Implementation and Comparison of Algorithms for Classification of Toxic Comments

Abstract

With the increase in number of online forums, presence of toxic comments has become prevalent as they end up hurting the sentiments of people participating on those forums. In this paper, our goal is to identify and classify toxic online comments. We present different Machine Learning based computational approaches used for classification of toxic comments into six categories. To evaluate, we use a large corpus of Wikipedia Talk Page comments from the Kaggle Toxic Comment Classification Challenge. We also evaluate the efficacy of data augmentation using Easy Data Augmentation in prediction task.

Index Terms

Multi-Label Classification, Wikipedia Talk Pages, Easy Data Augmentation, Back Translation, Machine Learning

I. Introduction

Online communication has enabled humans to participate in varied interactions and engagements instantaneously. It is a great way to network and connect along with discovering valuable information. But with debates, discussions and disagreements negative comments have become almost unavoidable.

Negative online behaviour and conversational toxicity has become tremendously common in social networking sites and online communities. Disrespectful and toxic comments can lead to increased stress, anxiety, uneasiness and low self esteem and users end up participating in online discussions. Due to this, many communities are forced to limit the conversation or shut down user comments completely.

This project focuses on implementing various approaches to detect different types of toxicity like severe-toxic, obscenity-based, threats, insults, and identity-based hate. We perform Data Augmentation and analyze its efficacy. Algorithms were evaluated on a corpus of Wikipedia comments from the Kaggle Toxic Comment Classification Challenge.

The toxic behavior of comments have been labeled by human raters. The data has been visualized through graphical representations in order to detect any outliers and anomalies.

Thus this project tries to predict a probability of each type of toxicity for each comment and analyze the efficacy of Data Augmentation. The 6 types of toxicity in the dataset are: toxic, severe-toxic, obscene, threat, insult and identity-hate. The structure of paper is as follows: Section II gives an overview of the past work done to classify toxic comments. In Section III, we describe our infrastructure that is the description of dataset, and the Exploratory Data Analysis performed. Section IV describes our experimental setup including feature engineering and data augmentation techniques. Further in Section V, implementation details are described: the vectorization methods, machine learning and deep learning models. We compare and analyze our results in Section VI. Finally we conclude and describe our future work in Section VII.

II. Literature review

Table I gives an overview of the base papers we followed as well the methods used by them.

TABLE I
BASE PAPER ANALYSIS

Ref	Title	Description
1	Can We Achieve More with Less? Exploring Data Augmentation for Toxic Comment Classification and Wikipedia	In this paper, the author experimented with Easy Data Augmentation (EDA) and Backtranslation, and implemented Logistic Regression, Support Vector Machine (SVM), Bidirectional Long Short-Term Memory Network (Bi-LSTM) models.
2	Convolutional Neural Networks for Toxic Comment Classification	In this work, the authors compare CNNs against the traditional bag-of-words approach for text analysis combined with a selection of algorithms including kNN, Naive Bayes and Support Vector Machines
3	Challenges for Toxic Comment Classification: An In-Depth Error Analysis	The authors have studied two datasets: Google Jigsaw during Kaggle's Toxic Comment Classification (ref) and a Twitter Dataset by Davidson et al. (2017). The authors have used various word embeddings and applied classifiers, such as Logistic Regression, bidirectional RNN and CNN. Based on these, they created and analysed an ensemble that improved results in detail.

III. Infrastructure

A. Dataset

We have performed our analysis on the ”Wikipedia Toxic Comments” dataset, from Kaggle Comment Classification Challenge. Approximately 158K Wikipedia comments are present with 6 labels for the nature of hate speech - toxic, severe-toxic, obscene, threat, insult and identity-hate. For the dataset, toxic comments of the above categories constitute the positive class while the rest of the lot makes up the negative class. After analyzing the data and plotting the same using matplotlib library, Figure 14 (supplementary) shows that comments range from 1 to 1750 characters and maximum comments contain less than 500 characters. From Figure 15(supplementary) it is clear that the number of words in a comment are between 1 to 400 and most sentences have less than 100 words. Figure 16 (supplementary) depicts that the average word length ranges between 2 to 10 with 5 being the most common word length. Does it mean that people are using really short words in comments? One reason for this can be, stopwords(”the”/”a”/”an”) . Due to which the average word length might be incorrectly left-skewed. Thus, in the data pre-processing state, the,important processes in order to better understand the data are:

- Lowercasing: Like converting ”The” to ”the”
- Remove stopword and punctuation: Such as “the”, ”to”, “of”, ”and” and “a”
- Tokenize: Convert sentences into a list of tokens
- Lemmatize/Stemming: Reduce the inflectional forms of words to its stem, like ”who” and ”whose”

B. Exploratory Data Analysis

After analysing the dataset, we found that the toxicity level is unevenly spread across the classes, indicating that this dataset suffers from class imbalance as seen in Figure 1, and many comments are a part of multiple classes as seen in Figure 2, which indicates the problem is of *multilabel classification*.
There are 159,571 comments in the dataset and the distribution is as follows : [(’toxic’, 15294), (’severe_toxic’, 1595), (’obscene’, 8449), (’threat’, 478), (’insult’, 7877), (’identity_ hate’, 1405)]. Figure 3 depicts correlation which helps us in finding relationship/dependencies.

- ”Toxic” and ”severe toxic” are weakly correlated.
- ”Obscene” comments and ”insult” comments are highly correlated.

Word Cloud - Representation of text data in visual format. The words more often mentioned within a given text are bigger and bolder. We see that the Word Cloud of clean comments in Figure 4 highlights the words talk, page, article, think etc. which are explanatory since the frequency of non-toxic comments is much higher than toxic comments and Wikipedia is a platform for communication, normal words are more prominent. Figure 5 shows the word cloud for ”toxic” comments, words like fuck, hate, nigger and other derogatory terms are highlighted. As we can observe the comments which are severe toxic

Fig. 1. Distribution of comments across various labels

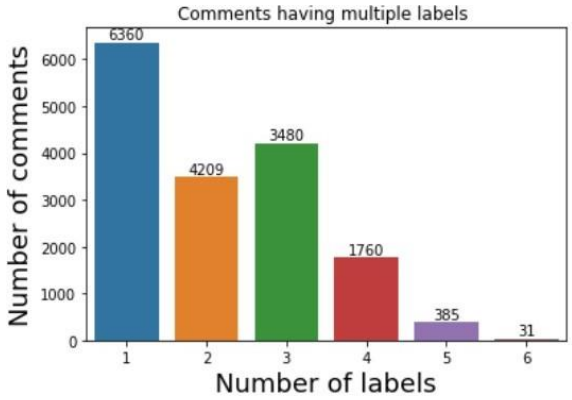


Fig. 2. Distribution of comments with multiple labels

are a super set of toxic comments, thus the same derogatory words repeat again. Threatening words like must die, kill etc are more prevalent. We observe that insulting comments are more related to body shaming (fat), racism (Jew, Nigger), gender based (faggot), thus indicating the various types of categories of insults.

Fig. 3. Correlation plot for the dataset

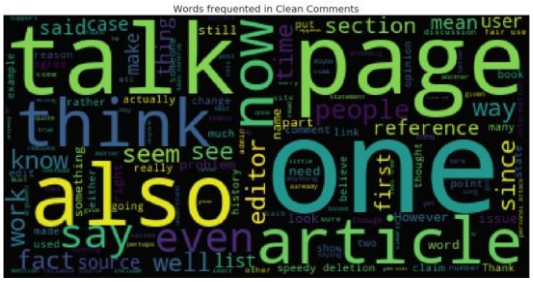


Fig. 4. Word cloud for clean data

Fig. 5. Word Cloud for toxic, severe-toxic, threat, insult comments

IV. Experimental Setup

A. Feature Engineering

Natural Language Processing is usually applied for classification of text data. In this problem statement, categories are assigned to text data according to its semantic meaning.The crucial step of text classification is feature engineering which is the task of creation of features for an ML model from the raw text data we have. As part of the feature engineering task, we use N-grams.

