2022. This second dataset consisted of 30,000 comments that were collected and processed from three prominent social media platforms: Twitter, YouTube comments, and Reddit. The inclusion of these three datasets served two main purposes: firstly, to introduce diversity into the dataset, and secondly, to enhance the overall sample size. In order to effectively integrate this new multi-social media dataset with the hate portion of the Kaggle data,

ii. Hyperparameters

The set of optimal hyperparameters that were manually set to control the learning process and minimizes the loss function for the previous data are provided in Table 1. These parameters were learnt automatically without the programmers interference to tune the optimal search space or vectors for the best performance, maximum accuracy and minimum error.

Table 1: Hyperparameters used in the experiments

Parameters	**Hyperparameters experimented with**	**Best hyperparameters selected**
Choice of Model	BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters	BERT uncased,
Batch size	16, 32	32
Learning Rate	2e-5, 3e-5	3e-5
Optimization method	ADAMX, ADAM	ADAM
Sequence length	128	128
Epochs	1, 2, 3	2
Dropout	0.2, 0.3	0.2
Loss	Binary cross entropy (BCE)	Binary cross entropy (BCE)
Activation	Sigmoid	Sigmoid
Training Label shape	39,893	39,893

iii. Data Cleaning and Exploration

After acquiring the data from the three platforms the data went through the process of cleaning and preprocessing to normalize the dataset. Usernames, stop words, and other unnecessary characters were removed. The data was then lowercased, lemmatized and stemmed to gain more insight into the data. Exploratory data analysis were carried out before preparing into bar plots, heatmaps, and wordcloud distributions. For example, Figure 5, shows the wordcloud representation for “toxic comments (left) and clean comments (right)”.

Figure 5: Word cloud representation for Toxic (left) and Clean comments (right). The visual representations clearly shows toxic and a clean comment clearly being representatives of the wordclound representation.

B. Generation of synthetic data.

a. Generation of synthetic data using GPT-2.

Data augmentation (DA) technique was implemented in this study to add diversity and size to the dataset. Figure 8, shows some examples of the generated text from our GPT-2 model. Figure 8 are some examples of the generated text

Table 2.

Table 2: Examples of cleaned synthetic comments generated by GPT-2

#	**Examples of Augmented text generated by GPT-2**
1	Ï am a jew no you are a jews yeah I don't understand … oh god you believe me thatd suck',
2	Hey I know you like being a cunt but youre a whore And then Ive ..
3	'block is fake not my work petertodd I see what you mean by that ….'
4	Haha u should never have to be a slave to a bunch of white ass …
5	It gets u a real life nigga that will never get u a fucking job …
6	Is your boyfriend faggot Fuck off and now that I see you talking …
7	Ú are a shill for jews'
9	You are white, you are hore and horrible

b. Implementation of AUGLY Library for DA

The second method of augmentation used in this research is the AUGLY Python open-source library (facebook.com).

c. Implementing Pseudo-labeling for labels assignment

The semi-supervised pseudo-labeling flow is shown in Figure 6:

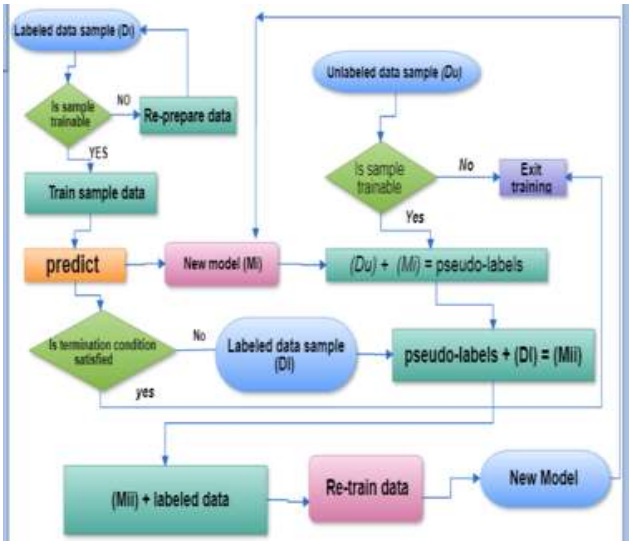


Figure 6: Pseudo-labelling workflow.

The fundamental idea in this workflow was to train the model with a BERT classification algorithm on the labeled Kaggle data (few data) first since it was already labeled adequately by experts, then we use the trained layers or knowledge to predict the labels for the newly generated and augmented samples that are unlabeled. The process is shown in Figure 9. However, it is worth pointing that, the predicted labels are those with a high probability confidence set at 50% threshold. consequently, the entre sample was automatically labeled. Pseudo-labeling relies on classes probabilities with the maximum predicted values of 1 for each unlabeled samples until a termination condition is attained (Dong-Hyun, 2013).

$$y_i' = \begin{cases} 1, \text{if i} = \text{argmax}_i,\ f_i(x) \\ 0, otherwise \end{cases} \quad \dots (1)$$

Pseudo-labels are used in fine-tuning with a dropout threshold. The loss function is given in Eqn. 1. It is equally important to give the differences in the number of labeled and unlabeled samples.

$$L = \frac{1}{n}\sum_{m=1}^{n}.\sum_{i=1}^{c}.L\,(y_i^m.f_i^m) + a(t)\cdot\left(\frac{1}{n'}\sum_{m=1}^{n'}.\sum_{i=1}^{c}.L\,(y_i'^m.f_i'^m)\right)\cdots (2)$$

Where n is the number of mini-batch in labeled data for the optimization scheme, n' is the unlabeled data. f_i^m is the output units of m, y_i^m is the labels of $f_i'^m$ for unlabeled data, $y_i'^m$ is the pseudo-label for unlabeled data, $a(t)$ is a coefficient balancing them. In other words, the unlabeled loss uses alpha to control the loss using weights as a function in the unlabeled data during each training epoch. This

happens for two reasons. First, the model initially focuses on the label data especially when its performance is not good. The weight gradually increases with the number of epochs over time, α(t), given in Eqn 2. Its parameters, t, T_1, T_2, α_f, are controllable hyperparameters based on the data, epoch, and model.

$$\alpha(t) = \begin{cases} 0 & t < T_1 \\ \frac{t - T_1}{T_2 - T_1}\alpha_f & T_1 \leq t < T_2 \\ \alpha_f & T_2 \leq t \end{cases} \quad \ldots (3)$$

C. Model Building: High-level implementation of the BERT model

The researcher ensured that the data fed into the raw architecture of BERT Based-cased is suitable based on the previous sections. BERT reads the entire sequence of input tokens of our test data in a non-directional manner.

