Identifying Provocative Sentences from Fearful WhatsApp Messages

Ganning Xu
Teacher/Mentor: Mr. Robert Gotwals/Mr. Vaibhav Garg
Research in Computational Science
North Carolina School of Science and Mathematics
9 November 2022

Identifying Provocative Sentences from Fearful WhatsApp Messages

Ganning Xu, 1219 Broad St, Durham, NC 27705

Teacher/Mentor: Mr. Robert Gotwals/Mr. Vaibhav Garg

Research in Computational Science, North Carolina School of Science and Mathematics

9 November 2022

Abstract

In India, people often identify with a particular group based on religion and their native language. However, these same attributes are often used by politicians to provoke Hindus and Muslims against each other, such as when 3,000 Sikhs were murdered over a span of three days in 1984 during the anti-Sikh riots. The recent rise of social media has provided a convenient medium for provocative messages to spread. Despite these potential consequences of provocation, there has been no previous research on provocative sentence detection. By leveraging an existing dataset of WhatsApp messages collected from Indian political groups, three types of provocative sentences and one type of non-provocative sentence were identified. 7,000 sentences were manually annotated into one of the four categories based on provocation type. Next, data analysis was performed on the 7,000-sentence dataset, which revealed that provocative sentences were more negative, longer, and more toxic than non-provocative sentences. Through empath lexical analysis and LDA topic modeling, it was discovered that provocative sentences mention more negative categories and topics than non-provocative sentences. 18 machine learning methods were evaluated to determine the best-performing model for provocative sentence detection. 6 state-of-the-art transformer-based models (BERT, RoBERTa, XLM-RoBERTa, DistilBERT, XLNet, DeBERTa) were trained on the 7,000-sentence dataset. 12 combinations of non-transformer-based machine learning models (Multinomial Naive Bayes, Logistic Regression, Random Forest, LinearSVC) with various embedding systems (Word2vec, TF-IDF, Universal Sentence Encoder) were also trained and evaluated. Transformer-based models far outperform non-transformer-based ones. DeBERTa and DistilBERT were the best-performing transformer-based models, with both achieving 99.2% recall, 99.2% precision, and 99.2% F1-score. Out of the non-transformer-based approaches, TF-IDF with LinearSVC performs the best, with 89.2% recall, 89.4% precision, and 88.8% F1-score. All metrics were evaluated using 10-fold stratified cross-validation. With any of these models, social media companies such as WhatsApp can implement them into their apps, creating an efficient method for provocative sentence detection. All code used and the 7000-sentence dataset have been made publicly available for all researchers.

1. Introduction

In 1984, almost 3,000 Sikhs were murdered in a span of almost three days (Gill). This frightening scene occurred during the anti-Sikh riots, a classic example of how politicians can provoke people to act against a particular religious group (Singh). According to the U.S. Department of State's 2021 Report on International Religious Freedom, in India, most individuals associate themselves with either Hinduism or Islam, comprising 79% and 14% of the population, respectively (state.gov). These differences in religious groups lead to divisions that have often been leveraged to provoke Hindus and Muslims against each other (Verma).

In recent years, the emergence of online social media platforms has created a convenient medium to spread these provocative messages (Bhargava). Over the last decade, the number of users on social media has grown exponentially (Malik et al.). WhatsApp alone has two billion active users who send 100 billion messages daily (Halder et al.). India, which has the most WhatsApp users out of any country, has seen many issues with the spread of provocative messages on WhatsApp, with the government even intervening at times (Bhargava).

In 2021, Saha et al. studied how fearful WhatsApp messages create tension between Hindus and Muslims in India. While similar to hate speech, *fear speech* is defined as any expression that aims to instill a sense of fear in an ethnic or religious target group (Buyse). Saha et al. collected messages from over 5,000 Indian political groups on WhatsApp. Many of these messages utilized religion to provoke readers. Saha et al.'s final dataset contained 4,782 unique messages with 1,142 labeled as fear speech and 3,640 labeled as non-fear speech. Example 1 shows a fearful speech message that instills fear of the Muslim group in readers. The blue sentence proposes an imminent threat for readers in the Uttar Pradesh, Assam, and Kerala states, making the message fearful. The purple sentence creates a negative stereotype of Muslims, causing Muslim readers to be angered. Lastly, the red sentence urges the reader to share the message or be considered a Muslim. Clearly, the purpose of this message is to portray Muslims negatively.

Example 1: Fearful message containing many provocative sentences

"Leave chatting and read this message or else all your life will be left in chatting. In 1378, a part was separated from India, became an Islamic nation - named Iran . . . and now Uttar Pradesh, Assam and Kerala are on the verge of becoming an Islamic state . . . People who do love jihad is a Muslim. Those who think of ruining the country — Every single one of them is a Muslim !!!! Everyone who does not share this message forward should be a Muslim. . . ."

Through examination of fear speech messages in the 4,782-message dataset, it was discovered that many sentences in fear speech messages were provocative. Formally, I define *provocative sentences* as sentences that (1) cause readers to feel anger towards another religious group or (2) urge readers to carry out some action that may be harmful.

Readers are likely to be distressed and emotionally harmed by provocative sentences (Repple et al.).

While the younger generation is more adept at identifying dangerous content, older adults are the most vulnerable to misinformation and dangerous content in messages (Brashier and Schacter). When these older adults act upon the content within dangerous provocative sentences, violence, chaos, and riots are potential consequences.

Additionally, on social media platforms such as WhatsApp, moderation is limited to hate speech (Saha et al.). Thus, building a machine-learning model that can identify provocative sentences in messages is important. While there is existing work on identifying hate speech (Aluru et al.) (Zhang et al.) (Mathew et al.) and fear speech (Saha et al.), there has been no previous work done on identifying provocative sentences. The consequences of the lack of moderation in provocative sentences surfaced in 2018, when India's government intervened to stop provocative messages from being sent on WhatsApp (Bhargava). In order to bridge this gap in scientific knowledge, this research aims to create a new dataset of provocative sentences for researchers and a machine-learning model that accurately and efficiently identifies provocative sentences within fearful WhatsApp messages. Specifically, the following types of sentences from WhatsApp messages are identified:

(1) Provocation against religious culture: Sentences that attack or speak negatively about a religion's scriptures, religious practices, or religious leaders. Moreover, sentences that create stereotypes all people of that religion are also considered culturally provocative. These sentences are referred to as "culture".

(2) Provocation against religious oppression: Sentences that highlight the wrongdoings and oppression (could be real or fake) of a religious group. Examples include a certain group being portrayed as violent, dominant, or superior as compared to others. These sentences are referred to as "oppression".

(3) Provocation against religious action: Sentences that urge its readers to take some action against a religious group. These sentences might urge readers to carry out violence or to boycott that group. These sentences are referred to as "action".

(4) Non-provocative sentences: Sentences that are not provocative do not fall into any other category. These sentences are referred to as "none".

The relationship between hate speech, fear speech, and provocative sentences are shown in Figure 1.

An example of each type of provocation is shown in Example 2.

In order to fill the gap of provocative sentence research, and with the understanding that many provocative sentences are in fearful WhatsApp messages, the 1,142 fear speech messages from Saha et al. were utilized to produce the three main contributions of our research:

1. A **manually annotated dataset of 7,000 provocative and non-provocative sentences**, labeled into the categories of action, culture, oppression, and none.

