Hate Speech Detection: a case of generated data

Marta Campagnoli[1*]

[1*]Data Science and Economis, Text Mining and Sentiment Analysis, ID 928635.

Corresponding author(s). E-mail(s): marta.campagnoli@studenti.unimi.com;

1 Introduction

Hate Speech detection is a text classification task, challenging because of the complex nature of hate language and humans manipulating written text in order to engage in abusive behavior while trying to trick flagging systems; in this work I tried to implement classification algorithms and to extract relevant key phrases of several hate speech categories by working on a synthetic hate speech dataset created for the purpose of further challenging the classification task in order to obtain more robust algorithms for hate speech detection though the semantic and syntactic structure of the text itself.

2 Datasets

The project has been developed by working on two datasets, respectively in English for both the classification and key phrases extraction portions of the work, and French for key phrases extraction exclusively.

English data is based on the Dynamically Generated Hate Speech Dataset from Vidgen et al. (2021) [1] with no duplicate entries; it was generated both to make up for the lack of balanced and big datasets for hate speech detection and to challenge the classification process by using language which is purposefully tricky and misleading. The data is completely synthetic but aims at imitating real hate speech posts with a large variety of targets, such as gender, sexual, racial and religious identities. It was built in four rounds, for a total of 41.144 entries, of which 22.175 (53.89%) constitute hate speech. 26.095 are classified as original, as in text written by annotators to mimic real life, while the rest are perturbation, additional entries obtained by modifying the generated texts in order to introduce challenges in the classification process, such as changing the order of words in order for the label to be switched from hate to non hate or using convoluted and long sentences, with the aim of providing a dataset to train robust hate detection models. Additionally, annotators imitate real social media speech by introducing censored or modified words and neutral speech used as insult. A simple text cleaning function was initially applied to the data; in this phase, stopwords were not removed in order to store the cleaned dataset for both keywords detection and binary classification (*"cleandata.csv"*), while numbers were kept in because hate speech often includes codes used by far right movements. The dataset provides different layers of annotation: we consider *"label"*, which classifies entries as "hate" or "nothate", where the "hate" label was encoded as 0 and 'nothate" as 1 for binary classification experiments, and *"target"*, providing 42 categories, of which 41 specify the target of hateful entries. Using the latter, a dataset for multiclass classification was prepared. The six most frequent categories were extracted for a total of 7.610 entries, the labels encoded according to the values in Table 1 and saved as (*"sixcat.csv"*).

Table 1 Target Encoding

Target	Encoding	Entries
Woman	5	2.035
Black	0	1.961
Jewish	2	1.096
Muslim	3	1.002
Trans	4	792
Gay	1	724

Encoding for the Six Categories Dataset

As for French Data, the french portion of a dataset for Multilingual and Multi-Aspect Hate Speech Analysis [2] was used after the application of a more complex text cleaning function (*"fr_dataset.csv"*).

3 Hate Speech Classification: Methods

The aim of the project is to analyze methods to classify text as hateful or not; in order to do this, experiments were conducted to see which algorithm gave a better performance, starting from binary classification on the entire dataset. In order to see if a more targeted vocabulary leads to improved performance, the algorithms were then trained and tested on the multiclass dataset, focusing on detecting the target of hate speech when the text is already known as hateful.

For each of the two datasets, the analysis was divided in two parts, one based on standard classification algorithms and one based on neural networks. For all classification experiments, text was stemmed using NLTK's Snowball Stemmer, and stopwords were removed with the exception of five words: *"they", "them"*, which are both used in a derogatory context to indicate a specific group of people and in hate speech against trans non-binary people, and *"no", "not", "don't"*, which drastically change the meaning of sentences from hateful to positive (i.e. *"arabs not welcome"* became *"arabs welcome"* before exceptions were implemented).

Because of resource limitations, all experiments have been conducted in Google Colab and are fully reproducible in the Colab Environment.

3.1 Classification: Part 1

In the first part of the analysis, the following algorithms were implemented:

- Logistic Regression, SGD (Stochastic Gradient Descent) Classifier and XGboost using Tf-Idf. Particularly, XGBoost (eXtreme Gradient Boosting) is an algorithm based on decision trees each working on an extracted set of features which consequently builds an ensemble of trees, manages to work on large amount of data and its known to work well on text classification. As for text vectorization with Tf-Idf (*Term Frequency-Inverse Document Frequency*), the choice was made to see whether this statistical method could work well by giving more relevance to words specific to each of the categories to classify. Tf-Idf is in fact a statistical measure based on Bag of Words which gives more relevance to terms which have high frequence but are specific to a certain subset of documents, or, in other words, measures how important a word is to a document in a collection of documents;
- XGBoost and SGD using Word2Vec word embeddings trained on the training data of each dataset with a context window of 2 words because of the limited length of texts, a vector size per word of 300 and considering only words appearing at least 10 times in the corpus. Word Embeddings are vector representations of words in a high-dimentional space that cluster similar words together; they capture the semantic meaning of words and their context, are trained through neural network architectures and have been known to perform better than Tf-Idf. In particular, Word2Vec has been chosen because the model learns vector embeddings of words based on their local context in the corpus; the assumption was that this could encapsulate semantic relationships between words specific to hate speech;
- XGBoost using pretrained Word2Vec and FastText word embeddings through Gensim, which have the same vector size as the trained model, for performance comparisons.

3.2 Classification: Part 2

In the second part of the analysis, two neural network structures were used for classification. A Convolutional Neural Network and a Bidirectional Long Short Term Memory network were implemented; in particular, the Bi-LSTM was chosen after an initial screening showed the model gave better results than both a simple LSTM and a GRU architecture.

Binary and categorical cross-entropy loss functions were used. In particular:

- the CNN architecture comprises 9 Conv1D layers, divided in 3 blocks of 3 layers, each followed by a MaxPooling and Dropout Layer, followed by a final block made of a Flatten, Dense, Dropout and dense Output Layer.
- the BI-LSTM architecture comprises 6 Bi-LSTM layers divided in 2 blocks of 3 layers, each followed by a Dropout Layer, ending with a Dense Output Layer.

For both datasets both architectures were trained using pretrained Word2Vec and FastText embeddings.

4 Keywords and Keyphrases: Methods

Three techniques were implemented to see which keywords and keyphrases could be the most relevant for classification, using the "cleandata.csv" for English and "fr_dataset.csv" after cleaning the dataset:

- for the English dataset only, WordClouds were generated after stopwords removal to visualize the most common words for the Hate and Nothate general categories and for each of the subcategories used in the multiclass classification case;
- KeyBert was used for both languages to automatically extract keywords of one and two words, and using the Keyphrase TfIdf Vectorizer, a vectorizer that can be used to extract features of no set lenght;
- BerTopic, defined as "a topic modeling technique that leverages Hugging Face transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions"[3], was used on both datasets once again using KeyPhrase TfIdf Vectorizer, in order to see if the topics detected could highlight relevant keywords. For both languages, an adaptation of the pipeline suggested by documentation was adopted. In the case of the English Dataset, this part of the work was implemented after extracting a 20% sample of the original dataset proportionally to the "target" annotation. The extracted dataset was stored in the "hatenew.csv" file for reproducibility.

5 Results

The obtained results underline the overall challenges presented by the original dataset, such as the difficulties arising from working on a dataset engineered to trick classification, overfitting in neural networks when working on data obtained by permutation, along with the main differences between working on binary classification, with a broader or more generic vocabulary, and multiclass classification on categories characterized by a more specific set of terms. For each of the algorithms, classification reports have been obtained in order to examine precision, recall and F1 measures, along with confusion matrices; precision is defined as the accuracy of positive predictions, recall is the fraction of positive predictions correctly identified, and the F1 score, which combines the two, can be seen as the percentage of correct positive predictions. These metrics are similarly applicable to multiclass classification.

Additionally, binary cross entropy and categorical cross entropy used in neural network training can be seen as a measure of certainty of the classification outcome.

While specific comments are made on some of values obtained, tables with values of accuracy for all algorithms are reported here; in fact, models have been compared according to their overall classification accuracy, because the objective of hate speech classification is not only detecting hate but also to not incorrectly flagging discussions related to bigoted behavior as hateful because of the mere presence of terms referencing hateful behavior.

Before illustrating results, we remind that the Hate label is encoded as 0 (Negative) and NoHate as 1 (Positive).

5.1 Classification: Part1

As previously anticipated, three algorithms are used in the first part of the work in order to understand whether one had significantly better performance in binary classification; for comparability, the multiclass portion of the work replicates the choices made for binary classification.