Tokenization

The process of transforming each character into words internally, known as tokenization, can be exemplified using the openAI playground for example". christains do not deserve to live has 7 tokens and 33 characters. 43533, 1299, 466, 407, 10925, 284, 2107 coud represent the tokens where one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¾ of a word (so 100 tokens ~= 75 words).

Internally, these inputs combines 1) the token embedding from sequences of tokenized word piece with a hidden size is 768, (*SEQ_LEN* X 768). The token embedding also includes [CLS] and [SEP] markers which separates beginning and end of each sentence. 2) Positional embedding gives a representation of the position of each token in the sentence where each Nth row is a vector representation for the Nth position 3) Segment Embedding that identifies different unique sentences in the text. The proximity of positions between words is calculated using the positional equation 10. Tokens are passed to the *BertEmbeddings* layer for training using the *BertTokenizer*. After training the embedding layer, the vectors now has information about the similarity that exist within the vector space, thereby capturing the semantic similarity between words. A *LayerNorm* is applied to train the model faster and generalize better. Dropout is also applied to reduce overfitting and reduce the loss. An attention function helps to maps a query (Q) and a set of key (K) value (V) pairs to an output this is illustrated in the Architecture of the BERT Model.

This described the main internal benefits of the Multi-head attention that allows the model to jointly learn information from various sub-spaces or positions. This attribute is not possible with single attention head or previous models. An example of the internal representation of the Q, K and V relation as captured for a piece of sentence is illustrated in Figure 7

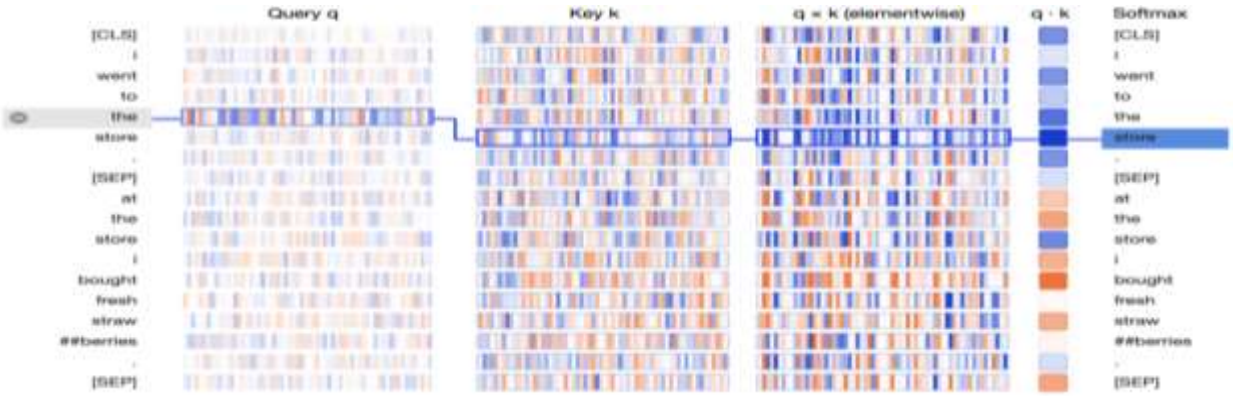


Figure 7.The internal working process of the Attention mechanism

Figure 7: describes the Attention mechanisms at a high level, it works by mapping an input sequence to a sequence of queries (Q), keys (K), and values (V). The query and key are used to calculate a "score" that determines how much attention should be given to each value. The values are then combined to generate an output, giving the model the ability

to "attend" to only the most important parts of the input. However, this phenomena present a challenge of efficiency in the transformer. Where each token representation is updated by attending to all other tokens at different layers. This incurs a computational quadratics time complexity of $O(n^2)$ that increases the training time. With multi-head attention a broader range of relationships between words are captured.

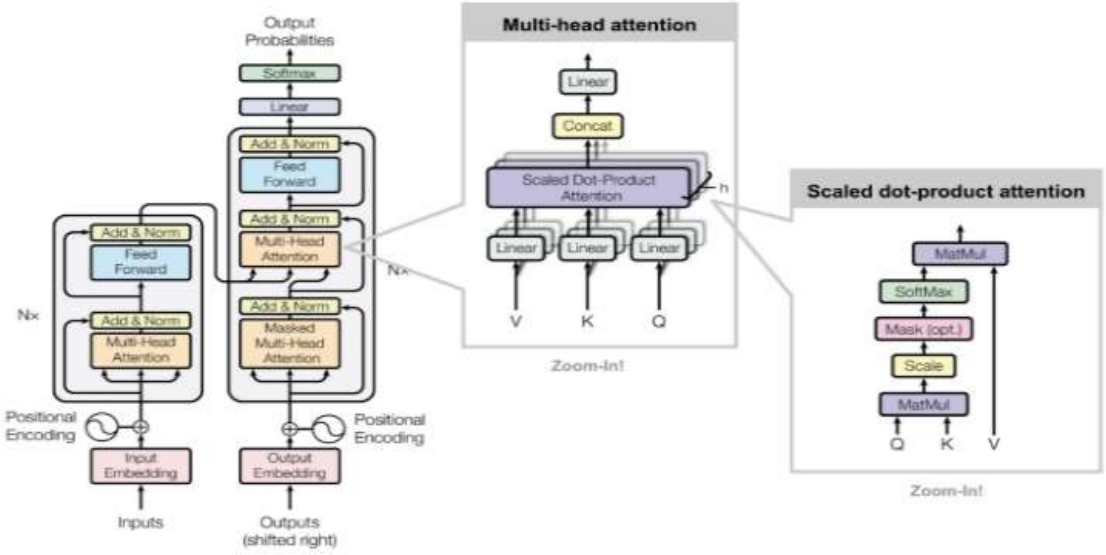


Figure 8*:* The full model architecture of the Transformer consisting of the *Scaled Dot-Product Attention (right) and a Multi-Head Attention (middle)* adapted from Fig 1 & 2 in [24].