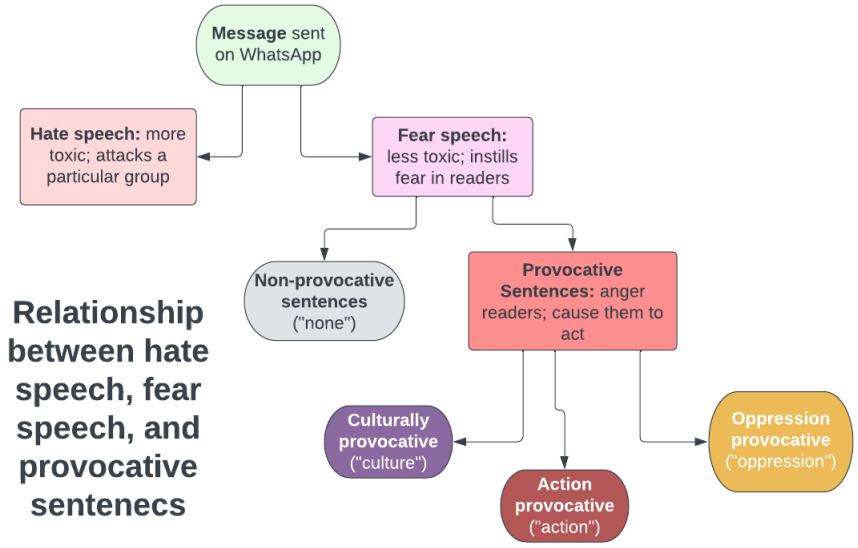

Figure 1: Fear speech and hate speech are different. Provocative and non-provocative sentences are sub-categories of fear speech. The three types of provocative sentences (action, culture, oppression) are sub-categories under provocative sentences. *Graphic created by author.*

Example 2: Types of provocative sentences

Religious Culture: "Muslims always blame the countries where they are happy .....!" (This sentence creates a negative stereotype about Muslims and their tendencies to blame others)

Religious Oppression: "2 years ago, 400 people were killed by Muslim people in a mall in Kenya." (This sentence explains the wrongdoings of Muslims two years ago)

Religious Action: "Hindu society should stop worshiping these tombstones, tombs, pirs." (This sentence urges Hindus to stop doing something, which may be against their will)

2. **Data analysis** on the textual differences between each category of provocative sentences, including sentiment, wordclouds, toxicity, empath (lexical analysis), and topic modeling.

3. A state-of-the-art performing **machine learning model to accurately detect and classify provocative sentences**.

2. Computational Tools

For all parts of our research, Python in Jupyter Notebooks was run using Google Colab. Various Python libraries were also utilized for data analysis and model development.

- **Data Collection & Annotation:** Python was used to pull the original dataset with the methods described in the "Data Collection" section. Google Sheets was used for dataset annotation.

- **Data Analysis:** When analyzing textual differences between provocative and non-provocative sentences, Natural Language Toolkit (NLTK), wordcloud, Google's Perspective API, empath, Matplotlib, Seaborn, and gensim was used.
- **Model Development:** When creating a machine learning model to classify types of provocative sentences, Simple Transformers and scikit-learn were used.

3. Data Collection

Using the dataset collected from "Short is the road that leads from fear to hate" (Saha et al.), a sentence-level dataset was created. The dataset from Saha et al. was obtained from 5,000 WhatsApp groups relating to politicians and political parties from August 2018 to August 2019, thus including major political events such as the National Elections and the major terrorist attack on Indian soldiers. The authors of Saha et al. then filter out spam and non-English messages to create a dataset of 4,782 WhatsApp messages. Then, they hired annotators from Amazon Mechanical Turk to annotate each message into either fear speech or non-fear speech, producing a final dataset of 1,142 fear speech and 3,640 non-fear speech messages.

When examining the dataset, it was clear that many fear speech messages contained provocative sentences. Thus, in order to create a sentence-level dataset, fear speech messages must be split into individual sentences. However, the context of each sentence within a message cannot be maintained when looking at individual sentences (S). Without the context of each sentence within a message, sentence annotations may be inaccurate, as factors such as sarcasm and jokes would be challenging to detect. Thus, the context of each sentence was preserved by keeping the sentence before (S_{before}) and the sentence after (S_{after}) within the message. When the selected sentence is at the beginning of a fearful message, S_{before} is an empty string. Similarly, if a sentence is at the end of a fearful message, S_{after} is an empty string. This process is shown in Figure 2.

Utilizing this method, all fearful WhatsApp messages were split into individual sentences while preserving the context of each sentence. Next, these rows were appended together to form one final dataset. Our final dataset contained 25,468 sentences, each with context.

4. Dataset Annotation

In the 25k sentence-level dataset, there were two main categories that these sentences fell under: provocative and non-provocative. Recall that provocative sentences caused readers to feel angry/annoyed or motivated readers to carry out some harmful action. Additionally, provocative sentences appeared to show three main themes: sentences that tell the reader to do something (action), sentences that attack one's culture (culture), and sentences that talk about a group's wrongdoings/dominance (oppression). Thus, provocative sentences were divided into these three categories: action, culture, and oppression. These categories can be summarized

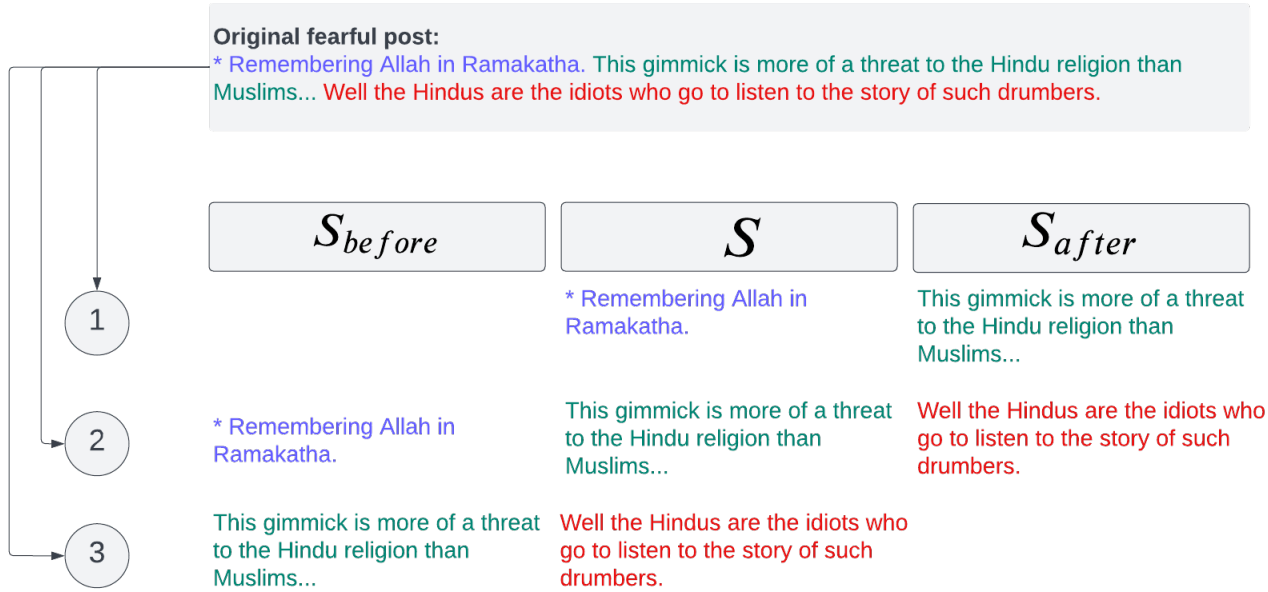

Figure 2: Fearful message being split into sentences. Since this message had three sentences, three rows were produced. Each of the 1,142 fearful messages would produce rows similar to this. These rows are then all appended together to produce the final dataset. *Graphic created by author.*

in Figure 3. When annotating each sentence into a category, S_{before} and S were considered to utilize context. To annotate each sentence, the procedures outlined in Figure 4 were followed.