- Unigrams : A 1-gram (or unigram) is a one-word sequence. For example, ”This is an AI and NLP based project”. The unigrams of this sentence are ”This”,”is”,”an”,”AI”,”and”,”NLP”,”based”,”project”. Unigrams do not contribute much to the meaning of the sentence.
- Bigrams : A 2-gram (or bigram) is a two-word sequence. For example, ”This is an AI and NLP based project”. The bigrams of this sentence are ”This is ”,”is an”,”an AI”,”AI and ”,”and NLP”,”NLP based”,”based project”. Bigrams contribute moderately to the meaning of the sentence.
- Trigrams : A 3-gram (or Trigram) is a three-word sequence. For example, ”This is an AI and NLP based project”. The bigrams of this sentence are ”This is an”,”is an AI”,”an AI and”,”AI and NLP”,”and NLP based”,”NLP based project”. As we can anaylze that trigrams are able to bring efficient meaning of the sentence, we are able to capture important word sequences like ”AI and NLP” and ”NLP based project” which maybe further used for various analysis like word auto-complete etc. Figure 6 shows the trigram chart for Wikipedia data before pre-processing. The trigram analysis is more insightful. Phrases like ”thank you for”, ”if you have”, ”my talk page”, ”please do not”, and ”would like to” dominate most comments.

B. Data Augmentation:

1) Easy Data Augmentation [1][7][8]: Easy Data Augmentation is done by performing any of 4 kinds of operations on the sentences in the dataset. The words to be operated upon are chosen at random. The 4 operation performed by us to augment our data are:

- Synonym Replacement(SR): Randomly choose n tokens from the sentence(not stop words), and replace chosen token by one of its synonym.
- Random Swap(RS): randomly choose words from the sentences, and swap their positions.
- Random Insertion(RI): Randomly choose n tokens from the sentence(not stop words), and insert a synonym in place of each word.

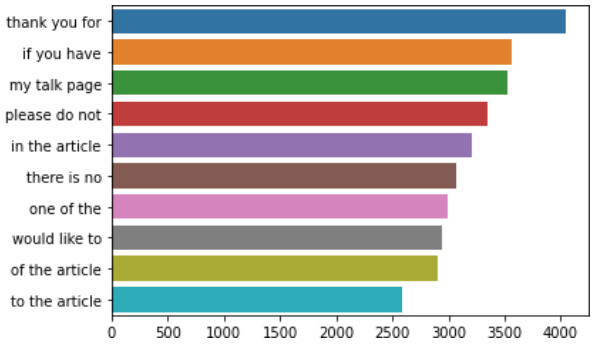


Fig. 6. Trigrams in dataset

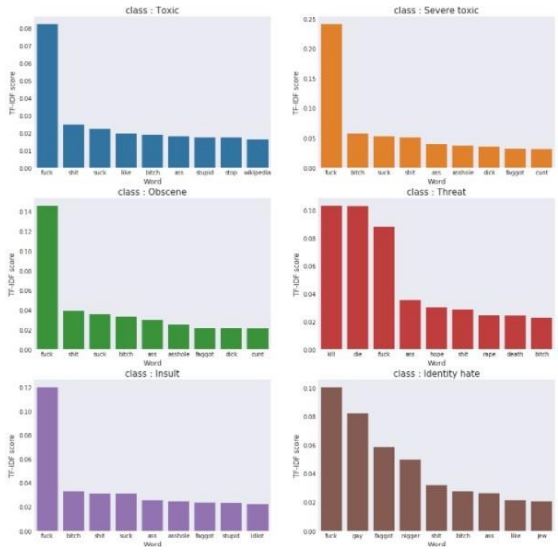


Fig. 7. TF-IDF vs Words relation for each class

- Random Deletion(RD): For each word in a sentence, delete words with probability greater than p.

Each of the above four functions tackle different problems encountered in small datasets. SR helps in maintaining the same syntactic and semantic meaning [6] as the original sentence while generating new words that might not be present and will help the model to learn. RI and RS maintain introduce disorder that helps the model identify and process unknown patterns. RD helps in reducing model over-fitting, by deleting words randomly, it ensures that the model is not ”memorizing” particular patterns and is exploring all features.