Figure 8, shows that, embedding, go through different linear projections matrices that enables the Transformer to pick which components of the embedding is to be focused on or to pass further instructions to. This leads to a scalar products between the keys and the queries representing the relationships that matter by attending to 12 different "heads" of input in parallel. A commonly used attention function, with an additional scaling factor of $\frac{1}{\sqrt{d_k}}$ is given by:

$$Attention\,(\mathrm{Q},\mathrm{K},\mathrm{V}) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right).V \quad \ldots. \ 11$$

In Equation 11, the Scaled Dot-Product Attention computes the dot product of the query and the keys, scaled by the value of d_k and normalized by a softmax activation function to obtain the weights on the values. Instead of one attention function, the Transformer Network uses parallel attention functions with the output values concatenated and projected. The multi-head attention is defined by:

$$MultiHead\ (Q,K,V) = Concat\ (head_1,\ldots,head_h)W_0$$

Where $head_i = Attention\,(\mathrm{QW^k},\mathrm{KW^Q},\mathrm{VW^V})$, where the projections are parameters matric

D. Optimization algorithm

The performance of the optimization algorithm often affects the model's training efficiency. Therefore, proper parameterization of hyperparameters is important to improve the performance of the NLP task. Examples of optimization algorithms include Adam, Adagrad, Adamax, RMSprop, Nesterov, Rprop, etc. The training error is the error of the model calculated on the training dataset, while generalization error is the expectation of the model's error to an infinite stream of additional data drawn from the same distribution as the original sample. The overall essence of optimization is to improve the accuracy of the model on the test set where the end goal is to reduce the cost function.

1. Adam moment (Adam) optimizer

The Adaptive moment Optimizer (Adam) is a stochastic gradient optimizer that was presented by

Diederik Kingma from OpenAI and Jimmy Ba from the University of Toronto in their 2015 ICLR paper (poster) titled "Adam is a Method for Stochastic Optimization". The usefulness of their research has been demonstrated in practical applications. This optimizer uses pass gradients to calculate current gradients. It calculates the decaying average of the first moment (m_t) and second moment (v_t) of pass gradients. The main advantage of this optimizer is that: the hyperparameters such as learning rate, exponential decay rates for the moment estimate, etc. are automatically set to predefined values. Therefore, they do not need to be altered or tuned. As the default optimizer in ML, Adam performs a form of learning rate annealing with adaptive step size. It applies the first two moments of gradients (v and m) and calculates the average over them. Adam was designed to handle non-stationary of the objective function in RMSProp and tackles their sparse gradient limitation.

$$\theta^{\leftarrow}\theta - \frac{\alpha}{\sqrt{\hat{v}+\varepsilon}}\hat{m}$$

$$g_{it} = \nabla_\theta J(\theta_i, x_i, y_i)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_{i,t}$$

$$m_t = \beta_2 m_{t-1} + (1-\beta_2) g^2{}_{i,t} \quad \ldots 13$$

Where m_t is the first moment and v_t indicates the second moment that both are estimated, β_1 and β_2 are hyperparameters close to 1. Since the two estimates m_t and v_t have biases towards zero, the following estimates $\hat{m}_t$ and $\hat{v}_t$ are used

$$\hat{m}_t = \frac{M_t}{1-\beta_1^t},$$

and

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad \ldots 14$$

The parameter θ_i is updated with the following rules:

$$\theta_{t+1.i} = \theta_{t,i} - \frac{\eta}{\varepsilon+\sqrt{\hat{v}_t}}\hat{m}_t \quad \ldots 15$$

Varnishing learning rates for Adagrad are resolved in Adam. It performs well for most neural networks and sparse gradients.

Classification Losses

Hamming Loss

This measure computes the proportion of incorrectly predicted labels to the total number of labels.

For a multilabel classification, we compute the number of False Positives and False Negative per instance and then average it over the total number of training instances. The loss function uses Binary cross entropy, it increases as the predicted probability diverges from the actual label. It is expressed as:

$$Binary\ Cross\ Entropy\ Loss = -\sum.\left(y_i \log\ (y_i^{true}) + (1-y_i)\log\ (1-y_i^{predicted})\right) \ldots$$

E. Evaluation

α- Evaluation Score

One evaluation measure used in this work is the α-Evaluation score [25], used originally for Learning multi-label scene classification. It is a generalized version of Jaccard Similarity for evaluating each multi-label prediction. The α-evaluation score provides a flexible way to evaluate multi-label classification results for both aggressive as well as conservation tasks.

$$\alpha - \text{evaluation score} = \left(1 - \frac{|\beta M_x + \gamma F_x|}{|Y_x \vee P_x|}\right)^\alpha$$

$$(\alpha \geq 0, 0 \leq \beta, \gamma \leq 1, \beta = 1 | \gamma = 1)$$

Where,

$M_x \implies$ Number of Missed labels / False Negatives
$F_x \implies$ Number of False Positives
$Y_x \implies$ TP + FN
$P_x \implies$ TP + FP
$\vee \implies$ logical OR operator

F. Hyperparameters implementation

This work utilized the BERT-Base, uncased: 12-layers, 768 tokens, 12-attention-heads, 110M parameters. PyTorch framework for HuggingFace's transformer library for BERT-based model implementation, Gradio. All models were trained on Google Colab Pro's High-RAM environment using a single NVIDIA P100 GPU. BERT model were tuned on batch sizes of 32. All the models were fine-tuned for approximately 2 epochs and an early stopping to reduce overfitting. All other hyper-parameters were set to their default values based on HuggingFace's implementation. A random seed was set for all the experiments. The performance matrix used is the alpha Evaluation and the classification report. The Binary Cross-Entropy loss and Gaussian Error Linear Units (GELUs) activation function were used. The BERT models performed its best on a test set on batch size 32, with a learning rate of 3e-5 and 2 epochs, a sequence length of 128, weight decay of 0.01.

G. Implementation of Gradio

Visualization of machine learning models is very essential for developers to be able to get feedback from

other users and to explain how model and prototypes works in practical terms. Gradio is a relatively new library that enables the creation, deployment and sharing of ML models to the outside world. being more minimalistic, it enables fast deployment for Python package that are optimized with ML frameworks like PyTorch and TensorFlow using Python scripts. Gradio can be executed within a Colab or Jupiter notebook or script.