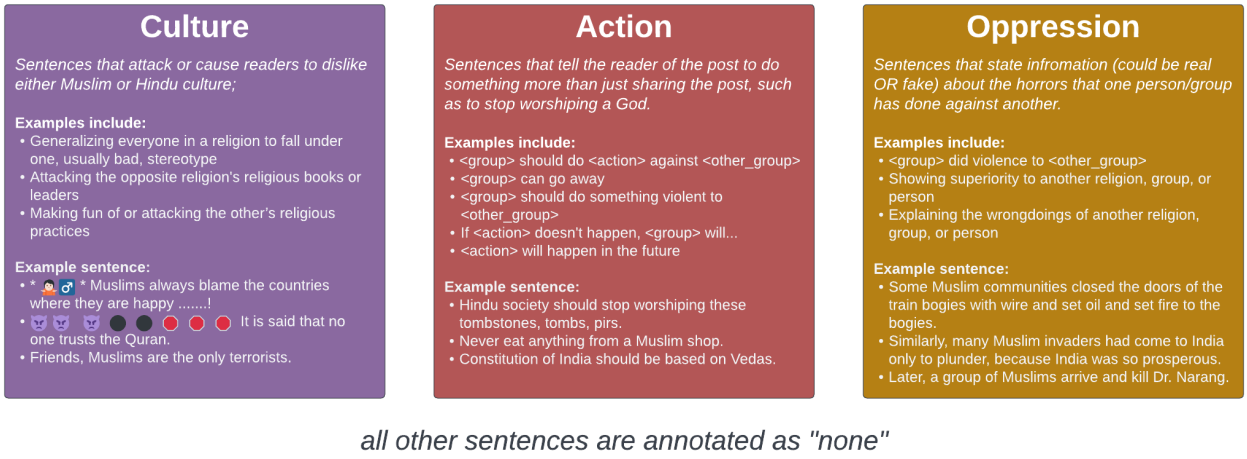

Figure 3: Differences by category of provocative sentences. *Graphic created by author.*

7,000 sentences were randomly sampled from the 25,000 sentence dataset for annotation, as it would be too time-consuming to annotate all 25,468 sentences. These 7,000 sentences were annotated by Ganning Xu (author) and Vaibhav Garg (mentor) into the categories of: "none", "culture", "action", or "oppression", with each sentence only labeled as one of these categories. Cohen Kappa scores were used as the annotation

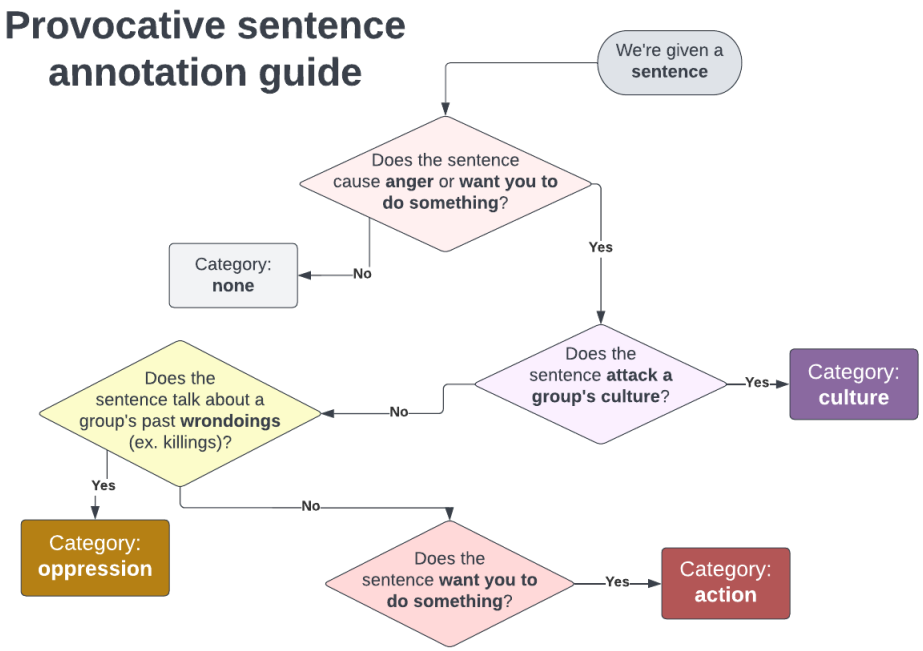

Figure 4: Categories for sentence annotation. *Graphic created by author.*

agreement metric to measure the agreement of annotators. Essentially, Cohen Kappa scores describe the reliability for two annotators who are labeling the same sentence (McHugh). Cohen Kappa scores are also adjusted for the number of sentences that were annotated the same purely by chance (McHugh). When annotating the same sentences, the more sentences the annotators classify as the same label, the higher the Cohen Kappa score. These scores fall in a range from -1 to 1 and can be interpreted as follows: ≤ 0 (no agreement), $0.01 - 0.2$ (none to slight agreement), $0.21 - 0.40$ (fair agreement), $0.41 - 0.60$ (moderate agreement), $0.61 - 8.0$ (substantial agreement), $0.81 - 1.00$ (almost perfect agreement) (McHugh). The annotation process for these 7,000 sentences were divided into three phases: cooperative annotation, sparse cooperative annotation, and separate annotation.

Phase 1: Cooperative Annotation (sentences 0-400): Phase 1 covers sentences 0 to 400. The dataset was split into four divisions: 0-100, 101-200, 201-300, and 301-400, with each division representing the sentence number. During this phase, both annotators annotate all sentences within each split. For example, after annotating sentences 0-100, both annotators discussed disagreements before annotating sentences 101-200. The average Cohen Kappa score for phase 1 was 0.48 (moderate agreement). There were four separate annotation rounds, each round having 100 sentences. The respective Cohen Kappa scores increased with each round of annotation and discussion: 0.24, 0.54, 0.61, and 0.56, with each score representing the agreement of each split. As expected, the agreement of annotations increased as more sentences were annotated.

Phase 2: Sparse Cooperative Annotation (sentences 401-1,000): Phase 2 covers sentences 401-1000. Sentences were split into 401-700 and 701-1000. With more sentences to annotate between

discussions, annotators completed the majority of annotations alone. Annotators met twice, once after annotating sentences 401-700 and once after sentences 701-100, to discuss disagreements. These meetings primarily settled the rest of the disagreements between annotations. The average Cohen Kappa score in phase 2 was 0.73 (substantial agreement). There were two separate rounds of annotation, with the respective Cohen Kappa score of each round being 0.7 and 0.75.

