CentraleSupélec

Ensemble Learning

Final Report

Detection of Cyberbullying Using Ensemble Methods

Team member	Email
Matteo Salvalaggio	matteo.salvalaggio@student-cs.fr
Sauraj Verma	saureyj.verma@student-cs.fr
Vincent Wilmet	vincent.wilmet@student-cs.fr
Asrorbek Orzikulov	asrorbek.orzikulov@student-cs.fr
Sofya Simakova	sofya.simakova@student-cs.fr

`https://github.com/Asrorbek-Orzikulov/Cyberbullying`
`https://github.com/Asrorbek-Orzikulov/Decision_Trees`

March 11, 2022

CentraleSupélec

Contents

Abstract

This project focuses on cyberbullying in tweets, with two different tasks: one aimed at detecting the type of cyberbullying being done and the other detecting if cyberbullying is present in the tweet or not. Previous work in this field has shown that good prediction performance can be achieved using ensemble learning methods. The two main types of features considered in the past are style-based features, which take into account how information is communicated, and topic-based features, which take into account the content of the communication. Recent publications indicate that style-based features provide better predictive power compared to topic-based features. In this paper, topic-based features are constructed using a bag of words approach, and the style-based features are based on the features found in past research. The style- and topic-based features are combined to boost the performance of tree-based and meta-learning based models, achieving a maximum F1-score on 0.83 on unseen data. Furthermore, the models are also used to compare style-based features to topic-based features, and the results show that better performance is obtained by using style-based features, which is consistent with the literature to date. A deeper analysis of the most relevant features for the models suggests that the links within the tweets contain the most relevant information for the classification algorithms.

Introduction

Cyberbullying detection on social media is an application of Natural Language Processing, also known as NLP, which utilizes machine learning algorithms in order to identify whether a tweet contains cyberbullying against another user.

The majority of studies in this field have concentrated on formal writings, for example books and poems. However, in the modern information age, there is an abundance of brief, colloquial writings. It has become increasingly important to establish whether identification techniques can be applied to these texts. Examples of these writings include emails, forums, blogs and social media.

This project is focused on identifying the nature of cyberbullying on Twitter, a social network that imposes a limit of 280 characters per tweet for their users. [1] The motivation behind the project is to lay a foundation for future social science and mental health research in the context of social media. After proving that the cyberbullying types can be differentiated based on tweets' characteristics and the topics they cover, one can expand this research for example by looking into the effects of these differences and how they affect the social and mental health of online users.

The analysis will be based on both the way in which cyberbullying happens on Twitter, as well as the vocabulary used and topics covered by each of the cyberbullying category. Finally, this analysis will be done using supervised learning algorithms for classification.

Problem Definition

The problem statement of this project is as follows:

Given Twitter data
Use Natural Language Processing (NLP) techniques and classification algorithms
To identify if a tweet poses cyberbullying or not against another user

There are two main constraints on the data used in this project. Firstly, Twitter restricts the number of tweets that can be retrieved from a specific user account/post. Secondly, since the content and topics of tweets are coming from different types of cyberbullying profiles, it is important for the tweets to have similar characteristics for an even comparison. For the cyberbullying detection problem, we are faced with the same problem posed under two different categories:

- **The detection of cyberbullying**: Is a certain tweet actually inflicting cyberbullying or not?
- **The classification of the type of cyberbullying**: If the tweet does contain cyberbullying, what is the nature of the cyberbullying?

The problem then is divided into a multi-class classification problem and a binary classification problem, both of which form a part of the supervised learning sphere of machine learning. The determination of the type of cyberbullying is considered to be a multi-class problem and the detection of the presence of cyberbullying is a binary classification problem. Because of the nature of the problem, the F1-Score was chosen as the measure to maximise for multi-class classification and the Precision-Recall scores were chosen for binary classification.

Methodology

Data set

The cyberbullying Classification data set has been provided to us through Kaggle. The number of tweets in the data set amounted to 47,692 and the composition of the tweets by cyberbullying type can be seen in table 1. As we can see from the table, the tweets are uniformly distributed throughout the categories.

Table 1: Relative share of the data set per Cyberbullying Type

Cyberbullying Type	Relative Share of the Data set
Religion	16,77%
Age	16,76%
Gender	16,72%
Ethnicity	16,69%
OtherCyberbullying	16,40%
NotCyberbullying	16,66%

Pre-processing and feature engineering

Two different approaches to feature selection were considered in this project, and both of these approaches had distinctly different methodologies for data pre-processing. The two approaches are the 'bag of words' approach and the 'style marker' approach, and both of these and their accompanying pre-processing and feature selection steps are discussed below.