Very few research pay attention to putting up ML interface. In an attempt to deploy our multi-label hate speech solution, this research is set to deploy a customizable frontend UI for our model without worrying about scalability, CSS styling etc. The tool used in deploying this sharable web interface is the well-known Gradio, in the same category as other Hugging face, Tkinter etc. “Gradio allows you to quickly create customizable UI components around your TensorFlow or PyTorch models or even arbitrary Python functions”.

Gradio was used to simplifies the deployment process by providing an easy-to-use custom web interface to display the output of the trained model. An inputs function for the model was defined along with text attribute to be inputed to get predictions. The appearance of the interface and other controls was used to allow users to tweak the inputs hate text and see the effect on the output. Gradio was useful to handles all the behind-the-scenes work of serving the model over the web API to make it easy to share the application with others through a URL. Gradio is compatible with a wide range of ML frameworks and libraries, including TensorFlow, PyTorch, Scikit-Learn, and many others. It also supports a variety of input and output formats, such as images, audio, and text.

H. Result and impact of GPT-2 for Data augmentation

Since this study requires a high degree of language understanding and the potential impact on predictive models depends on the data, a data augmentation technique was implemented. The goal was to increase the quantity and quality of data for the minority classes, as well as the general training samples for the semi-supervised learning task. This technique proved useful in improving the model's performance by exposing it to a greater variety of hate language patterns.

Data augmentation (DA) was also important for the generalizability of the learning models. It provided additional samples of synthetic data, which helped the models learn new insights into hidden patterns. Additionally, DA had an impact on reducing variance and bias in the model, and it facilitated the generation of more diverse and creative text with minimal effort. The results are presented in Table 3.

Steps taken to develop and deploy the Gradio app

Gradio was installed using the command "pip install gradio." The necessary libraries were imported. Thereafter, the TensorFlow framework, which is ready to make predictions on new data, as well as the Gradio interface that includes the input and output fields, were implemented. The layout was designed with numerous features. The prediction function was defined to take the input data and use the learning model to make predictions. The function is called when a user interacts with the Gradio app. The deployment environment was configured for the environment variables, dependencies, and packages required to run on the HuggingFace deployment platform. The app was tested and deployed using Gradio's built-in sharing functionality. Other cloud platforms, such as Heroku, AWS, and GCP, could also be used to host the app.

IV. RESULTS AND DISCUSSION

'I am a Jew no you are a Jew you are a Jew yeah I dont understan
'oh god you dont believe me thatd suck',
'Hey I know you like being a cunt but youre a whore And then Ive
'block is fake not my work petertodd I see what you mean by that
'haha u should never have to be a slave to a bunch of white ass
'it gets u a real life nigga that will never get u a fucking job
'is your boyfriend faggot Fuck off and now that I see you talkin
'u are a shill for jews',

Table 3: Figure shows the results of the Data augmentation from the GPT-2.

I. Classification report

Table 4 displays the classification report, which includes the precision, recall, F1-score, Micro Average, Macro Average, and alpha-eval score. The model performs goes well for most labels, except for "identity hate," which has a significantly low prediction score.

Table 4: Table of the classification report along with te alpha evaluation scores

Label Category	Precision	Recall	F1-Score	Support	Alpha-evaluation
Toxic	**1.00**	**0.51**	**0.68**	**15276**	**0.92**
Severe_toxic	**0.94**	**0.83**	**0.89**	**7729**	**0.89**
Obscene	**0.99**	**0.59**	**0.74**	**12332**	**0.55**
Threat	**0.55**	**0.96**	**0.70**	**3582**	**0.86**
Insult	**0.91**	**0.70**	**0.79**	**8470**	**0.07**
Identity_hate	**0.04**	**0.44**	**0.07**	**395**	**0.88**
Micro avg	**0.95**	**0.91**	**0.93**	**44305**	
Macro avg	**0.91**	**0.80**	**0.83**	**44305**	
Weighted avg	**0.95**	**0.91**	**0.93**	**44305**	
Sample avg	**0.20**	**0.20**	**0.20**	**44305**	

The overall classification results indicate a mixed performance of the model in detecting hate speech from the multi-social media dataset based on the provided labels. The model achieves high precision values for categories such as "Toxic" (1.00), "Severe_toxic" (0.94), and "Obscene" (0.99), suggesting accurate identification of instances belonging to these categories. However, the precision for the "Identity_hate" category is extremely low (0.04), indicating a high number of false positives. This means that the model incorrectly classifies non-hate speech instances as "Identity_hate." In terms of recall, the model performs well for the "Threat" category (0.96), capturing a high proportion of actual instances. However, the recall for the "Toxic" category is relatively low (0.51), indicating that the model may miss some instances of hate speech within this category. The F1-scores provide a balanced measure of the model's performance, with higher scores for categories like "Severe_toxic" (0.89), "Obscene" (0.74), and "Insult" (0.79), indicating a good balance between precision and recall. On the other hand, the "Identity_hate" category has a significantly low F1-score (0.07), reflecting poor performance in both precision and recall. The support values reveal the distribution of instances across different categories, with higher numbers of instances in categories such as "Toxic" (15,276), "Obscene" (12,332), and "Insult" (8,470). The alpha-evaluation metric ranges from 0 to 1, with higher values for "Toxic" (0.92), "Severe_toxic" (0.89), and "Threat" (0.86), indicating relatively better model performance. However, the "Insult" category has a very low alpha-evaluation value (0.07), suggesting poor performance according to the specific evaluation criterion used. Overall, the model demonstrates effectiveness in detecting certain categories of hate speech, such as toxic, severe toxic, and obscene content, but shows limitations in accurately identifying instances related to identity-based hate speech. Based on the provided values, the support suggests that hate speech instances in categories such as "Toxic," "Obscene," and "Insult" are more prevalent in the multi-social media dataset. The model shows higher alpha scores for categories like "Toxic," "Severe_toxic," and "Threat," indicating better performance in identifying hate speech instances in these categories. However, the alpha score for the "Insult" category is very low, suggesting poor performance in detecting hate speech within that category. The model's overall performance in detecting hate speech, as indicated by the micro average precision, recall, and F1-score, is relatively good. The macro average scores, which consider each category equally, show room for improvement, particularly in certain categories. The weighted average scores, which take into account the distribution of instances in each category, highlight the model's good performance for prevalent categories. However, the sample average scores, regardless of the number of instances in each category, are low, indicating the need for improvement across all categories.