Phase 3: Separate Annotation (sentences 1,001-7,000): Phase 3 covers sentences 1001-7000. In phase 3, annotators did not collaborate, and each annotated different sentences. Phase 3 produced the majority of the 7,000 sentence annotated dataset. Annotators worked separately, with each annotator annotating 3,000 sentences.

Figure 5 shows the Cohen Kappa score progression between annotators in phases 1 and 2.

Thus, combining the sentences annotated in Phases 1, 2, and 3, a dataset of 7,000 annotated sentences was created, with categories of none (3,923 sentences), action (433 sentences), culture (1,065 sentences), and oppression (1,579 sentences).

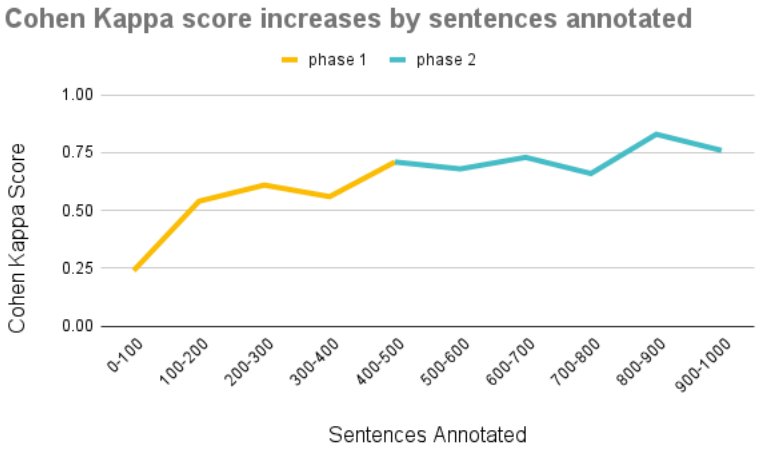

Figure 5: Cohen Kappa score increases by number of sentences annotated. *Graphic created by author.*

This annotation process produced a 7,000-sentence annotated dataset, with each sentence labeled as: none, culture, oppression, or action. A small sample of the dataset can be seen in Figure 6.

S_before	S	S_after	Label
* * When that Muslim When he came to attack 26/11, he knew very well 🐖 😈 😈 😈 🐖 😈 😈 😈 that he would not be saved himself.	* 4 * When those Muslims came to attack Parliament, they knew that they would be killed.	* * Even when those Muslims were going to fight the ship at the World Trade Center, they knew that their dead bodies also would not be recognized.	oppression
So these Muslim friends start visiting the homes of such secular friends.	His sisters or sisters-in-law begin to force him.	And how many such girls from Hindu families are spoiled or driven away.	none
The greedy wolves of vote in India even supported this massacre to please the Muslims of India.	Well, this was expected in a country like India.	Today, a Muslim organization called "Jaish-e-Mohammed" killed 42 CRPF soldiers in a suicide attack in India.	culture

Figure 6: Sample of three rows from the 7,000-sentence annotated dataset.

5. Data Analysis

After data collection and annotation, data analysis was performed to understand the differences between non-provocative sentences and the sub-types of provocative sentences. Specifically, length, sentiment, wordclouds, toxicity, empath (lexical analysis), topic modeling of these categories of sentences were examined.

5.1 Length Comparison

Provocative sentences are generally longer than non-provocative sentences. On average, non-provocative sentences are 96 characters long, while provocative sentences are about 135 characters long. Within the categories of provocative sentences (culture, action, oppression), the average lengths are 133, 132, and 138 characters, respectively. Intuitively, provocative sentences should be longer, as they are designed to anger or annoy the reader, usually resulting in more characters used. The distributions of lengths of non-provocative and provocative sentences are shown in Figure 7.

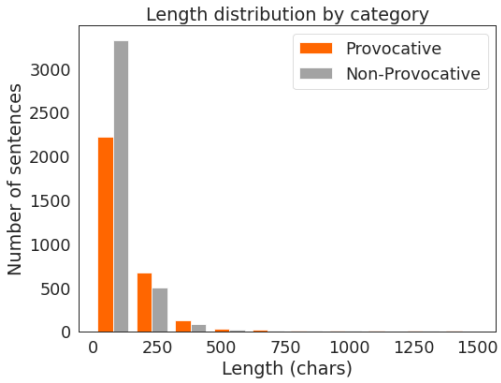

Figure 7: Provocative sentences are generally longer than non-provocative sentences. *Graphic created by author.*

The differences in length by each category of none, action, culture, and oppression is shown in Figure 8.

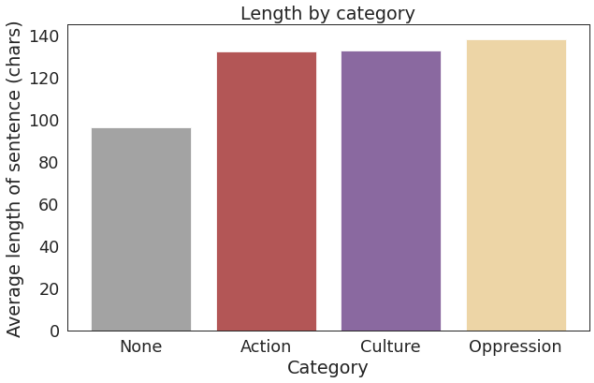

Figure 8: The lengths within categories of provocative sentences were strikingly similar. *Graphic created by author.*

5.2 Sentiment Comparison

Provocative sentences are generally more negative in sentiment than non-provocative sentences, as analyzed by NLTK's Vader Sentiment Intensity Analyzer. Sentiment scores range from -1 to 1, with scores closest to -1 as more negative ad scores closest to 1 as more positive. Non-provocative sentences and provocative sentences have an average sentiment of -0.17 and -0.02, respectively. The average sentiments for action, culture, and oppression are -0.04, -0.106, and -0.24, respectively. The drastic dip in sentiment for oppression sentences is likely due to their tendency to reference deaths and a group's past wrongdoings. However, it is important to note that the sentiment of action sentences was very similar to the sentiment of non-provocative sentences. The distribution of sentiment in provocative and non-provocative sentences can be seen in Figure 9.

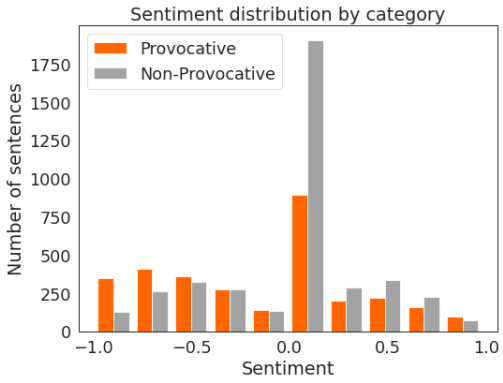

Figure 9: Provocative sentences occur more frequently near negative sentiments. Both provocative and non-provocative sentences show a spike at neutral sentiment. *Graphic created by author.*

A comparison of the sentiment of each category (none, action, culture, oppression) is shown in Figure 10.