We implement EDA such that number of tokens being operated upon, in every iteration, are proportional to the length of the sentence. This is because long sentences are more robust to noise than short sentences.[1] A parameter (a) dictates how many tokens(n) will undergo any of the operations, for simplicity, we take the probability in case of RD as p=a. This serves as a hyperparameter, which can be tuned for better performance. Thus, we define four functions, one corresponding to each operation, which are called uniformly to generate an augmented instance of a given instance.
In our case,

- sample dataset size = 7979 (5% of original dataset)
- n = number of tokens augmented
- l = 10 (avg length)
- a = 0.1
- So, n = a*l = 0.1* 10 = 1
  implying that 1 token will be changed for each instance.
- Also, we generate 5*4 sentences for each sentence So total number of augmented sentences are:
  4 * 5 * 7979 = 159580 data points.

Having built an augmented dataset, We ran all our previously built models on it and compared them to the results obtained on the baseline dataset.

2) Back Translation based Data Augmentation [1]: It is a technique to increase the training data in which the data present in source language into translated into a target language. Then back translation is performed from the target language to source language. The intermediate language used for back translation is french, using which for every sentence one augmented sentence is generated.
Before back translation: ”You’re just at it again!” After back translation: ”You’re just again!”

V. Implementation details baseline and comparative

Here we discuss the various methods and techniques used by us for analysis on both wikipedia dataset and created EDA dataset.

A. Vectorization Methods

- GloVe
  GloVe is an unsupervised machine learning algorithm used to obtain vector representation of words. Here the model is trained on training non-zero entries of “ global word-word co-occurrence matrix”, which illustrates how frequent words co-occur with each other in a given data sample. We have used glove.6b.200d.txt for our vectorization.

- TF-IDF

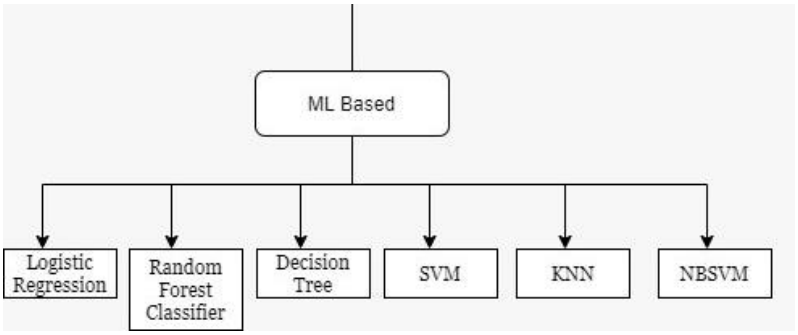
  TF-IDF scores the relative importance of a word based on its semantics. It consists of Term Frequency (number of times a word appears in a document divided by the total number of words in the document) and Inverse Data Frequency (IDF) which is the log of the number divided by the number of documents that contain the word w. Finally the TF-IDF is simply the TF multiplied by IDF. Figure 7 depicts the relation between TF-IDF scores and words in each class.

B. Learning Algorithms

Figure 8 depicts the models we use for our analysis. Results obtained are in Results and Discussion section

Fig. 8. Models used

- Machine Learning based Approaches



1. Logistic Regression
Logistic regression is a statistical ML model used for the purpose of discrete classification(hypothesis statement with more than 1 class). It is a simple technique used in the paper to validate the performance of the data set used, both original Wikipedia comments and data created using EDA techniques. The model is used with “categorical variables” and predicts the probability of occurrence of a variable. The evaluation metrics used are accuracy, precision, recall, F1-score and confusion matrix(supplementary).

2. Decision Tree
Decision tree algorithm, as the name suggests, is a tree-like structure based approach where internal nodes represent the feature, branch determines a decision rule with leaf nodes giving the outcome. DT algorithm is considered as a white box ML algorithm as it shares the internal decision-making logic and the training time is much faster than Neural networks. We have used “DecisionTreeClassifier” from sklearn.tree library with default values.

