centific

How Businesses Can Use Technology to Fight Abusive Language

By Dr. Sreenivasulu Madichetty, Senior AI Engineer

It is impossible to overstate the role that social media plays in the world, from shaping how people behave to influencing the way businesses operate. There are 4.9 billion people using social media, or more than 60 percent of the world's population. And 73 percent of marketers believe that their efforts through social media marketing have been "somewhat effective" or "very effective" for their business.

At the same time, the popularity of social media has a downside, including the rise of abusive language – or expressing hatred of a particular group of people. Abusive language has wreaked havoc on social media. Abusive language harms people and degrades society. From a business standpoint, abusive language also creates serious brand safety issues. Let's take a closer look at the problem and how to fight it.

Severity of the Problem

Unfortunately, abusive language is a widespread problem, as the following data indicates:

48%
people globally report experiencing threats including sustained bullying (5 percent), stalking (7 percent), and account takeover by someone they know (6 percent).

Over the past 3 years, the odds of users experiencing abuse have increased by 1.3 times.

41%
Americans reported personally experiencing varying degrees of harassment and bullying online, in a survey by Pew in 2017. Globally, 40 percent of people reported similar experiences.

And the problem will only get worse as more social media apps take hold. TikTok, which did not even exist until recently, is now a worldwide phenomenon. Unfortunately, more people using TikTok also means a rise in abusive language.

What Does It Cost for Organizations to Address Abusive Language?

Abusive language has a direct impact on organizations, costing them resources and money to combat its spread.

Facebook, Twitter, LinkedIn and many other technology companies spend millions of euros to address abusive language.

Moreover, legislative bodies are taking action. For instance, the European Union's Digital Services Act requires Big Tech companies to police their platforms more strictly to better protect European users from abusive language, disinformation, and other harmful online content. Failure to comply could result in fines of up to 6 percent of a company's annual global revenue and a ban from the European Union.

To cite another example: French parliament has enacted a law mandating social media and tech firms such as Twitter, Facebook, and Google remove abusive language within 24 hours of being flagged. Failure to comply could end in these companies facing fines of up to $1.36 million.

Hence, early detection of abusive language can have a significant impact on a company's, brand reputation – and, more importantly, protect people from harm.

Facebook said it has tripled the size of its teams working in safety and security since **2016 to over 35,000 people**

How to Detect Abusive Language

People cannot stop abusive language without technology. And technology cannot stop abusive language without people being involved. No person can review every social media post and tag it as potentially abusive language. But artificial Intelligence (AI) can help by identifying potentially abusive language and take appropriate action.

The recommended methodology is to collect data around abusive language and train a machine learning model to be used for early detection of abusive language. Following are recommended steps:

1. Data collection: data should be collected relevant to the abusive language.

2. Data annotation: data need to be annotated by people.

3. Data preparation: data need to be cleaned and preprocessed to fed into the models.

4. Feature engineering: data needs to be transformed into features that can be used in the model training.

5. Model training: a model is selected (either machine learning or deep learning) for training the data.

6. Model testing: the model is tested in real time with various cases.

7. Model improvement: the model is improved based on its performance.

This graphic shows each step aligned with a success criteria:

1. Data Collection

Abusive language data can be posted in different formats such as text, image, audio, and video on various platforms.

The authors of "Directions in abusive language training data, a systematic review: Garbage in, garbage out" demonstrated that data about abusive language can be posted on a variety of platforms including Twitter, stormfront, Facebook, Reddit, GAB, and Wikipedia. They collected 63 datasets for the different platforms. There are 27 datasets relevant to Twitter as a result. This is because Twitter is considered to be an especially toxic platform.

In this white paper, we will only be discussing text data. Furthermore, most Twitter datasets are available in the form of tweetids. The tweets must be extracted from the tweetids using the Twitter API (Application Programming Interface).
Text data for abusive language is available in a variety of languages including English, German, Hindi, Greek, Indonesian, Spanish, Portuguese, Slovene, Croatin, French and others. According to "Directions in abusive language training data, a systematic review: Garbage in, garbage out," English language data sets are the most used among other languages. As a result, we focus on data sets in English.

2. Data Annotation

Data annotation refers to activity of assigning a label to the collected data. For example, if we have a tweet which says "Navaratri is a festival of demons," this tweet should be annotated as "abusive."

If open-sourced datasets are used, they are already annotated. But sometimes for specific business use cases, data needs to be annotated manually. This process is time consuming.

At Centific, we have expertise in annotating video and text data. We use software that speeds up the annotation process.