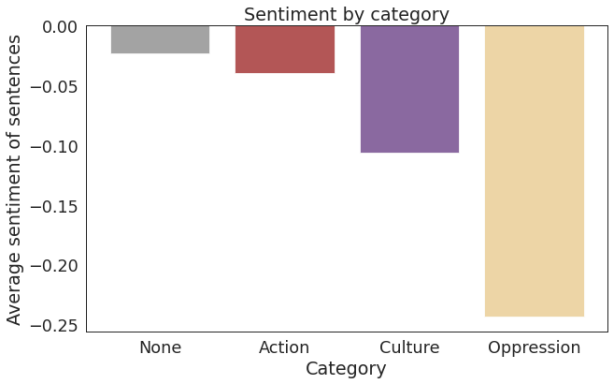

Figure 10: Culture and oppression sentences are more negative than action and non-provocative sentences. *Graphic created by author.*

5.3 Wordcloud Comparison

By recognizing the most common words in each category of provocative sentences, social media users can pick up on keywords that signal a type of provocation. Therefore, wordclouds for each category were created to determine the most common words for those types of sentences, as shown in Figure 11. It is important to note that for all wordclouds, the words "Hindu", "Muslim", "India", "Hindus", and "Muslims" were removed as they appeared in sentences of all categories. This common appearance is likely due to the fact that these sentences were extracted from Indian political WhatsApp groups. In action sentences, the most common word was "will", followed by "country" and "now". In action sentences, the author of the message wants the reader to do something, thus creating a sense of urgency with the word "now". In culture sentences, the most common word was "will", followed by "Quran", which follows the theme of attacking a group's culture, as the Quran is the religious text of Islam. In oppression sentences, the most common words are "will" and "girl". Through visual examination, many oppression sentences mention love jihad, a theory that involves the conversion of Hindu women to Islam by marriage (Ellis-Petersen and Khan), resulting in the prevalence of "girl". In "none" sentences, the words show no apparent pattern, as they are non-provocative.

However, "will" is one of the most common words across all sentences. This is likely because these sentences were extracted from fearful WhatsApp messages, which use fear or an imminent threat to scare readers.

Figure 11: Wordcloud of action, oppression, culture, and none. *Graphic created by author.*

5.4 Toxicity Comparison

Provocative sentences are generally more toxic than non-provocative sentences. Toxicity scores were determined by using Google's Perspective API. Toxicity scores range from $0 - 1$, with scores closest to 1 being more toxic. Non-provocative sentences have the lowest toxicity (0.13), followed by oppression (0.23), culture (0.26), and action (0.27). These toxicity scores are shown in Figure 12. Non-provocative sentences are generally neutral, as shown in sentiment analysis, making them less toxic. The three categories of provocative sentences had significantly higher toxicity scores, averaging 0.253, which is almost double the toxicity score of non-provocative sentences. Since Google's Perspective API has a rate limit of 60 requests

per minute, 60 sentences were randomly sampled from each category to determine the toxicity of the entire category.

When comparing these toxicity scores to sentiment scores, non-provocative sentences were the least negative and the least toxic. The range in toxicity of provocative sentences was only 0.04, which may easily have been caused by differences in sampling, as the sample size was only 60 sentences.

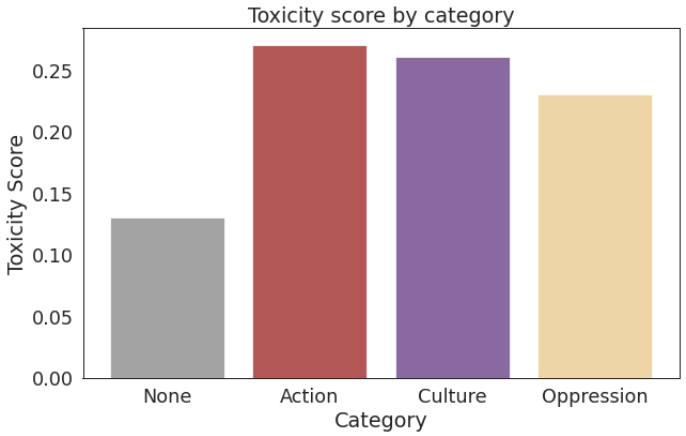

Figure 12: Average toxicity by category. *Graphic created by author.*

5.5 Empath Comparison

Additionally, lexical analysis was performed using Empath, a categorical analysis tool developed by Stanford University (Fast et al.). Empath measures the categories present in a piece of text. Oppression sentences consistently scored high on all of the negative categories tested, such as "negative_emotion", "crime", "suffering", "war", and "death". In contrast, non-provocative sentences consistently scored lower in all the previously mentioned negative categories tested but scored relatively higher in the positive, "fun" category. Through empath lexical analysis, it can be seen that provocative sentences contain more negative categories than non-provocative sentences. Empath results are shown in Figure 13.

5.6 LDA Topic Extraction

Latent Dirichlet Allocation (LDA) modeling was performed to extract topics from each category of sentences. Within each category, five topics were extracted. Table 1 shows the top two topics for each category. Within each provocative sentence category, negative sentiment is clearly shown, matching the high toxicity and negative sentiment of provocative sentences. Additionally, the theme of death and killing was prevalent across all types of provocative sentences. As expected, the topics extracted from non-provocative sentences show no clear sign of hatred or toxicity.

6. Model Development

Non-transformer and transformer-based models were tested to determine the best architecture for detecting and classifying provocative sentences. The non-transformer-based models tested were Random Forest

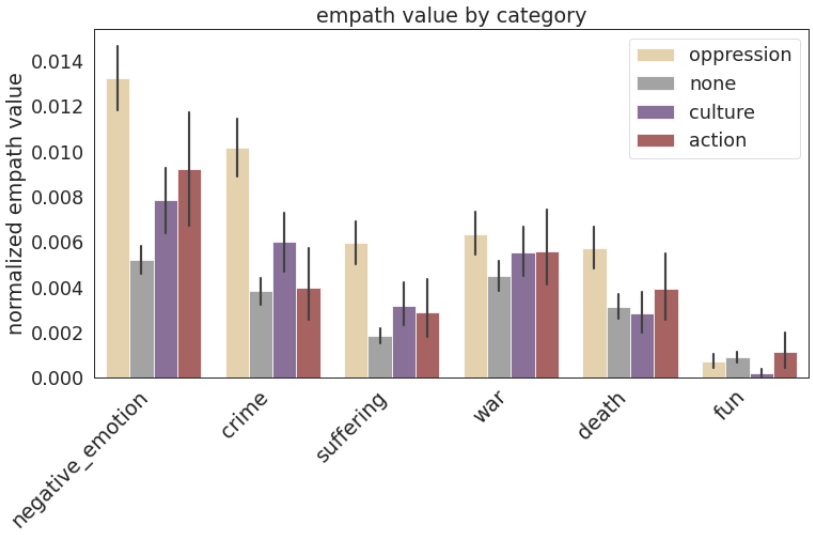

Figure 13: Empath distribution by category. *Graphic created by author.*

Category	Topic Extracted
Culture	countri, religion, terror, mosqu, children, work, hous, give, women, know
	hindu, kill, pakistan, india, communiti, tell, christian, today, start, go
Oppression	say, popul, burn, kashmir, take, bodi, go, like, place, dead
	peopl, year, terrorist, india, countri, percent, terror, attack, pakistan, children
Action	peopl, religion, wake, want, friend, countri, today, india, islam, brother
	countri, islam, stop, kill, children, like, today, want, peopl, leav
None	say, pakistan, today, allah, minist, like, call, go, write, build
	come, time, know, india, give, year, secular, peopl, govern, histori

Table 1: Table of topics modeled from LDA topic extraction. Note that all words were lemmatized to maintain consistency, resulting in some unusual spellings.