- For binary classification, we see that the three algorithms trained with Tf-Idf have similar performances in terms of accuracy, and none goes above 69% with XGBoost (see table 2). Values of precision and recall are between 64% and 72% for both categories in all three algorithms; because of this reason, the analysis through Word2Vec word embeddings was continued discariding Logistic Regression. This first part shows that the dataset is particularly challenging, which will be further underlined by comparison with multiclass results. Training embeddings on the data does not lead to good results; both algorithms obtain a 57% accuracy, and while SGD obtains a 92% recall on the Hate category, it's offset by a 57% precision and a 19% recall for the Not Hate category, meaning that the algorithm simply tends to classify most text as hateful. This could be because the dataset is too little to properly train embeddings, and that a window of only 2 words is not sufficient to properly encapsulate local context, especially when it's been synthetically created to be misleading by using the same language for both Hate and NotHate categories. The last step uses one algorithm only: XGBoost trained on both W2V and Fasttext embeddings provided by Gensim both obtain a 61% accuracy, and values of precision and recall generally lower or comparable with those obtained with Tf-Idf. This could mean this kind of classification needs to be implemented with methods that better encapsulate the context and meaning of words in a very specific setting; FastText is trained on Wikipedia, while Word2Vec is trained on Google News, where words are used in an informative and formal context, completely different from the one of hate speech, even when simulated.
- Multiclass classifcation with Tf-Idf shows that a more specific vocabulary leads to improved results; as shown in table 2, accuracy is as high as 90% for SGD with good results for precision and recall for all categories. Even in this case, training the embeddings doesn't work, with SGD completely misclassifying three of the six categories. Pretrained embeddings work better (78% and 80 % accuracy for respectively W2V and FastText); however, an interesting result can be seen for category 4, corresponding to the "trans" label, which consistently has the lowest values for recall and almost always for precision. As can be seen in A2 for XGBoost trained with Fasttext, where the category has 65% precision and 44% recall, most instances are misclassified as category 5, "woman"; this can be caused by the similar vocabulary shared by the two categories, as hate speech against trans people often makes references to gender related terms such as "woman" or "man".

Classificaation Part 1, Tf-Idf: Accuracy Values			
Classification	XGBoost	Logistic Regression	SGD Classifier
Binary	0.69	0.67	0.68
Multiclass	0.87	0.87	0.90

Table 2 Accuracy results for algorithms using Tf-Idf.

Classificaation Part 1, Word Embeddings: Accuracy Values				
Classification	XGBoost Trained W2V	SGD Trained W2V	XGBoost Gensim W2V	XGBoost Gensim FastText
Binary	0.59	0.59	0.61	0.61
Multiclass	0.64	0.31	0.78	0.80

Table 3 Accuracy results for algorithms using Word Embeddings.

5.2 Classification: Part2

Given the results of Part 1, embeddings trained on the dataset were not used. In general, results show that more complex structures of the networks and alternative word representations, better encapsulating the context and semantic meaning of words, should be implemented. Neural Networks, which have been trained with the help of an earlystopping function, also show the tendency of text data to overfit because of similarities and repetition in the text itself, a problem which is in this case exacerbated by the nature of the large part of the data created by perturbation.

- For binary classification, accuracy results (shown in A1) for all models are comparable to those of algorithms trained using Tf-Idf, with the highest value for the Bi-LSTM with W2V; while some of the values of precision and recall are higher than for Tf-Idf, we can see that training stops as early as the 12th epoch, with values of binary cross entropy higher than 0.56, which means an high number of the probability results obtained through the sigmoid output layer are near the 0.5 threshold and thus uncertain.
- Similarly, for multiclass classification, training stops as early as the 16th epoch, accuracy values are lower than those for Tf-Idf, but categorical cross-entropy goes as high as 0.8910 for the CNN trained with Word2Vec, and the lowest value of 0.5453 for the Bi-LSTM with Fasttext.

5.3 Keywords and Key Phrases

As previoulsy introduced, three techniques were implemented to see whether relevant terms could be extracted from the data. WordClouds were not used for the French Language because of the limited size of the dataset, and on the other end, BerTopic needed the English dataset to be restricted.

5.3.1 WordClouds

Wordclouds, while simply displaying the most frequent words in a vocabulary, were useful to understand the nature of the data: hate speech classification is complicated by the presence of extremely common words used in derogatory manner, and this underlines the importance of context and semantic meaning in this subject. Moreover, this tool was useful for the choice of not integrating sentiment polarities in classification: numerous words connected to hate categories are completely neutral in meaning. An example of this is the *"Jewish"* category: hate speech against Jewish people is often connected to political issues and conspiracy theories about media control and hidden powers behind governments and banks, and common words connected to this category are thus of everyday use or in fact neutral, as in *"media", "money", "control", "Israel", "zionist'*. Wordclouds also confirmed each category is characterized by the presence of specific insults and slurs, and this could explain the better performance of Multiclass classification outside of Word2Vec trained on the dataset, while the Hate and NotHate categories, besides presenting a very high presence of generic words, are also characterized by the same terminology: some of the most common terms in both categories are words such as *"woman", "women", "people", "black", "think"*.