3. Data Preparation

For any natural language processing (NLP) task, we need to perform text cleaning/processing tasks. This can help us in building robust models. Some of the common data preparation steps that we do as part of our best practices are:

- **Conversion.** We convert the text from capitalization to lower case to standardize the data because the same words appeared differently, reducing accuracy.
- **Tokenization.** Tokenization is the process of breaking down a sentence into words that can be assigned meaning more easily.
- **Stop word removal.** We remove stop words from sentences that do not add to the content's meaning. Stop words include a, an, the, of, is, from, and so on.
- **Lemmatization.** The lemmatization technique is used to remove inflections and convert to the root form. For instance, lemmatization could be used to change the word "better" to "good."
- **Removal of punctuation marks, numerals, and unknown symbols.** We remove punctuation marks, numerals, and unknown symbols from the text because they provide little context.

Most of the pre-processing steps can be applied to any classification model.

4. Feature Engineering

Computers are bad at understanding plain text. All they understand and interpret well is numbers. Hence, there is a need for us to convert our text data into numerical representation so that computers can interpret them. We use the following feature engineering methods:

- Bags-Of-Words models (BOW)
- Term Frequency and Inverse Document Frequency (TF-IDF) features for uni-gram, bi-gram and tri-gram
- Counting hashtags, mentions, retweets, and URLs, as well as features for the number of characters, words, and syllables
- Parts-Of-Speech (POS) tags
- Sentence embedding using the Universal Sentence Encoder (it can be used in deep learning methodologies also)
- TF-IDF vector
- BoW vector
- Lexicon of abusive words

The above-mentioned steps come under the feature engineering phase. In the feature engineering phase, TF-IDF vectors and Lexicon of abusive words can be used for abusive language detection.

5. Model Training

Once the data is collected and preprocessed, and feature engineering is completed, it needs to be used to train a machine learning model. There are three different approaches to train a machine learning model as shown below:

- Feature-based approach
- Neural network-based approaches
- Transfer learning approaches

Feature-Based Approach

In this approach, any of the above-mentioned feature extraction methods, or any combination of them, can be used to create a feature vector. The feature vector can be fed into any machine learning algorithm to learn the patterns.

Some of all the algorithms that can be used in this approach are given below:

- Logistic regression
- Support Vector Machine (SVM) classifier
- Naive bayes
- Decision tree
- XGBoost
- Gradient Boosting
- Random forest and others

To identify the best machine learning algorithm for the problem, all algorithms must be implemented in general. All algorithms, however, can be implemented in a matter of minutes with the help of the Lazy Predict library. For more details about setting up the Lazy predict library, check out this link.

For setting up lazy predict library and more details please check the link below: https://lazypredict.readthedocs.io/en/latest/installation.html

Here is an example of how one can use the Lazy Predict library for classification:

```
from lazypredict.Supervised import LazyClassifier
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
data = load_breast_cancer()
X = data.data
y= data.target
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=.5,random_state =123)
clf = LazyClassifier(verbose=0,ignore_warnings=True, custom_metric=None)
models,predictions = clf.fit(X_train, X_test, y_train, y_test)
models
```

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score
LinearSVC	0.989474	0.987544	0.987544	0.989462
SGDClassifier	0.989474	0.987544	0.987544	0.989462
MLPClassifier	0.985965	0.986904	0.986904	0.985994
Perceptron	0.985965	0.984797	0.984797	0.985965
LogisticRegression	0.985965	0.98269	0.98269	0.985934
LogisticRegressionCV	0.985965	0.98269	0.98269	0.985934
SVC	0.982456	0.979942	0.979942	0.982437
CalibratedClassifierCV	0.982456	0.975728	0.975728	0.982357
PassiveAggressiveClassifier	0.975439	0.974448	0.974448	0.975464
LabelPropagation	0.975439	0.974448	0.974448	0.975464
LabelSpreading	0.975439	0.974448	0.974448	0.975464
RandomForestClassifier	0.97193	0.969594	0.969594	0.97193
GradientBoostingClassifier	0.97193	0.967486	0.967486	0.971869
QuadraticDiscriminantAnalysis	0.964912	0.966206	0.966206	0.965052
HistGradientBoostingClassifier	0.968421	0.964739	0.964739	0.968387
RidgeClassifierCV	0.97193	0.963272	0.963272	0.971736
RidgeClassifier	0.968421	0.960525	0.960525	0.968242
AdaBoostClassifier	0.961404	0.959245	0.959245	0.961444
ExtraTreesClassifier	0.961404	0.957138	0.957138	0.961362
KNeighborsClassifier	0.961404	0.95503	0.95503	0.961276
BaggingClassifier	0.947368	0.954577	0.954577	0.947882
BernoulliNB	0.950877	0.951003	0.951003	0.951072
LinearDiscriminantAnalysis	0.961404	0.950816	0.950816	0.961089
GaussianNB	0.954386	0.949536	0.949536	0.954337
NuSVC	0.954386	0.943215	0.943215	0.954014