(RF), Multinomial Naive Bayes (MNB), Logistic Regression (LR), and LinearSVC (LSVC). When using non-transformer-based models, each sentence must first be transformed into its numerical representation. Word2vec, Term Frequency-Inverse Document Frequency (TF-IDF), and Universal Sentence Encoder were used to transform all sentences in the 7000-sentence dataset into numbers. The numerical representations from these word embedding systems were then fed into the non-transformer-based machine learning models for classification. Each non-transformer-based model was tested with word embeddings generated from Word2vec, TF-IDF, and Universal Sentence Encoder. Word2vec was developed in 2013 and utilizes a shallow, two-layer neural network to generate word embeddings (Mikolov et al.). TF-IDF is a measurement of the significance of each word to the meaning of a document (Aizawa). Similarly, the Universal Sentence Encoder (univ) was developed in 2018 and encodes sentences into high-dimensional vectors, which can then be used for text classification (Cer et al.).

In contrast, transformer-based models incorporate the process of transforming each sentence into its numerical representations into their architecture. Thus, word embeddings are not needed. Six transformer-based models were tested: BERT, RoBERTa, XLM-RoBERTa, DeBERTa, DistilBERT, and XLNet.

When training the model, S_{before} and S were concatenated together to form a longer input sentence for the model, to utilize context, and maintain consistency with annotations. The precision, recall, and F1-score of each of the four categories (none, action, oppression, culture) were evaluated (eight metrics total) to determine the performance of each model.

6.1 Oversampling

Since the sentence-level dataset is heavily imbalanced, oversampling (with replacement) was performed to equalize the number of sentences within each category before training each model. The original dataset contained the following distributions: none (56%), oppression (23%), culture (15%), and action (6%). Since non-provocative sentences make up the majority of the dataset, if a model was trained on this imbalanced dataset, the model would artificially favor the "none" class. Oversampling prevents the model from over-predicting a specific class due to its more frequent appearance in the dataset.

All provocative sentence categories were oversampled, so each class contained the same number of sentences as the non-provocative (none) class. Random rows in each provocative category were duplicated until each category contained 3923 sentences (number of sentences in the "none" category). An oversampled dataset of 15,692 sentences was produced utilizing this method. Figure 14 shows the difference between the original and oversampled dataset.

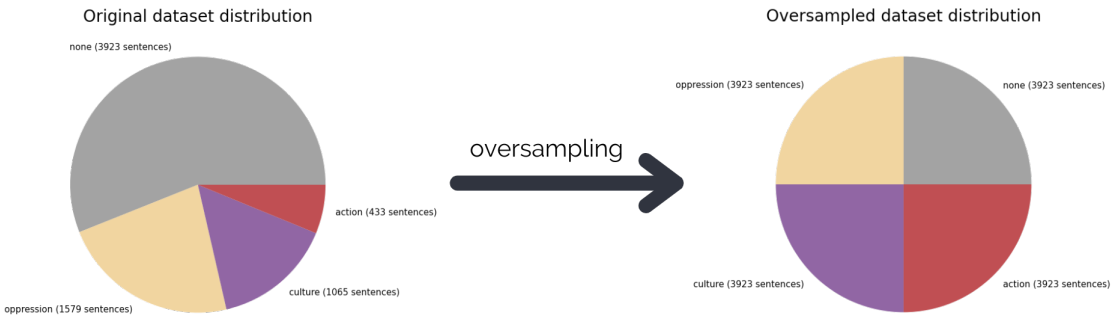

Figure 14: Oversampling balances each category in the dataset. Oversampling the original dataset produced a 15,692-sentence dataset, with each category representing 3923 sentences.

6.2 Cross Validation

When training each model, stratified 10-fold cross-validation was performed. Essentially, this splits the oversampled dataset into 10 sections. Each section contains an equal or very similar number of sentences from each category. In 10-fold cross-validation, a model is trained and evaluated in 10 iterations. Within each iteration, a different section of the dataset acts as training data and testing data. 90% of the dataset (14,123 sentences) was used to train, and 10% of the dataset (1,569 sentences) was used to evaluate the model each iteration. For each iteration, all transformer-based models were trained on seven epochs, while non-transformer-based models were trained with default settings. When computing the overall score for each evaluation metric, each metric's score across all ten iterations was averaged to determine the actual value.

6.3 Model Results

Table 2 shows the metrics for each of the eight categories. To save space, labels have been substituted with the following: Random Forest = RF, Multinomial Naive Bayes = MNB, Logistical Regression = LR, and LinearSVC = LSVC. These results are also shown graphically in Figure 15.

Model	None			Oppression			Action			Culture		
	Prec	*Rec*	*F1*	*Prec*	*Rec*	*F1*	*Prec*	*Rec*	*F1*	*Prec*	*Rec*	*F1*
BERT	1.00	0.98	0.99	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99	0.99
RoBERTa	1.00	0.98	0.99	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99	0.99
XLM-RoBERTa	1.00	0.98	0.99	0.99	0.99	0.99	0.98	1.00	0.99	0.99	0.99	0.99
DeBERTa	1.00	0.98	0.99	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99	0.99
DistilBERT	1.00	0.98	0.99	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99	0.99
XLNet	1.00	0.98	0.99	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99	0.99
(tfidf) RF	0.51	0.56	0.53	0.65	0.53	0.58	0.65	0.57	0.61	0.53	0.63	0.58
(tfidf) MNB	0.70	0.67	0.69	0.79	0.78	0.79	0.90	0.97	0.93	0.82	0.80	0.81
(tfidf) LR	0.83	0.67	0.75	0.83	0.86	0.85	0.92	0.99	0.95	0.84	0.89	0.86
(tfidf) LSVC	0.90	0.69	0.78	0.85	0.92	0.88	0.95	1.00	0.97	0.87	0.96	0.91
(word2vec) LR	0.38	0.27	0.31	0.34	0.43	0.38	0.36	0.60	0.45	0.40	0.15	0.22
(word2vec) MNB	0.28	0.43	0.34	0.34	0.22	0.26	0.31	0.51	0.38	0.35	0.06	0.11
(word2vec) LR	0.28	0.29	0.29	0.33	0.34	0.33	0.31	0.50	0.38	0.29	0.09	0.14
(word2vec) LSVC	0.29	0.25	0.27	0.35	0.39	0.37	0.32	0.51	0.39	0.30	0.13	0.18
(univ) RF	0.53	0.40	0.45	0.60	0.56	0.58	0.57	0.62	0.59	0.49	0.61	0.55
(univ) MNB	0.51	0.50	0.50	0.57	0.51	0.54	0.56	0.45	0.50	0.46	0.61	0.53
(univ) LR	0.57	0.51	0.54	0.65	0.64	0.65	0.66	0.73	0.69	0.60	0.61	0.61
(univ) LSVC	0.83	0.69	0.75	0.84	0.86	0.85	0.92	0.99	0.96	0.84	0.91	0.87

Table 2: Table of both transformer and non-transformer based models. All transformer-based models performed very well, while the performance of non-transformer-based models varied based on the word embedding technique and the model used.

