Analysis of aggression detection in social media comments and data cleaning techniques for multilanguage datasets.

Saniya Mukhambetkaliyeva

Abstract

The project aims to study how well logistic regression and Support Vector Machine(SVM) models can predict aggression levels in text from social media comments. Another goal of the project is to analyze different ways to clean the data and how it affects the result.

Introduction

Detecting aggression is crucial for monitoring content in social media, and automation is key in order to streamline the process and make it efficient. I'm personally interested in this topic as I've taken a Sociology of Violence class last semester and liked diving deeper into anonymized cyber harassment and bullying.

The question I'm trying to answer in this project is whether simple Machine Learning algorithms are effective enough in assessing aggression in text without tuning for context. And how much does cleaning up the data affect the results.

Dataset

I used a TRAC dataset that includes raw text extracted from facebook comments, a user ID and an aggression classification(Non-aggressive, Covertly-aggressive and Overly-aggressive). The comments have various lengths and contain emojis, links, mentions etc. I used the development dataset that was included, and it includes 4100 data entries. I chose to not use user ID as a feature, so only the text and labels are extracted.

Feature extraction

First I compared Bag of Words feature extraction and TF-IDF (Term Frequency-Inverse Document Frequency) methods on a dataset that I did not clean.

Method	Bag of Words	TF-IDF
Accuracy(Logistic Regression, SVM)	0.4958, 0.4942	0.5075, 0.5225

Figure 1. Accuracies for Bag of Word and TF-IDF feature extraction methods

The difference is very small but makes sense that TF-IDF performed better because aggressive comments may rely on specific rare words (e.g., insults, slurs) that TF-IDF emphasizes more effectively than BoW. While BoW doesn't differentiate between common words like "the" and gives all words equal weights.

For the latter part of the experiments I've decided to stick to the TF-IDF method.

Dataset cleanup

Upon inspection I've noticed that the dataset was rather messy and had many different types of text like emojis, links, user mentions. And there were many typos in the spelling of the words.

Since the dataset was using user comments from India, despite the dataset being uploaded as english, comments included many hindi words.

No cleanup

Logistic Regression

	precision	recall	f1-score	support
NAG	0.63	0.71	0.67	1257
CAG	0.50	0.52	0.51	1060
OAG	0.56	0.38	0.46	684
accuracy			**0.57**	3001
macro avg	0.56	0.54	0.55	3001
weighted avg	0.57	0.57	0.56	3001

SVM

	precision	recall	f1-score	support
NAG	0.63	0.70	0.66	1257
CAG	0.49	0.53	0.51	1060
OAG	0.58	0.39	0.47	684
accuracy			**0.57**	3001
macro avg	0.57	0.54	0.55	3001
weighted avg	0.57	0.57	0.56	3001

Simple cleanup(removing links, emojis)

Logistic Regression

	precision	recall	f1-score	support
NAG	0.62	0.70	0.66	1257
CAG	0.49	0.52	0.50	1060
OAG	0.56	0.37	0.45	684
accuracy			**0.56**	3001
macro avg	0.56	0.53	0.54	3001
weighted avg	0.56	0.56	0.56	3001

SVM

	precision	recall	f1-score	support
NAG	0.62	0.69	0.66	1257
CAG	0.48	0.53	0.50	1060
OAG	0.57	0.38	0.46	684
accuracy			**0.56**	3001
macro avg	0.56	0.53	0.54	3001
weighted avg	0.56	0.56	0.56	3001

Spelling fixed

Logistic Regression

	precision	recall	f1-score	support
NAG	0.55	0.66	0.60	247
CAG	0.42	0.48	0.45	212
OAG	0.48	0.22	0.30	142
accuracy			**0.49**	601
macro avg	0.48	0.45	0.45	601
weighted avg	0.49	0.49	0.48	601

Manual cleanup

Logistic Regression

	precision	recall	f1-score	support
NAG	0.53	0.67	0.60	246
CAG	0.45	0.47	0.46	211
OAG	0.54	0.25	0.34	142
accuracy			**0.50**	599
macro avg	0.51	0.47	0.47	599
weighted avg	0.51	0.50	0.49	599