To find the source code please check the link below:
https://lazypredict.readthedocs.io/en/latest/usage.html#classification

Our Recommendation for the Feature-Based Approach

The recommended methods of the feature-based approach are BoW and TF-IDF vector with the combination of SVM and Random Forest classifiers. We have used the data of 42,416 tweets (26,081 abusive language tweets, 16335 non-abusive language tweets) for experimenting with the he methods. Among them, 38,175 and 4,241 tweets were used for training and testing. We achieved the results with precision of 94 percent in feature-based approaches.

Neural Network-Based Approaches

Neural network-based approaches consist of three or more layers and attempts to simulate the behavior of the human brain. It is capable of automatically learning the relationships among the vast amounts of data.

The same pre-processing steps mentioned in a feature-based approaches can be used in neural network-based approaches. There is no feature engineering phase in deep learning approaches; instead, features are extracted automatically from tweets using the following embeddings:

Embeddings:

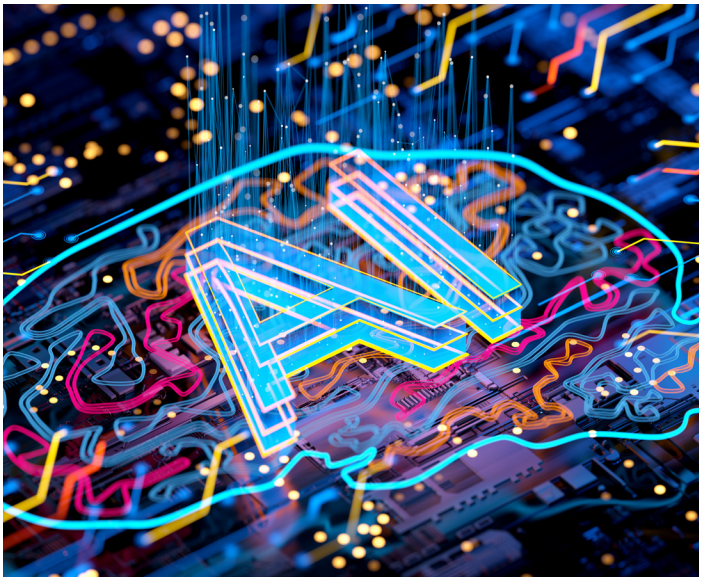
- Word2vec
- Glove
- FastText
- Character Embeddings

Methods:

- Convolutional Neural Networks (CNN)
- Long Short Term Memory (LSTM)
- LSTM with attention mechanism
- Bi-directional Long Short Term Memory (Bi-LSTM)
- Bi-LSTM with attention mechanism
- Gated Recurrent Unit (GRU)

Drawbacks of Feature-Based and Neural Network-Based Approaches

- Feature-based and neural network-based approaches have the major disadvantage of not capturing contextual information about abusive language.
- Feature-based and neural network-based approaches do not capture semantic information if the training and testing vocabulary are different. These are vocabulary dependent.
- Deep learning methodologies are hampered by a lack of labelled data or an inability to improve generalized properties.



Our Recommendation for Neural Network-Based Approaches

With the neural network-based approaches, we recommend using Bi-LSTM with an attention mechanism - with the combination of Glove embeddings and character embeddings for the detection of abusive language.

Transfer Learning Approaches (Pre-Trained Models)

A transfer learning approach is one in which a model is trained with a large amount of data and then applied to another problem with less data. The main idea behind transfer learning is to apply what a model has learned from one task as a starting point to another task. It can be used when the model has been trained with a lot of labelled data and applied to another task where the labelled data is little.

Like the deep learning approaches, we can apply the pre-processing steps or not required. We can use the following transfer-learning approaches directly:

- BERT
- HURTBERT
- HateBERT

BERT: How does the BERT model detect hate speech?