J. Results showing the ranking of the severity of hate speech Comments form social media comments

The severity of hate speech is shown in Figure 9.

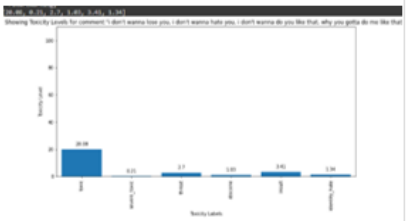
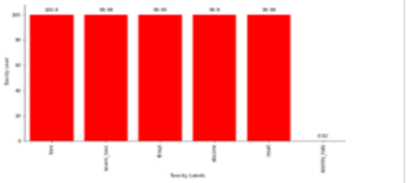
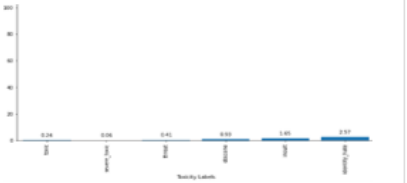
	Hate speech Intensity levels	Toxic comments	Severe Toxic comments	Threat comments	Obscene comments	Insult comments	Identity hate coments	Bar chat of intensity score
1.	"I don't wanna lose you, I don't wanna hate you. I don't wanna do you like that. Why you gota do me like that?"	20.08%	0.21%	**2.7%**	1.03%	3.41%	1.34%	
2.	"I ll kill you// summer walker is my jam"	100.0%	99.98%	99.90%	99.9%	99.98%	0.02%	
3.	"You look Good"	0.24%	0.6%	0.41%	0.93%	1.65%	2.57%	

Figure 9: Severity of hate speech comment

Figure 9 shows the ranking evaluation for the comment "i don't like you". The classifier, classified this comment as clean and this coincide with the human annotation or our notion of hatespeech that stipulates that merely outhering an honest negative opinion does not translate into hatespeech. For example, if the sentence were to be "i hate you", this system will flag this as hate speech. This also demonstrate a contextual understanding based on the learning from the dataset.

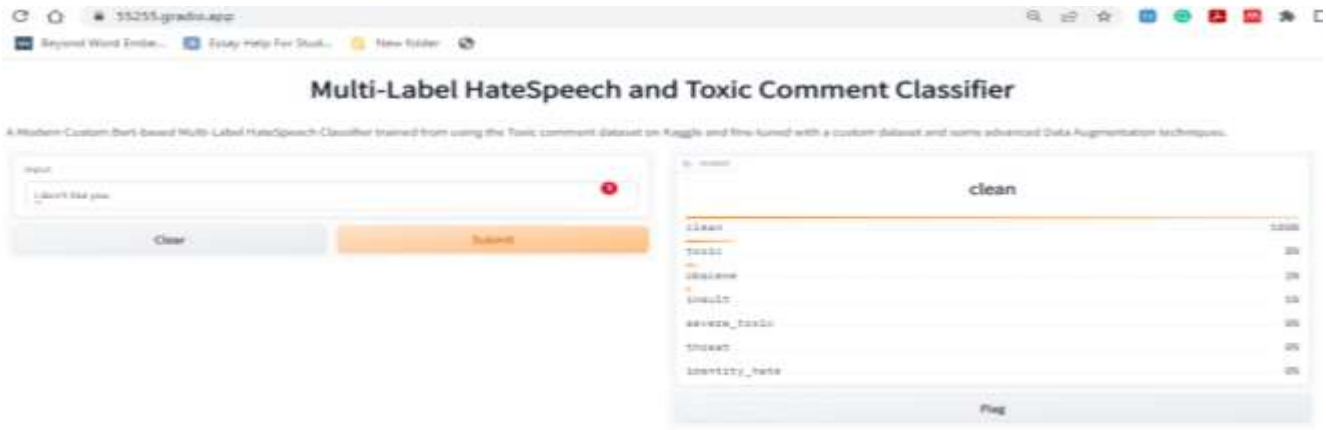


Figure 10: User Interface showing the visulised severity ranking outcome for a comment "I don't like you".

In our analysis, the comment "I don't like you" in Figure 10 was one of the comments predicted from our testing sample. The result indicates that the comment is clean since it expresses an honest opinion about someone. This demonstrates that the comment would not be flagged as hate speech, unlike statements such as "I hate you."