It is clear that transformer-based models far outperform non-transformer-based models in this multi-class classification task. The transformer-based models achieved an average precision, recall, and F1-score of 0.99. While all transformer-based models achieve almost identical metrics, DeBERTA and DistilBERT achieve the highest recall, precision, and F1 scores. For provocative sentence detection, technically speaking, DeBERTa and DistilBERT likely are the most effective. Both models scored equally on the macro precision and recall scores, each with 0.992. This means that: (1) when a sentence is predicted to be within a specific category, it is correct, on average 99.2% of the time, (2) on average, 99.2% of the total sentences from each category were correctly identified from the testing dataset. However, the difference in metrics across transformer-based models is so tiny that any transformer-based model can probably be used for provocative sentence detection. It is important to note that the high-scoring evaluation metrics from transformer-based models are likely not a result of overfitting, as cross-validation testing prevents overfitting (Pius). Additionally, these metrics were evaluated on testing data, which are sentences that the model was not trained on, making overfitting highly unlikely.

Additionally, a loss per epoch by model type graph for all transformer-based models was created, as shown in Figure 16. It can be seen that at around three epochs, all transformer-based models reach a

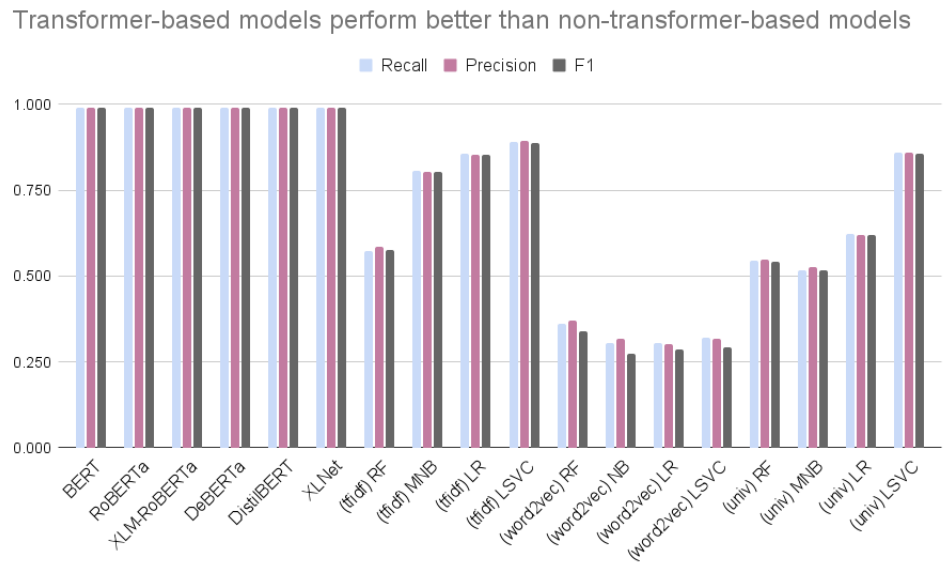

Figure 15: Transformer and non-transformer-based models evaluated on recall, precision, and F1-score. *Graphic created by author.*

minimum loss value.

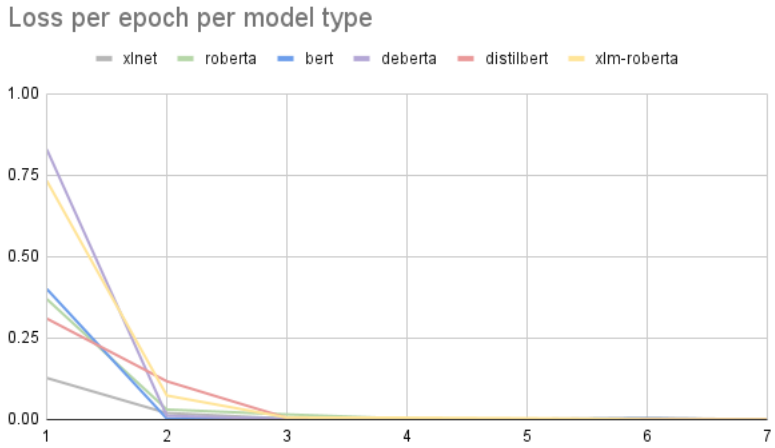

Figure 16: Deberta starts off with the highest loss, but all six transformer-based models end up with a loss very close to zero. *Graphic created by author.*

Non-transformer-based models performed worse than transformer-based models, with average precision, recall, and F1-Score of 0.58, 0.58, and 0.57, respectively. The non-transformer-based models performed the best when utilizing TF-IDF embeddings, with average precision, recall, and F1-score all being 0.78. LinearSVC combined with TF-IDF embeddings produced the highest precision, recall, and F1 scores of 0.892, 0.894, and 0.888, respectively. Embeddings created by the Universal Sentence Encoder paired with LinearSVC performed similarly to TF-IDF embeddings, producing recall, precision, and F1-scores of 0.861,

0.859, and 0.858, respectively. When examining Word2vec, it only achieves average precision, recall, and F1-scores of 0.32, 0.33, and 0.30, respectively. The best model when using Word2vec embeddings was Random Forest, which achieved recall, precision, and F1-scores of 0.361, 0.370, and 0.340, respectively. To summarize, transformer-based models classify provocative sentences the best, followed by TF-IDF embeddings, Universal Sentence Encoder embeddings, and lastly, Word2vec embeddings utilizing non-transformer-based techniques. To simplify the results in Table 2, the precision, recall, and F1-Score across all four categories were averaged to produce Table 3.

Model Type	**Recall**	**Precision**	**F1-Score**
(univ) RF	0.546	0.548	0.543
(univ) MNB	0.517	0.526	0.517
(univ) LR	0.623	0.621	0.621
(univ) LSVC	0.861	0.859	0.858
(tfidf) RF	0.574	0.585	0.575
(tfidf) MNB	0.806	0.803	0.804
(tfidf) LR	0.856	0.854	0.852
(tfidf) LSVC	0.892	0.894	0.888
(word2vec) RF	0.361	0.370	0.340
(word2vec) NB	0.304	0.319	0.273
(word2vec) LR	0.305	0.301	0.285
(tfidf) LSVC	0.892	0.894	0.888
BERT	0.991	0.992	0.991
RoBERTa	0.989	0.990	0.990
XLM-RoBERTa	0.989	0.989	0.989
DeBERTa	0.992	0.992	0.992
DistilBERT	0.992	0.992	0.992
XLNet	0.990	0.990	0.990

Table 3: Macro averages between recall, precision, and F1-score across all categories. The "best" model for transformer-based, Word2vec embeddings, TF-IDF embeddings, and Universal Sentence Encoder embeddings are colored.

The worse performance of non-transformer-based models compared to transformer-based ones are likely due to transformer models' implementation of self-attention, which differently weighs each part of the input data to determine characteristics that hint at sentence type.

6.4 Example Predictions

Several sentences not in the 7,000-sentence dataset, but in the 25k-sentence dataset were tested with the BERT transformer-based-model. These predictions are shown in Example 3.