The authors of "A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media" proposed a transfer learning approach for abusive language understanding. It combines the unsupervised pre-trained BERT model with some new supervised fine-tuning strategies to address the drawbacks of feature-based and neural network-based approaches. In order to detect abusive language, the BERT model collects contextual information. In order to improve abusive language detection, the authors used various fine-tuning strategies on the BERT model, such as:

- Add nonlinear layers
- Insert a Bi-LSTM layer
- Insert a CNN layer
- BERT based fine-tuning
- Insert an LSTM layer

Recent studies based on BERT suggest that additional hate-specific knowledge from outside the fine-tuning dataset may help with generalization. Such knowledge can be obtained through further masked language modelling pre-training on an abusive corpus (as cited in "HateBERT: Retraining BERT for Abusive Language Detection in English") or features from a lexicon of abusive language (as cited in "HurtBERT: Incorporating Lexical Features with BERT for the Detection of Abusive Language").

HURTBERT: How are the lexical features used in the BERT model (HURTBERT) for detection of abusive language?

The authors of "HurtBERT: Incorporating Lexical Features with BERT for the Detection of Abusive Language" introduced HURTBERT model for identifying abusive language on social media. Domain-specific knowledge is incorporated into the BERT model in two ways in this work, using HURTLex such as Encoding and Embeddings.

HURTLex is a lexicon of abusive words divided into 17 categories. The tweet vector has a dimension of 17 and is built from tweets based on the presence of words in different categories. Then it is referred to as HURTLex encoding.

To obtain vector representation, word embeddings are extracted from each word category and concatenated vectors are passed to the LSTM layer. To obtain the tweet prediction, the final representation is concatenated with the BERT layer output and then passed to the dense layer.

The process of obtaining the final representation is referred to as HURTBERT embeddings.

HateBERT: How does the BERT model detect hate speech using Reddit Abusive Language (HateBERT)?

The authors of "HateBERT: Retraining BERT for Abusive Language Detection in English" developed the HateBERT model for detecting abusive language in tweets. The model employs the BERT model and is re-trained with the Reddit Abusive Language English (RAL) dataset, which contains all posts from banned profiles or communities and posts containing hateful, offensive, and abusive language information. The dataset is referred to as the RAL-E dataset.

The authors used 1,478,348 messages (for a total of 43,379,350 tokens) to retrain the English BERT base-uncased model using the Masked Language Model (MLM) objective. As a test set, the remaining 14,932 messages (441,271 tokens) were used.

They retrained in batches of 64 samples for 100 epochs (nearly 2 million steps), including up to 512 sentencepiece tokens. They used Adam with a learning rate of 5e-5 and trained on a single Nvidia V100 GPU using the hugging face code. Because it was trained on domain-specific content rather than general content, the HateBERT model outperforms the general BERT model in several abusive language detection tasks.

We have summarized and compared the machine learning and deep learning models which includes time to implement, volume of data, approximation cost of implementation, deployment complexities, and recommendation approaches:

Models	Time to Implement (in weeks)	Volume of Data	Approx cost of imple-mentation	Deployment Complexities	Recommendation
Feature-based Approaches	12 - 24	Small (10k – 20k)	$4,99,200	Simple	These are recommended when the model deploy-ment
Deep learning based approaches	24- 32	Large (>20k)	$ 6,65,600	Complex (large number of parameters, GPU needed)	To be used if large amount of data and computing resources are available
Transfer-learn-ing approaches	24-32	Small or Large (based on context)	$ 6,65,600	Less complex than DL mod-els	Used when the accuracy is important

6. Model Testing

In general, at Centific, we use 10-percent-to-20-percent of the available data to test any trained model. We put the model through various tests. For example, we train the model on one domain and then test it on another. We will evaluate the model's performance using various metrics such as accuracy, precision, recall, and the F1-score. We also examine whether the model is underfitting or overfitting. If either of these conditions is met, we proceed to the model improvement stage.

Model overfitting occurs when a model performs better on training data but performs worse on testing data. Model underfitting occurs when a model fails to improve performance on both training and testing data.

7. Model Improvement

If the model is either underfitting or overfitting, we proceed with the model improvement steps.

When the model is overfitting, we:

- Reduce the number of features
- Increase the amount of training data by data augmentation
- Early stopping
- Regularization including lasso and ridge regression
- Reduce the complexity of the model.
- Add dropout layer

When the model is underfitting, we:

- Increase the number of features
- Increase the model complexity
- Remove the noise from the data
- Increase the number of epochs

Findings

If anyone needs to deploy a simple model for detection of abusive language on social media, feature-based approaches can be used, such as random forest or an SVM classifier based on TF-IDF features

compared to other classifiers. This delivers better accuracy compared to the other classifiers.