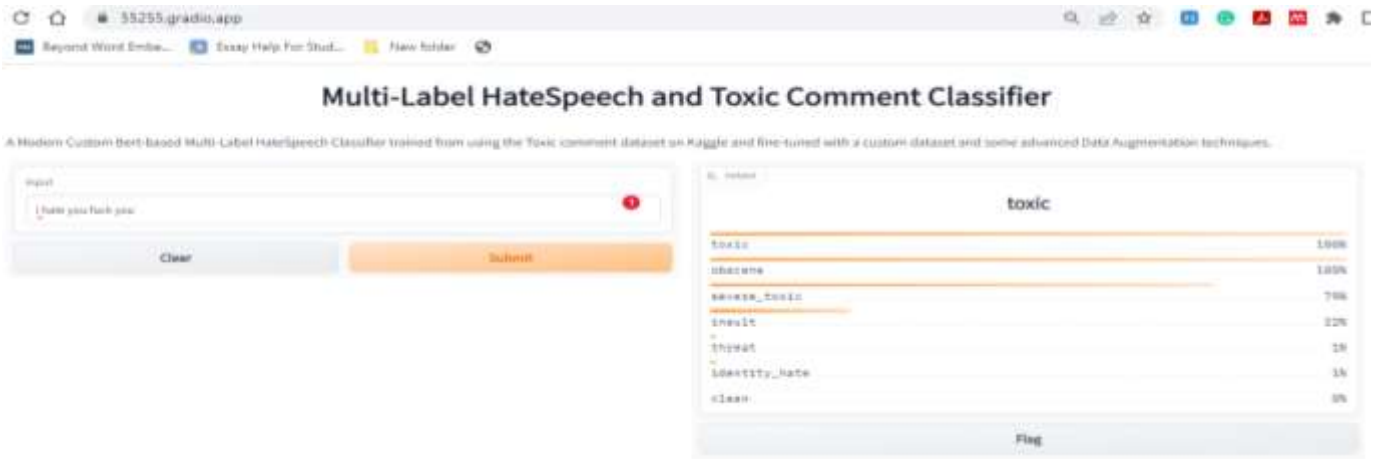
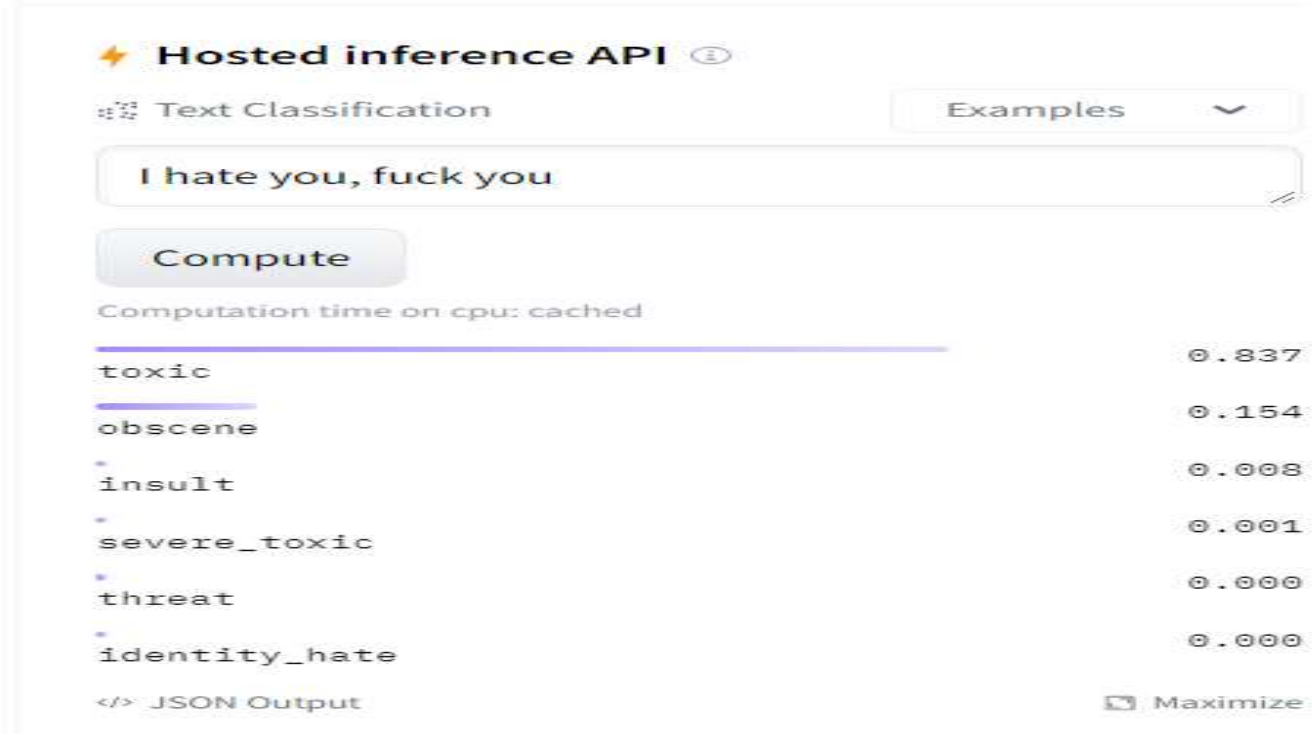


Figure 11: UI for the comment "I hate you, *uck you"

Contrary to the comment in Figure 10, the comment "I hate you, *uck you" in Figure 11 demonstrates a high intensity of hate, as indicated by the following results: toxic (100%), obscene (100%), severe toxic (79%), insult (22%), threat (1%), identity hate (1%), and clean comment (0%). The correlation among these classes highlights the strong hate intensity within the comment. Consequently, the comment should be flagged as hate speech due to the usage of offensive language. Comparatively, the results obtained from HuggingFace using Distilroberta for the same toxic comment are as follows: toxic (0.72), severe_toxic (0.38), obscene (0.72), threat (0.52), insult (0.69), identity_hate (0.60), and Macro-F1 (0.61). The API provides a visualization of these results.



Based on the evaluation provided by Distilroberta toxic detection model available at the URL (https://huggingface.co/jpcorb20/toxic-detector-distilroberta?), the result for the comment "I hate you, *uck you" is as follow: Macro-F1: 0.61. The score of 0.837, indicating a high likelihood of toxicity. The scores ranking of other categories provide insights into various aspects of the comment's.