5.3.2 KeyBert

- For English, KeyBert Unigrams and Bigrams for the six hate categories showed results coherent with expectations: one word keywords are strictly related to the analyzed category (for example *"women", "feminist", "patriarchy"*), annotators imitated modifications to avoid detection from algorithms (*"jewssss"*) and slurs are common, especially for the "Black people" category, while two words keywords are generally comprised of a word combined with an insult, such as "feminist bitch". An exception seems to arise in the "gay" category, because topics such as gay pride being overly present in hate speech make it so that bigrams such as "proud gay" are detected as keywords. Because the texts are short, the KeyPhrases option generally detects the same or very similar bigrams or at most trigrams. Keywords detection for the Hate and NotHate categories once again underlines the complicated nature of the data and task; NotHate keywords contain just as much insults and slurs as the ones for the Hate categories; this imitates a common problem in real life, where people tend to repeat such words to explain while they're wrong or to quote someone. All Keywords can be found in the corresponding notebook.
- The scope of the work in French is limited by the difference in categories in the dataset, but an attempt was done to use the most similar as possible, by first extracting keywords for the whole dataset, then for general targets ("individual", "other") and people of arabic and african descent ("arabs", "african_descent"). Differently from the English dataset, the targets seem to be more political ("la gauche", the left), often targeting people with ableist insults ("mongole"). The extracted keywords, however, often find similar translation in english, such as "sale arabe", which translates to "dirty arab", which can be assumed to be similar in intention to "dirty muslims".

5.3.3 BerTopic

For both languages BerTopic was ran removing stopwords and setting a minimum support of 50 documents per topic. Topic -1 is to be ignored because, per documentation, it's typically built on outliers documents. The idea was to use clustering as an alternative method to see which keywords and keyphrases arose from the cluster themselves. Barcharts of keywords and keyphrases for English are reported in appendix, with the disclaimer that they contain an high namber of violent and offensive slurs which have been left uncensored.

- For English, 14 Topics are detected as shown in A1, broadly corresponding to those provided through the "target"annotation in the original dataset. Of these, at least two could be combined in a unique category referring to Black people; the second of these categories suggests the connection of hate speech to discourse regarding police violence, as "officer" is one of the terms characterizing the topic. Additionally, we can notice the "muslim" topic is associated to terms referring to terrorism, that the "immigrants" and "refugees" categories are also very similar to each other, and that a category, called "539s" has been created which seem to contain words misspelled on purpose or words typical of social media speech but of difficult attribution, such as "chad". In general, we can notice once again slurs are typical of almost every category, but that common prejudices are stereotypes are also present, with terms such as "covid" and "corona" being associated to a "chinese" category and "curry" to "indians".
- While the algorithm finds 10 topics, and specific multilingual settings have been used, the results for the French dataset are poorer than the ones in English; this can be because of the need of a larger dataset, or of better knowledge on how to clean text in French. The terms shown in the topics are in some cases extremely generic, such as verb conjugations or connectives. The first two topics could be combined, as they both correspond to ableist insults; similarly "arabe" and "rebeus" (a racist insult) could be joined, and can be seen as related to a "terrorisme" topics; while not many new insights are gained by this method, it can be noticed that even in french hate speech slurs are common.

6 Conclusion

Despite different approaches to classification, the best results for the chosen data are obtained through standard algorithms with Tf-Idf; however, no experiment for binary classification managed to surpass an accuracy of 69%, while performance is improved on multiclass classification, characterized by a more specific vocabulary; further analysis on the vocabulary has confirmed the dataset presents challenges such as the presence of very similar frequent words for both the Hate and NoHate categories, which also correspond to terms which are extremely common and neutral in meaning. For this reason, this work could be improved by using more complex techniques to take context and semantic meaning of words in consideration, while more refined neural network structures should be used to prevent overfitting impacting training. As a last remark, a first attempt to improve the results could be implemented by creating a more restricted vocabulary for the Hate and NoHate categories by removing those words that frequently appear for both labels.

References

[1] Vidgen, B., Thrush, T., Waseem, Z., Kiela, D.: Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection (2021)

[2] Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., Yeung, D.-Y.: Multilingual and multi-aspect hate speech analysis. In: Proceedings of EMNLP (2022)

[3] Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint arXiv:2203.05794 (2022)

Appendix A Images

Classificaation Part 2, Neural Networks: Accuracy Values				
Classification	CNN Gensim W2V	BiLSTM Gensim W2V	CNN FastText	BiLSTM FastText
Binary	0.6775	0.6952	0.6812	0.6923
Multiclass	0.7587	0.8178	0.7652	0.8296

Table A1 Accuracy results for Neural Networks.