If someone does not care about complexity and is only concerned with performance, they can use transfer learning approaches -- specifically the HateBERT model, which achieved the F1-score of 92 percent for detecting tweets with abusive language. We have used the data of 42,416 tweets (26,081 abusive language tweets, 16,335 non- abusive language tweets) for experimenting the methods. Among them, 38,175 and 4,241 tweets were used for training and testing.

All traditional approaches can be applied to any classification task, such as event detection, disaster tweet detection, and so on. With minor modifications, transfer learning approaches can also be employed for classification tasks. For example, in the HurtBERT model, the authors used the HurtLex Lexicon; however, depending on the classification task, we can use a domain-specific lexicon.

Technology Needs People and Processes

As noted above, fighting abusive language requires people, technology, and a process working

together. To keep this document concise, we have focused on a process (or methodology) for training a technology (specifically, a machine learning model). In our client work, we are careful to keep people need to be in the loop at multiple levels, including, but not limited to:

- Strategy: to define a company's overall approach and game plan for moderating content of all types, harmful and otherwise. This strategy needs to be global, considering the need to localize content moderation.
- Governance: to define abusive language, protocols for flagging harmful content, and what to do about it.
- Training: to train an AI model in a way that is free of bias.

- Moderation: to moderate content that AI does not catch; to make judgment calls on content that falls in a grey area; to follow protocols for flagging harmful content and following up with parties who have posted it.

We recommend sourcing a global team that represents the cultural diversity of the markets a business serves. They're needed to ensure that bias does not creep into AI on initial training. And they are needed to make a machine learning model more effective and accurate from the start by applying their knowledge of local cultures and languages.

In addition, the right processes for fighting abusive language go well beyond the methodology we've described for training a machine learning model. Processes include, but are not limited to:

- Protocols for revising content moderation policies. There needs to be a process in place for monitoring changes in standards and updates to the moderation rules.
- Workflows: for instance, how borderline cases or grey areas will be acted on.
- People management: at a higher level, people management encompasses everything from how diverse resources will be recruited to how their performance will be evaluated.

A culture that manages well-being with performance does not happen organically. Processes help ensure consistency and ongoing improvement.

About the Author

Dr. Sreenivasulu Madichetty

Dr. Sreenivasulu Madichetty is a senior AI engineer at Centific. His previous work experience includes being a senior data scientist at Skoruz Technologies in Bangalore, India. He received a Masters in Artificial Intelligence from JNTU Anantapur in 2015 and a Ph.D. from the National Institute of Technology, Tiruchirappalli, in 2021. Before this, he worked as a faculty at IIIT Idupulapaya. His research interests are in the areas of deep learning, natural language processing and social media mining. He has published multiple research papers in journals and conferences as the first author.

References

1. A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media.

2. HateBERT: Retraining BERT for Abusive Language Detection in English.ßß

3. HurtBERT: Incorporating Lexical Features with BERT for the Detection of Abusive Language.

4. Jahan, M.S. and Oussalah, M., 2021. A systematic review of hate speech automatic detection using natural language processing. arXiv preprint arXiv:2106.00742.

5. Ajakaiye, O.O., Ojeka, J.D., Osueke, N.O., Owoeye, G., Ojeka-John, R.O. and Olaniru, O.S., 2019. Hate Speech and Fake News: a study of meanings and perceptions in Nigerian Political Culture. International Journal of Scientific and Engineering Research, 10(5), p.15.

6. Vidgen, B. and Derczynski, L., 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. Plos one, 15(12), p.e0243300.

7. Alexander Brown. 2017. What is hate speech? part 1: The myth of hate. Law and Philosophy, 36(4):419– 468.

8. Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4):1–30.

9. Teresa Quintel and Carsten Ullrich. 2020. Self- regulation of fundamental rights? the eu code of conduct on hate speech, related initiatives and beyond. In Fundamental Rights Protection Online. Edward Elgar Publishing.

10. https://financesonline.com/number-of-twitter-users/

11. https://semiocast.com/half-of-messages-on-twitter-are-not-in-english/

12. https://www.oberlo.in/blog/social-media-marketing-statistics

About Centific

Centific is a global digital and technology services company. We design, build, and optimize human-centric intelligent digital platforms. Our core capabilities are in data, intelligence, experience, and globalization.

We help our clients combat abusive language as part of our AI Practice and AI Enablement Practice. We apply AI to collect various forms of speech relevant in multiple contextual environments. Our methodical practice enables the acceleration of abusive content detection due to the pre-training of models from an abundance of data sources curated and synthesize.