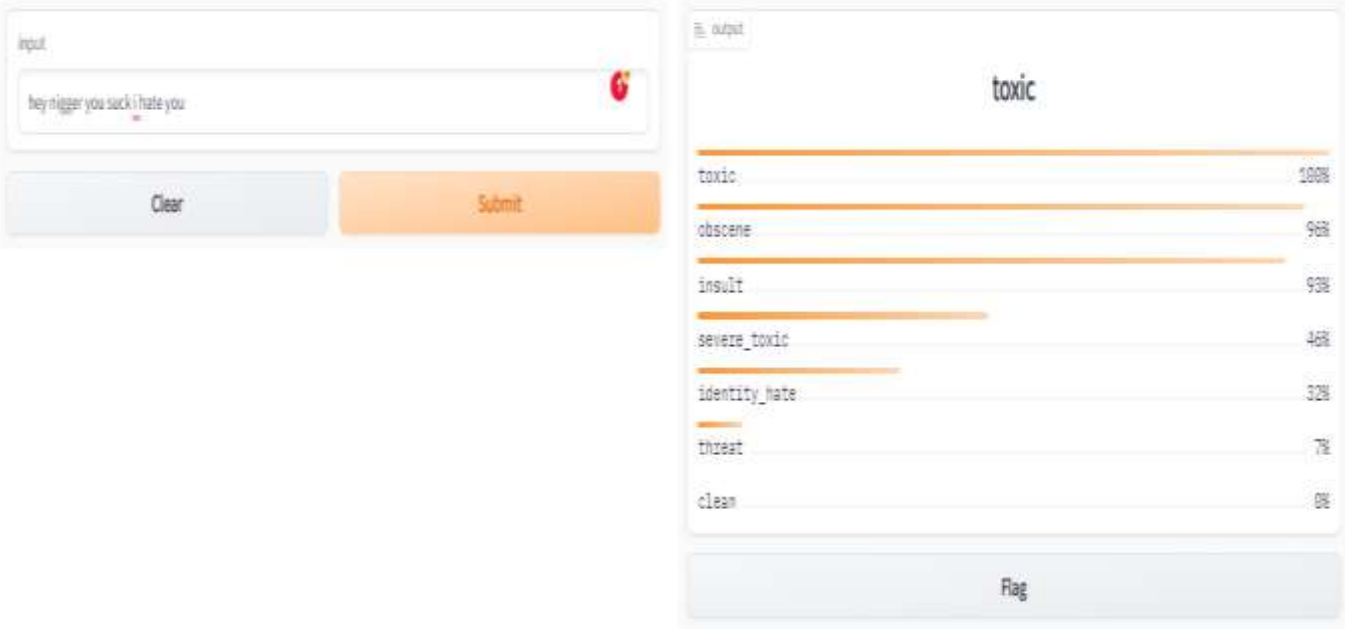


Figure 12. Result for “hey n*ger you su*k I hate you”.

The comment provided in Figure 12, "hey nigger, you suck I hate you," is perceived as hate speech with the following scores: toxic (100%), obscene (96%), insult (93%), severe toxic (46%), identity hate (32%), threat (7.5%), and clean (0%). Upon comparing these scores with the results obtained from the API available at the URL https://huggingface.co/jpcorb20/toxic-detector-distilroberta?, the following results are displayed: toxic (0.72), severe_toxic (0.38), obscene (0.72), threat (0.52), insult (0.69), and identity_hate (0.60). The visualization of these results is also provided. The comparison indicates that both the custom model and the API agree on the toxic, obscene, and insult labels, as they have high but dissimilar ranking scores. However, there are differences in the other labels, such as severe_toxic, threat, and identity_hate, where the custom model assigns higher scores compared to the API. It's important to note that these output may vary depending based of factors such as proper labelling of the training data or the algorithm or model used in the study

Discussion of Result

The study focused on evaluating the efficacy of BERT, a powerful language model, in detecting multi-label hate speech. Through the development of a ranking model and a user interface, the study aimed to simplify the process of hate speech detection and provide a tool to assist in moderating social hate content. The findings of this study demonstrate the effectiveness of BERT in identifying and classifying hate speech across multiple labels. By utilizing BERT, the study was able to develop a model that showed promising results in accurately detecting hate speech on social media platforms. The development of the user interface further enhanced the practicality of the study's findings. The interface provided a user-friendly tool that could indicate and assist in moderating social hate contents. This simplified approach to hate speech detection can be of great value for social media companies and individuals alike, as it empowers them to take active measures in combating hate speech and fostering a more inclusive online environment. However, it is important to acknowledge that the fight against hate speech on social media cannot rely solely on technological solutions. Social media companies must also take proactive steps to prevent the proliferation of hate speech on their platforms. This includes implementing robust content moderation

policies, investing in AI-based detection systems, and fostering a culture of inclusivity and respect among their user base. Individuals also play a crucial role in combatting hate speech. By being mindful of their own online behavior, promoting positive dialogue, and reporting instances of hate speech, individuals can contribute to creating a safer and more inclusive digital space. In conclusion, this study has highlighted the pressing issue of multi-label hate speech on social media platforms. By leveraging the power of BERT and developing a user interface for hate speech detection, the study has provided a valuable tool for addressing this issue. However, it is imperative for social media companies and individuals to work hand in hand to combat hate speech and create a more inclusive online environment, aligning with the 16th Sustainable Development Goal of promoting peace, justice, and strong institutions.

V. Conclusion

The aim of this paper to implementation a Visualized Multi-Label Hate-speech Intensity model using the BERT model and Gradio on social media data was achieved. This is accomplished along with a pseudo-labeling approach to overcoming the problem of data labeling for automatic labeling of our datasets to improving the model. Experimentally, our results showed that visualizing hate speech shows some level of explanability to justifying the output of the detection task. The statistical probabilities of each comment provides the ranks that indicates the severity of Hate speech of that comment

References

Bakwa, D.,D, Chukwudebe, G., A, Nwokorie, E., C, and Ikechukwu, I., A. (2022). Multi-Label Classification of Hate-speech Severity on Social Media using BERT Model. In Proceedings of the 2022 IEEE 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON) May 17-19TH, 2022.

Bahador, B (2020). Disinformation, Democracy, and Conflict Prevention Classifying and Identifying the Intensity of Hate Speech. web log post. trtrieve from https://items.ssrc.org/disinformation-democracy-and-conflict-prevention/classifying-and-identifying-the-intensity-of-hate-speech/

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017, May). Automated hate speech detection and the problem of offensive language. In Eleventh International AAAI Conference on Web and Social media (ICWSM), pages 512–515.

Jacob Devlin et al. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Mwadime, G., Odeo, M., Ngari, B., Mutuvi, S. (2020). Modeling Hate Speech Detection in Social Media Interactions Using BERT. 7(2). ISSN 2321–2705.

Mozafari M., Farahbakhsh R., & Crespi N. (2019) "A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media" in International Conference on Complex Networks and Their Applications. Springer, 2019, pp. 928–940.

Idris, David, Ogunseye, Elizabeth Oluyemisi and Akinola, Solomon Olalekan. (2020). Detecting Hate Speech on Social Media Using Deep Learning Techniques, University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR). Vol. 5 No. 1, pp. 22 – 38

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N. & Kumar, R. (2019). SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In Proceedings of the 13th International Workshop on Semantic Evaluation (pp. 75–86). Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Brinker, K., Hüllermeier, E.: Case-based multilabel ranking. In: Proceedings of20th International Joint Conference on Artificial Intelligence, IJCAI'07, pp. 702–707. Morgan Kaufmann (2007)

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K.: Multilabelclassification via calibrated label ranking. Mach. Learn. 73, 133–153 (2008)

Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learningpairwise preferences. Artif. Intell. 172(16), 1897–1916 (2008). MathSciNet (http://www.ams.org/mathscinet-getitem?mr=2459794)

Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification forautomated tag suggestion. In: Proceedings of European Conference on MachineLearning and Principles and Practice of

Knowledge Discovery in Databases, ECMLPKDD'08, pp. 75–83 (2008)

Jing, K., & Xu, J. (2019). A Survey on Neural Network Language Models. arXiv preprint arXiv:1906.03591.