7. Discussion and Conclusion

Our research is the first to present a 7,000-sentence annotated provocative sentence dataset for other researchers to perform provocative sentence detection. Data analysis was performed on this dataset to determine differences in length, sentiment, toxicity, empath lexical analysis, topics extracted, and most common words in each category of provocative sentence. Lastly, transformer-based and non-transformer-based ma-

Example 3: Testing detection of provocative sentences

Each of these sentences was fed into our BERT model:

- **Sentence**: "Every Hindu is a black hole." (**Predicted**: culture)
- **Sentence**: "To eliminate non-Muslims What to do with their property, their women, their children? Quran about him, Muslims should consider him as the gift of Allah and enjoy it." (**Predicted**: action)
- **Sentence**: "Or when the girl's brother comes to save her father, then they meet the Muslim father and son, the father and brother of the victim girl, both with knives, are mercilessly killed and escape." (**Predicted**: oppression)
- **Sentence**: "That evening, Rahul's father decided That it is no longer possible to live in Kashmir. The next day, he went to Jammu by a taxi." (**Predicted**: none)

All predictions were consistent with human agreement.

chine learning models were tested to determine the best-suited model for provocative sentence detection.

Through this research, provocative sentences have been observed to be generally longer, more negative, and toxic, and often contain topics relating to crime, suffering, and war. Additionally, provocative sentences are generally more toxic than non-provocative sentences. This may be because provocative sentences contain content that anger the reader or motivates them to carry out some action. We also observe that transformer-based models are far superior to non-transformer-based models, with DeBERTa and DistilBERT achieving almost perfect precision, recall, and F1 scores of 0.992. Among non-transformer-based models, LinearSVC with TF-IDF embeddings performed the best, achieving precision, recall, and F1-scores of 0.894, 0.892, and 0.888, respectively.

With almost 6 billion people being projected to use social media by 2027 (Dixon), the importance of provocative sentence detection is critical to social progress. Especially in times leading up to political elections, political leaders may utilize provocative to sway their target audience in favor of a certain candidate, causing riots and violence. While our research utilized data from political WhatsApp groups, our results apply to any social media platform.

7.1 Further Research

1. **Optimized model for smartphones**: To deploy a provocative sentence detection model onto social media platforms, further research should be completed on packaging the final model to a size that can reasonably fit on deployment and predict provocative sentences fast enough. This brings into consideration whether to deploy this packaged model on the client-side device or the server side.
2. **Continuous feedback system**: For a provocative sentence detection model to be deployed on social

media platforms, there needs to be a continuous feedback loop where users can indicate the correctness of provocative sentences detected by the model and a method to mark sentences as provocative. This allows our model to adapt continuously to emerging social media trends and recent topics.

3. **Expanding our dataset to other social media platforms**: Data from our model was acquired from political WhatsApp groups. While this data still applies to other platforms like Facebook or Twitter, it is beneficial to create a specialized annotated dataset to classify provocative sentences occurring outside of political WhatsApp groups accurately.

Our research proposes that provocative sentences are critically dangerous yet left undetected by current methods on social media platforms. Accordingly, this paper presents three main contributions:

1. A 7,000-sentence annotated dataset with each sentence annotated as one of four categories: non-provocative, culturally provocative, oppressive provocative, and action provocative
2. Data analysis to determine the textual differences in provocative and non-provocative sentences.
3. A thorough analysis of the effectiveness of non-transformer and transformer-based models on provocative sentence classification, and a state-of-the-art performing machine learning model to accurately detect and classify provocative sentences.

The 7,000-sentence dataset and all code used are publicly available on GitHub.

References

Aizawa, Akiko. "An information-theoretic perspective of tf–idf measures." *Information Processing Management*, vol. 39, no. 1, 2003, pp. 45–65. https://doi.org/https://doi.org/10.1016/S0306-4573(02)00021-3.

Aluru, Sai Saket, et al. "Deep Learning Models for Multilingual Hate Speech Detection." *arxiv*, 2020.

Bhargava, Yuthika. Be ready to trace the origin of messages, WhatsApp told. *The Hindu*, Dec. 2021. www.thehindu.com/news/national/whatsapp-not-seeking-decryption-but-location-identity-of-those-sending-provocative-messages-says-centre/article61518804.ece.

Brashier, Nadia M, and Daniel L Schacter. "Aging in an Era of Fake News." *Curr Dir Psychol Sci*, vol. 29, no. 3, May 2020, pp. 316–23.

Buyse, Antoine. "Words of Violence: "Fear Speech," or How Violent Conflict Escalation Relates to the Freedom of Expression." *Human Rights Quarterly*, vol. 36, no. 4, 2014, pp. 779–97. *JSTOR*, www.jstor.org/stable/24518298. Accessed 31 Oct. 2022.

Cer, Daniel, et al. "Universal Sentence Encoder." *CoRR*, vol. abs/1803.11175, 2018. *arXiv*, arxiv.org/abs/1803.11175.

Dixon, S. Number of worldwide social network users 2027. *Statista*, Sept. 2022. www.statista.com/statistics/278414/number-of-worldwide-social-network-users.

Ellis-Petersen, Hannah, and Ahmer Khan. 'they cut him into pieces': India Love Jihad Conspiracy theory turns lethal. *The Guardian*, Jan. 2022. www.theguardian.com/world/2022/jan/21/they-cut-him-into-pieces-indias-love-jihad-conspiracy-theory-turns-lethal.

Fast, Ethan, et al. "Empath: Understanding Topic Signals in Large-Scale Text." *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016. https://doi.org/10.1145/2858036.2858535.

Gill, Monique. The 1984 sikh genocide - 36 years on. *Human Rights Pulse*, June 2020. www.humanrightspulse.com/mastercontentblog/the-1984-sikh-genocide-36-years-on.

Halder, Debajyoti, et al. "fybrrChat: A Distributed Chat Application for Secure P2P Messaging." 2022. https://doi.org/10.48550/ARXIV.2207.02487.

Malik, Jitendra Singh, et al. "Deep Learning for Hate Speech Detection: A Comparative Study." 2022. https://doi.org/10.48550/ARXIV.2202.09517.

Mathew, Binny, et al. "Hate Begets Hate: A Temporal Study of Hate Speech." *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, 2020, pp. 1–24.

McHugh, Mary L. "Interrater reliability: the kappa statistic." *Biochem. Med. (Zagreb)*, vol. 22, no. 3, 2012, pp. 276–82.

Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*, 2013.

Pius, Subish. How to use cross validation to reduce overfitting. *Artificial Intelligence +*, Sept. 2022. www.aiplusinfo.com/blog/how-to-use-cross-validation-to-reduce-overfitting/.

Repple, Jonathan, et al. "From provocation to aggression: the neural network." *BMC Neuroscience*, vol. 18, no. 1, Oct. 2017, p. 73. https://doi.org/10.1186/s12868-017-0390-z.

Singh, Simran Jeet. It's time India accept responsiblity for its 1984 sikh genocide. *Time*, Oct. 2014. time.com/3545867/india-1984-sikh-genocide-anniversary/.

Verma, Monica. Opinion: Provocation theory will be death knell for not just Hindus but also Indian democracy. *News18*, Apr. 2022. www.news18.com/news/opinion/opinion-provocation-theory-will-be-death-knell-for-not-just-hindus-but-also-indian-democracy-5016739.html.

Zhang, Ziqi, et al. "Detecting Hate Speech on Twitter Using a Convolution-Gru Based Deep Neural Network." *European Semantic Web Conference*. Springer, 2018, pp. 745–60.