3. Random Forest Classifier

The next model used is random forest classification algorithm, which is a supervised learning method used for Classification and regression problems. This model creates decision trees on randomly selected data points(bootstrap = True), getting predictions from each tree and selecting the best accuracy-based solution.

4. Support Vector Machine (SVM)

Support Vector Machine, a supervised machine learning algorithm, is used to classify a data point by looking at the extremes of the dataset called 'support vectors' and creating a decision boundary called 'hyperplane' that maximizes the margin between two classes. Thus, this algorithm implies that only support vectors are important while classification and other training samples can be ignored.

5. K Nearest Neighbours (KNN)

K-Nearest Neighbour is a Supervised Machine Learning algorithm based on feature similarity. It classifies a new data point based on it's similarity with existing data points by calculating the shortest distance among it's neighbours. 'K' indicates the number of nearest neighbours the algorithm should consider. This algorithm is effective for large dataset and for classification of non linear data points .

VI. Results and discussion

Following table depicts the results after experimentation, Table II shows results on the baseline dataset, Table III depicts our results on the Augmented dataset while the Table IV mentions the best results as mentioned in analysed base papers.

VII. Conclusion and Future Work

The efforts done for this project is intended to understand, detect and classify toxic comments in online discussions in order to counter abuse and harassment online. Detecting different types of toxicity like threats, insults, obscenity, and identity-based hate is incredibly useful in ensuring that online discussions are more polite and respectful. Possible future scope of this work is to improve our algorithms by performing hyper-parameter tuning, trying more augmentation techniques such as back-translation which will also be helpful in performing toxic comment detection and classification on multiple and mixed languages. Another future research proposal could be analyzing the errors in mixed languages like idiosyncrasies and mitigating those to get better results.

TABLE I
IMPLEMENTED TECHNIQUES AND RESULTS ON WIKIPEDIA DATASET

Parameters / **Technique**	***Accuracy***	***Precision***	***F-score***	***Recall***	***AUC***
GloVe +Bi-GRU +CNN	71.08%	65.76%	-	69.67%	97.10%
LSTM	89.38%	-	-	-	97.98%
TFIDF +LR	96.3%	96.3%	96%	96.3%	-
TFIDF +RFC	66%	79%	64%	56%	-
TFIDF +DT	62%	63%	61%	60%	-
TF-IDF SVM	91.763%	83%	57%	68%	-
KNN	89.5%	70%	29%	19%	-

TABLE II
IMPLEMENTED TECHNIQUES AND RESULTS ON CREATED EDA DATASET

Parameters / **Technique**	***Accuracy***	***Precision***	***F-score***	***Recall***	***AUC***
GloVe + Bi-GRU + CNN	97.62%	-	-	69.67%	97.10%
LSTM	80.46%	-	-	-	75.71%
TF-IDF + LR	95%	92.1%	93%	95.5%	-
TF-IDF + RFC	79%	82%	66%	56%	-
TF-IDF + SVM	90.52%	85%	68%	57%	-

TABLE III
BASE PAPER RESULTS

Parameters / Technique	*Accuracy*	*Precision*	*F-score*	*Recall*	*AUC*
GloVe + CNN	91.2%	-	-	-	97.9%
LSTM	75.55%	74%	77.77%	84%	98.0%
TFIDF + LR	-	84%	77.76%	-	-
TFIDF + SVM	81.1%	-	72.4%	-	-
EDA + GloVe+ Bi-GRU +CNN	-	-	-	-	-
EDA + GloVe + LSTM	-	77.12%	71.09%	-	-
EDA + TF-IDF + LR	-	84.00%	62.91%	-	-
EDA + TF-IDF + SVM	-	74.53%	66.23%	-	-