Bag of words approach

For the bag of words approach, a large number of pre-processing steps were carried out. These steps are summarised below:

As the first step, the data set was checked and cleaned by filtering and removing links and emojis, because these were considered out of scope for this project. The tweets were then broken down into their basic units - tokens - which convert the string to an array of smaller strings. In the same step, stop-words, the most common words in a language, were removed, because they "aren't significant and distort the word frequency analysis"[2]. In the following steps, these tokens were lemmatized. This means that inflected words were replaced by their non-inflected form. This is done so that words sharing the same base form can be grouped together, as they most likely carry a very similar meaning in a sentence. Furthermore, it serves to further reduce the dimension of the data. In a similar fashion, common synonyms that could be expected in the tweets were replaced by a single synonym.

At this stage, every tweet has been mapped to a very short array of important words. This has been used to determine which words to use as features in the modelling. The next few paragraphs outline how the final set of word features was determined.

For easing the next step, a mapping between words and integer ids was introduced by constructing a gensim dictionary [3]. Based on this, a bag of words, which is "a representation of text that describes the occurrence of words within a document"[4] was created for each cyberbullying category taking into account all their tweets. Furthermore, this enabled the creation of a list containing the words

used by cyberbullying type in a tweet and the number of appearances of the words in that tweet, also known as the frequency.

For each cyberbullying type, this frequency of each used word was accumulated over all tweets for that cyberbullying category and then the words per cyberbullying subdivision were listed in a decreasing order of frequency. This was the basis for deciding which words to include in the feature list. This process was repeated for every new cyberbullying subgroup. This greatly reduced the dimension from thousands of words to only a few hundreds of words.

Style marker approach

The pre-processing for the style features consists of extracting from each tweet: nº of characters; nº of words; frequency of uppercase and lowercase letters, as well as of uppercase and lowercase words; nº of words of each of the lengths between 1 and 20; frequency of each alphabetical letter and digit, as well as of some ASCII symbols symbols such as '/'; count of non-ASCII symbols.

Combined pre-processing and feature engineering

The features from the bag of words approach and the style marker approach were combined to create a comprehensive feature list. No interactions between the features were considered, other than those inherent in the workings of the various classifiers.

Word explorations

To observe the most popular words used in the tweets, we have utilized the package word cloud, bigger is the word displayed in the following image, greater is the frequency that the word is used in the tweets. We can see that some of the most frequent words are profanities, including those based on gender or race, and race-related descriptors, such as 'white' and 'black'.

Figure 1: WordCloud

Models

The models that were considered were informed by the research conducted on cyberbullying detection. After extensive study of the literature and various ML competitions focused on detection of anomalous events, the models considered in this project were CatBoost [5], Random Forests and XGBoost [6].

To begin the classification problem, we use the three models for both multiclass and binary class prediction, and we first start with Catboost. Regarding CatBoost, in order to tune the model, a grid-search was carried out for the parameters *'l2_leaf_reg' , 'border_count', 'score_function'* and *'rsm'*. In addition, the *'boosting_type'* was set to 'Plain', which is optimal for large datasets. In order to ensure no over-fitting, the parameter *'random_strength'*, which sets the amount of randomness to use for scoring splits when the tree structure is selected, is set to 1. In addition, the depth of each iteration was kept to 6. Note that the maximum value is 16, but since CatBoost works with symmetric trees, each new level of depth increases the ramifications exponentially, increasing run-time and possibly over-fitting the model.

Next, it was decided to implement the XGBoost classifier. XGBoost is usually a highly robust ensemble algorithm that requires minimal amounts of parameter tuning to make it achieve a high predictive score. Regarding overfitting, neither of the meta-ensemble models (XGBoost and CatBoost) had this problem since both models implement iterative pruning during their training phase, throwing out features that have little to no impact on classifying the target variable. Furthermore, hyperparameter tuning was attempted for both models. However, this procedure did not result in a significant increase in the model performance.

Next, in order to tune the Random Forest classifier, a grid-search was carried out for the parameters *'max_depth' , 'criterion', 'class_weight', 'min_samples_leaf'* and *'n_estimators'*. In order to combat the class imbalance in the data, the *'class_weight'* parameter was set to *'balanced_subsample'*, which resulted in the highest F1 score for multiclass classification and accuracy for binary classification. In order to prevent over-fitting, the *'max_depth'* parameter was controlled. Lastly, the over-fitting was also monitored by performing a 5-fold cross validation. Overall, the Random Forest differential between the test scores and the CV scores was relatively small, suggesting the model did not overfit.

Evaluation

The following section lays out the results of the project along with how the models were evaluated. For evaluating both the problems at hand, we use F1 score for multiclass classification and Precision-Recall for binary classification, alongside with accuracy for ease of interpretation.