SVM	precision	recall	f1-score	support
NAG	0.58	0.65	0.61	247
CAG	0.45	0.52	0.48	212
OAG	0.46	0.25	0.32	142
accuracy			**0.51**	601
macro avg	0.50	0.47	0.47	601
weighted avg	0.51	0.51	0.50	601

SVM	precision	recall	f1-score	support
NAG	0.56	0.64	0.59	246
CAG	0.47	0.55	0.51	211
OAG	0.59	0.30	0.40	142
accuracy			**0.53**	599
macro avg	0.54	0.50	0.50	599
weighted avg	0.53	0.53	0.52	599

Figure 2. Classification reports for Logistic Regression and SVM model trained on a dataset with different cleanup methods.

The first cleanup method was to remove links, mentions, emojis, hashtags. Surprisingly accuracy didn't increase, but even dropped slightly from 0.57 to 0.56. This may indicate that the use of additional data except for text was not affecting the noisiness of data much.
The second cleanup method was to use an external library SymSpell to fix all the typos in the text. The accuracy dropped significantly from 0.57 to 0.49 and 0.51 for Logistic Regression and SVM respectfully. The reason being that many of the hindi words or people's names would get converted to the nearest english word, completely ruining the data.

The third method was simply a manual cleanup where I would fix the typos, while leaving some slang words or hindi words intact. Manual cleanup worked slightly better than Spelling cleanup with accuracy of 0.50 and 0.53, but still worse than the original dataset, and simple cleanup. Strangely enough, when I combined simple cleanup with the manual cleanup, the accuracies for Logistic Regression and SVM were 0.52 and 0.50.
Needs to be noted that for the Spelling cleanup and Manual, I used only the dev dataset with around 4000 data points, due to long computing times for the total dataset, and not being able to manually clean 20 000 lines of text.

Improvements for data cleanup

For the simple cleanup, one potential improvement could involve replacing these extra elements with placeholders (e.g., "URL," "MENTION") instead of removing them outright, as they might provide contextual clues about the comment's tone or aggression level.

To improve spelling cleanup method, a custom dictionary containing commonly used Hindi words, slang, and proper nouns could be incorporated into the spell-checking process to prevent these erroneous corrections.

For manual cleanup to retain original structure future improvements could include semi-automated methods where a model is trained to detect typos while flagging words likely to

be Hindi or slang for manual review. Additionally, leveraging pre-trained multilingual models like mBERT or XLM-R could mitigate the issue of multilingual text without requiring manual intervention.

Final results

So after experimenting with different cleanup methods, I came to a conclusion that no cleanup method is necessary in this instance, as it performed the best.

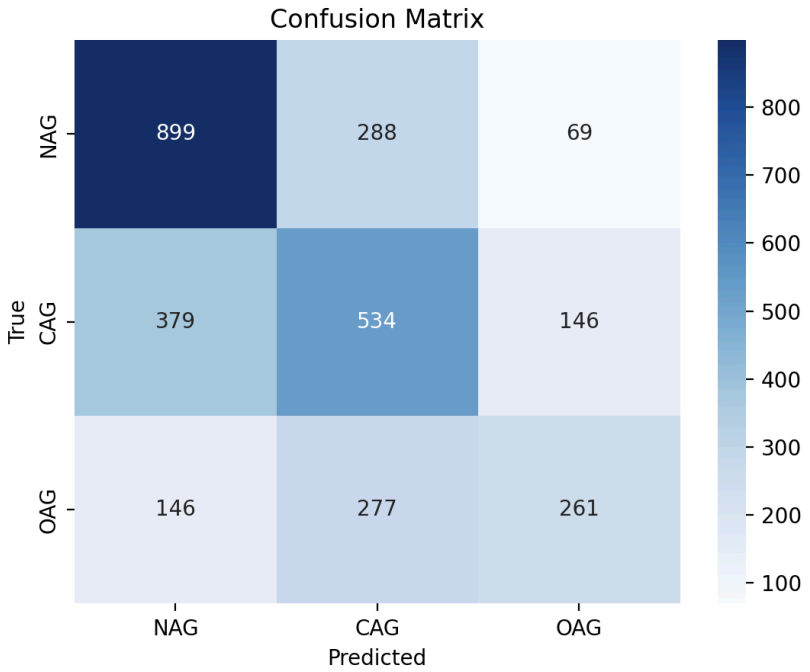

Figure 3. Confusion matrix for Logistic Regression model trained on an original dataset with no cleanup