Ruder, S. (2019). Neural Transfer Learning for Natural Language Processing (Ph.D. Thesis). National University of Ireland, Galway.

Salminen, J., Maximilian, H., Shammur, A., Chowdhury, S., J., Hind, A., and Bernard, J. (2020). Developing an online hate classifier for multiple social media platforms. https://doi.org/10.1186/s13673-019-0205-6. 10(1). pp. 1-34.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, 2013. 1, 2, 4, 6

Statistica, (2022). Global number of hate speech-containing content removed by Facebook from 4th quarter 2017 to 1st quarter 2022. Retrieved on August, 2022 from https://www.statista.com/statistics/1013804/facebook-hate-speech-content-deletion-quarter/

Poletto, F., Basile, V., Sanguinetti, M. et al. (2021). Resources and benchmark corpora for hate speech detection: a systematic review. Lang Resources & Evaluation 55, pp. 477–523. https://doi.org/10.1007/s10579-020-09502-8

Ibrohim, M.O., & Budi, I. (2019). Multi-label hate speech and abusive language detection in Indonesian Twitter. In Proceedings of the 3rd workshop on abusive language online (pp. 46–57).

Burnap, P. and Matthew L. W. (2015). Cyberhate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making. Policy and Internet, 7(2):223–242, 2015. doi:10.1002/poi3.85.

Wulczyn E, et al. (2017). Ex Machina: personal attacks are seen at scale. In: Proceedings of the 26th international conference on world wide web, Geneva; 2017. P. 1391–1399.

Maryam F., Alireza, D., Amin, H., F. (2018). Crime Data Mining, Threat Analysis, and Prediction. DOI: 10.35940/ijeat.A1768.109119. ISSN: 2249 – 8958, 9(1)

Presant, B., Mansfield, C., Nielsen, D., Mohan, P., Nayak, S., Longwill, B. (2019). Hate Speech Classification Using BERT LING 575: Group 2 "The Cool Kids". Idris, David, Ogunseye, Elizabeth Oluyemisi and Akinola, Solomon Olalekan. (2020). Detecting Hate Speech on Social Media Using Deep Learning Techniques, University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR). Vol. 5 No. 1, pp. 22 - 38.

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In Proceedings of the North American chapter of the association for computational linguistics: Human language technologies 2016, Association for Computational Linguistics (ACL) (pp. 88–93).

Gamback, B., & Sikdar, U. K. (2017). Using convolutional neural networks to classify hate speech. In Proceedings of the 1st Workshop on Abusive Language Online at ACL 2017. Vancouver, BC, Canada, 30 July-4 August 2017, pp. 85–90.

Rahman., M. (2020). Exploring Features for Multi-label Hate Speech Detection. Saarland University Department of Language Science and Technology (Master's Thesis)

PeaceTech Lab (2014). Social Media and Conflict in Nigeria, A Lexicon Of Hate Speech Terms ADL Education Division: Pyramid of Hate, available at http://www.adl.org/assets/pdf/educationoutreach/Pyramid-of-Hate.pdf.

Schmidt, A and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In International Workshop on Natural Language Processing for Social Media, pages 1–10. Association for Computational Linguistics, 2017. doi:10.18653/v1/W17-1101.

Gitari, N.,Zuping, Z., Damien, H., and Long, J. (2015). A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 10(10):215–230, doi:10.14257/ijmue.2015.10.4.21.

Devlin, J., Chang, M.,W, Lee, K., Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional

Transformers for Language Understanding. arXiv:181004805 [cs].

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SCIBERT: A pretrained language model for scientific text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing. 3615–3620.

Starosta, A. (2019). Building NLP Classifiers Cheaply With Transfer Learning and Weak Supervision. A Step-by-Step Guide for Building an Anti-Semitic Tweet Classifier. Retrieved on 7th January 2021 from https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15577251.pdf

Benk, M. (2019). Data Augmentation in Deep Learning for Hate Speech Detection in Lower Resource Settings. Unpublished Thesis. Master of Arts der Philosophischen Fakultat der Universit ¨ at Z ¨ urich. Retrieved on March 16th from https://www.cl.uzh.ch/dam/jcr:57406b34-02c8-496d-9b95-9968cee3a134/benk_ma_data_augmentation.pdf.

Mwadime, G., Odeo, M., Ngari, B., Mutuvi, S. (2020). Modeling Hate Speech Detection in Social Media Interactions Using BERT. 7(2). ISSN 2321–2705.

Salminen, J., Maximilian, H., Shammur, A., Chowdhury, S., J., Hind, A., and Bernard, J. (2020). Developing an online hate classifier for multiple social media platforms. https://doi.org/10.1186/s13673-019-0205-6. 10(1). pp. 1-34.

Mozafari M., Farahbakhsh R., & Crespi N. (2019) “A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media” in International Conference on Complex Networks and Their Applications. Springer, 2019, pp. 928–940.

Siyuan, L, (2018). Application of Recurrent Neural Networks In Toxic Comment Classification. UNIVERSITY OF CALIFORNIA Los Angeles [Master of Applied Statistics]. Chair, Professor Yingnian Wu.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8), p.9.

Bitton, J., and Papakipos, Z. (2021).AugLy: A data augmentations library for audio, image, text, and video. https://github.com/facebookresearch/AugLy, doi = 10.5281/zenodo.5014032.

Heereen Shim, Stijn Luca, Dietwig Lowet, and Bart Vanrumste. 2020. DataAugmentation and Semi-supervised Learning for, Deep Neural Networks-based Text Classfier. In The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic. ACM, Brno, Czech Republic, Article 4, 8 pages. https://doi.org/10.1145/3341105.3373992.

Boutell, M., R., Luo, J., Shen., Xipeng., Brown., M., C. (2004). Learning multi-label Scene Classification. Pattern recognition. 1757-1771.

Brinne, B. (2020). High-res 3D representation of a Transformer (BERT). [Web Log post]. Data science. Retrieved on June 30th, 2021 from hhtps://peltrion.com/blog/data-science/illustration-3d-BERT.

Bose, T., Aletras, N., Illina, I., and Fohr, D. (2022). Dynamically refined regularization for improving cross-corpora hate speech detection. In Findings of the Association for Computational Linguistics: ACL 2022, pages 372–382, Dublin, Ireland. Association for Computational Linguistics.