REFERENCES

1) Rastogi, Chetanya, Nikka Mofid, and Fang-I. Hsiao. "Can We Achieve More with Less? Exploring Data Augmentation for Toxic Comment Classification." arXiv preprint arXiv:2007.00875 (2020).
2) "A Machine Learning Approach to Comment Toxicity Classification" arXiv:1903.06765v1 [cs.CL]
3) Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos. 2018. Convolutional Neural Networks for Toxic Comment Classification. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence (SETN '18). Association for Computing Machinery, New York, NY, USA, Article 35, 1–6. DOI:https://doi.org/10.1145/3200947.32080
4) van Aken, Betty Risch, Julian Krestel, Ralf L öser, Alexander. (2018). Challenges for Toxic Comment Classification: An In-Depth Error Analysis. 10.18653/v1/W18-5105.
5) "Aggressive, Repetitive, Intentional, Visible, and Imbalanced: Refining Representations for Cyberbullying Classification" arXiv:2004.01820v1 [cs.SI]
6) Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level Convolutional Networks for Text Classification". In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 649–657. URL: http://dl.acm.org/citation.cfm?id=2969239.2969312.
7) M. Ibrahim, M. Torki and N. El-Makky, "Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 875-878, doi: 10.1109/ICMLA.2018.00141.
8) Jason W. Wei and Kai Zou. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks". In: CoRR abs/1901.11196 (2019). arXiv: 1901.11196. URL: http://arxiv.org/abs/1901.11196.

VIII. SUPPLEMENTARY

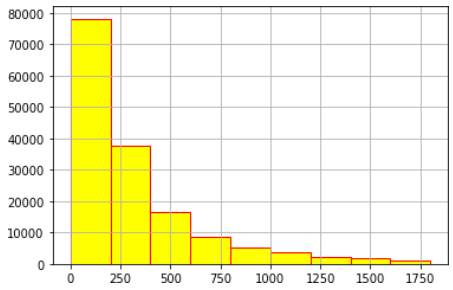


Fig. 14. Characters count in Wikipedia dataset

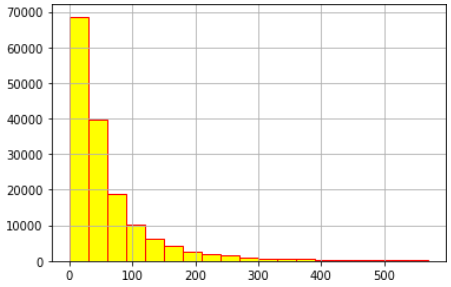


Fig. 15. Words count in Wikipedia dataset

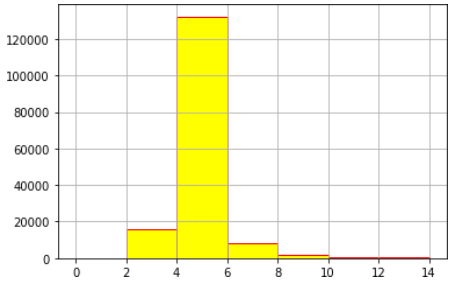


Fig. 16. Average word length in Wikipedia dataset

```
df_train.head(10)
```

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

Fig. 17. Original Wikipedia dataset

```
print(train.head(15))
```

```
                                        comment_text  ...  identity_hate
0   benidorm speak valencian atalan edited contemp...  ...              0
1   benidorm utter valencian catalan redact insult...  ...              0
2   benidorm speak valencian atalan edited insulti...  ...              0
3   benidorm speak valencian catalan editout insul...  ...              0
4   benidorm verbalize valencian atalan edited ins...  ...              0
5   benidorm speak valencian catalan edited insult...  ...              0
6   edited speak valencian catalan benidorm insult...  ...              0
7   benidorm would valencian catalan edited insult...  ...              0
8   insulting speak valencian catalan edited benid...  ...              0
9   benidorm speak valencian catalan edited insult...  ...              0
10  atalan benidorm speak valencian catalan edited...  ...              0
11  benidorm speak valencian catalan edited insult...  ...              0
12  benidorm speak valencian catalan edited exchan...  ...              0
13  benidorm speak valencian catalan edited insult...  ...              0
14  benidorm speak atalan valencian catalan edited...  ...              0
```

Fig. 19. An example from created EDA dataset