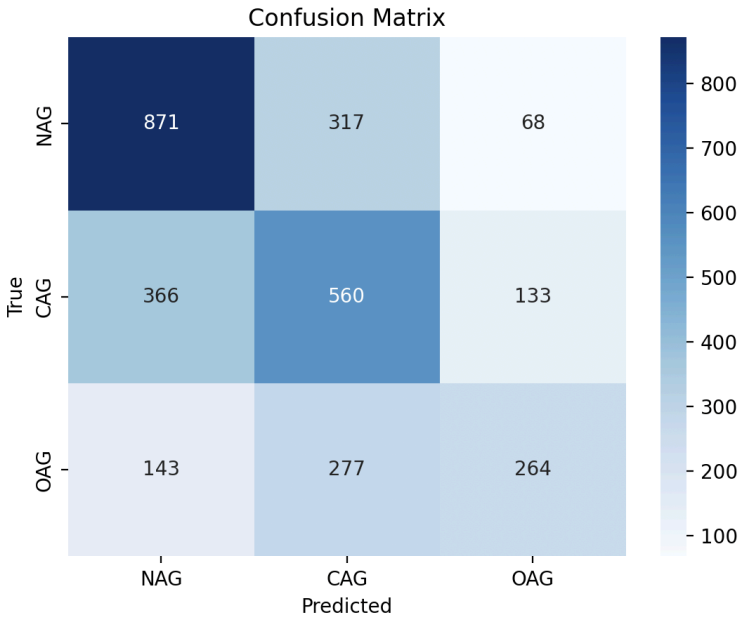

Figure 4. Confusion matrix for SVM model trained on an original dataset with no cleanup

So by analysing the confusion matrix for both Logistic Regression and SVM it's evident that the models are pretty good at detecting Non-aggressive comments with F1 scores of 0.67 and 0.66.

While for Covertly-aggressive comments the F1 scores are 0.49 and 0.51. Lastly for Overtly-Aggressive they are 0.45 and 0.46.

This goes to show that the model struggles with identifying specific types of aggression, while it's moderately good at identifying straight out non-aggressive comments. If we were to instead train the model to identify in binary, i.e only aggressive and non-aggressive, the performance would likely improve. However, such a binary model would lack the granularity needed to differentiate between subtle and overt forms of aggression, limiting its practical applications in moderating nuanced social media discussions.

Using an english dataset from Indian users also clearly shows the difficulties of multi language analysis without human intervention. Results indicate that simple Machine learning models like Logistic Regression and SVM are not enough to do complex analysis of aggression in text.

Currently, aggressive comments on social media are being detected using advanced transformer-based models like BERT or fine-tuned multilingual versions such as mBERT and XLM-R. These models leverage contextualized embeddings and are trained on large, diverse datasets, enabling them to better understand the intricacies of human language, including multilingual and code-switched text. They typically achieve higher accuracies, often exceeding 80% for binary classification tasks and showing significant improvement in nuanced multi-class detection tasks. However, these models require substantial computational resources and

access to high-quality annotated datasets, making them more challenging to deploy at scale in real-time systems.

Sources

1. Kumar, R., Reganti, A. N., Bhatia, A., & Maheshwari, T. (2018). Aggression-annotated Corpus of Hindi-English Code-mixed Data. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (pp. May 7-12). Miyazaki, Japan: European Language Resources Association (ELRA). ISBN: 979-10-95546-00-9.
2. Zhang, Ziqi & Robinson, D. & Tepper, Jonathan. (2018). Detecting hate speech on Twitter using a convolution-GRU based deep neural network.