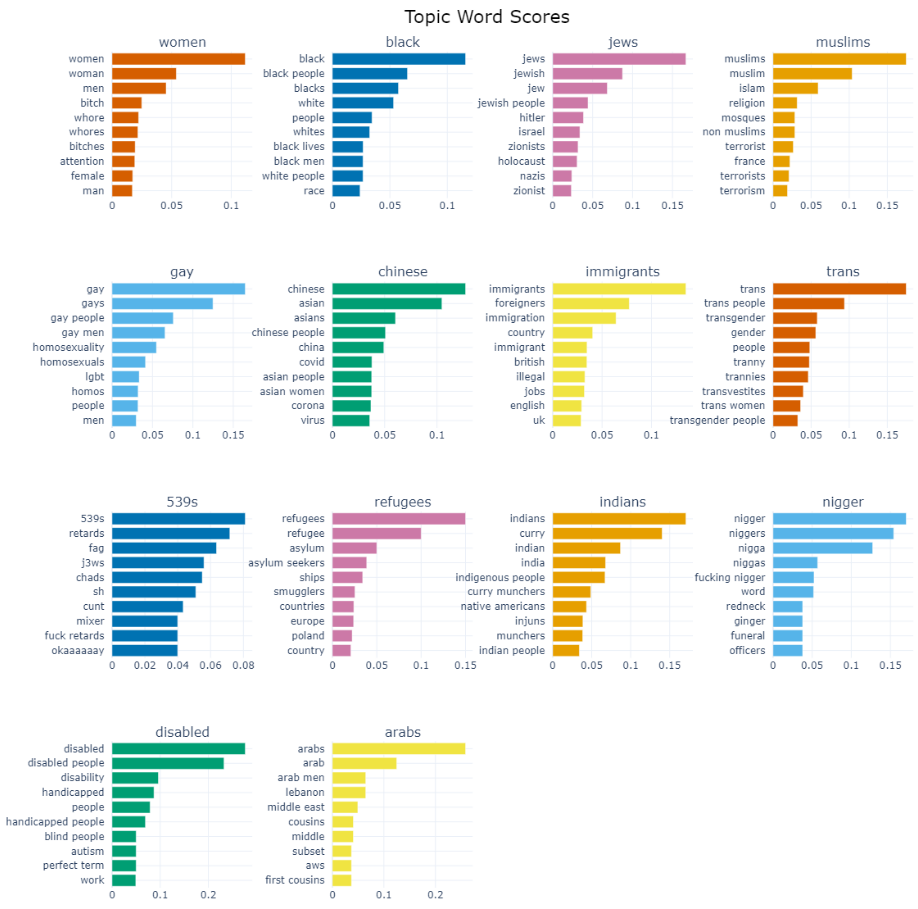

Fig. A1 BerTopic: English

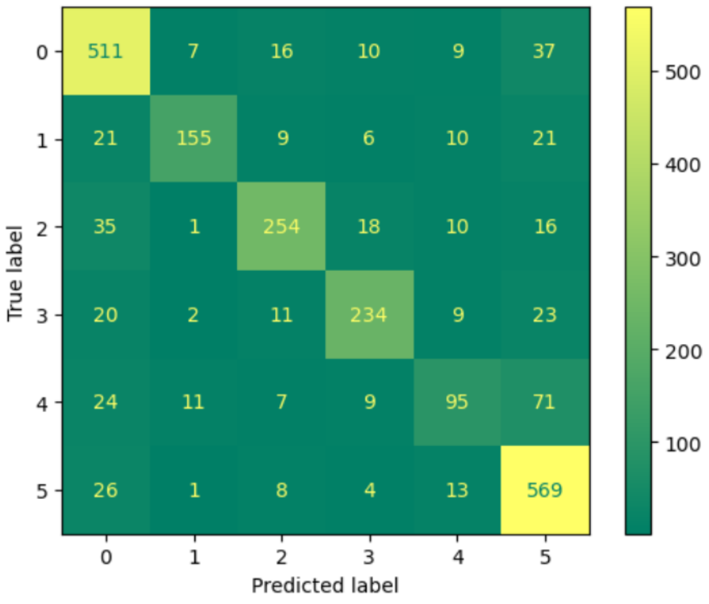

Fig. A2 Confusion Matrix for multiclass XGBoost with Fasttext