The models were split into two parts - a training set consisting of 80% of the data, and a test set consisting of the remaining 20% of the data. A 5-fold cross validation was performed to ensure that the models do not overfit and would perform well on unseen data. After training the models on the training set, the model performances on the test set were evaluated.

The main results are shown in tables 2 and 3.

As observable by the scores, CatBoost yielded the best results for both multiclass and binary classification. In order to better understand the reason why CatBoost performs better, we analyze its Precision-Recall ratio and how well the model performs on detecting sensitive topics. Figure 1 shows the classification report of CatBoost on the test set. What makes CatBoost perform better

	CatBoost	RandomForest	XGBoost
CV Mean F1-Score	0.833	0.813	0.827
Test set F1-Score	0.827	0.819	0.823
CV Mean Accuracy	0.831	0.808	0.824
Test set Accuracy	0.825	0.82	0.825

Table 2: F1-score of the three models tested (Multiclass Classification)

	CatBoost	RandomForest	XGBoost
CV Mean F1-Score	0.727	0.724	0.721
Test set F1-Score	0.722	0.726	0.730
CV Mean Accuracy	0.876	0.810	0.824
Test set Accuracy	0.871	0.812	0.810

Table 3: F1-Score of the three models tested (Binary classification)

than other classifiers is its high precision to recall score, with a high precision rate on age, ethnicity and religion. What it means is that CatBoost's prowess to detect cyberbullying in terms of age, ethnicity and religion is quite stronger compared to Random Forests and XGBoost, with a high recall rate showing that CatBoost detects cyberbullying types much better and is further able to accurately distinguish their nature.

	precision	recall	f1-score	support
age	0.98	0.97	0.98	1619
ethnicity	0.98	0.94	0.96	1603
gender	0.89	0.82	0.85	1587
not_cyberbullying	0.59	0.56	0.58	1585
other_cyberbullying	0.62	0.73	0.67	1585
religion	0.95	0.95	0.95	1559
accuracy			0.83	9538
macro avg	0.84	0.83	0.83	9538
weighted avg	0.84	0.83	0.83	9538

Figure 2: Classification Report for CatBoost (Multilabel Classification)

The results are consistent with those in [7] - using style markers as features result in a more accurate classification. The results also confirm the initial hypotheses in this research project - using both style markers and bag of words features result in a stronger classifier compared to using only one of them.

Table 4: Bag of words - % of Total word count as features

	158 words 20%	**350 words 30%**	**593 words 40%**	**913 words 50%**	**1487 words 60%**
F1 score - Bag of words only	0.435	0.489	0.515	0.519	0.526
F1 score - All features	0.718	0.740	0.730	0.733	0.730

The results in table 4 show that the number of words to use as features in the bag of words approach could be reduced considerably while still maintaining a similar F1 score. When only using the bag of words feature set, there is a monotonic increase in the F1 score as the number of words increase. However, when using both the style markers and the bag of words features, a larger number of words in the bag of words did not lead to a monotonic increase in the F1 score. In fact, the optimal number of words to use was found to be 350 words, which corresponds to the words whose cumulative frequencies make up 30% of the total word count, as described in the methodology section. This discovery meant that the dimension of the features could be reduced considerably, as the first set of features used in this project comprised 1487 word features (60%).

Additionally, the most important features for the CatBoost model, as given by the method *model.feature_importance_*, are depicted in Table 5.

Table 5: CatBoost model Feature Importance

Feature	/	Words in tweet	Word length = 3	Lower case words	Upper letters
Relative Importance	17.85	10.62	3.08	3.05	2.3

On the same line, Antonio Castro and Brian Lindauer [**?**] include as well the words per tweet and frequency of lowercase words in their top 3 most important features. Additionally, note that the top features are part of those considered style features. This is also in line with the work presented by Green et al. [8]
Furthermore, it was decided to compare the most important features of CatBoost with the fundamental ones of XGBoost. The table below contains the most important features for XGBoost.

Table 6: XGBoost model Feature Importance

Feature	characters per tweet	nb_capitalized	nb_lower	a	t
Relative Importance	550	352	297	283	279

Conclusions

The overall best result with an F1-Score of 0.83 achieved by using CatBoost proves that the tweets can successfully be detected for aspects of cyberbullying.

Future work

The work in this paper was limited to the number of tweets obtainable from the Kaggle dataset. An avenue for further research would be to use tweets that can be retrieved using an API directly

from Twitter. Through this, one can establish whether the bag of words approach remains effective over a larger data set, since a dataset with maybe some data imbalance can greatly impact the choice of features to use in the bag of words approach.
Term frequencies were used to calculate which words to use as features in the bag of words approach. Another methodology that could have been followed is the Term Frequency - Inverse Document Frequency (tf.idf) measure. This measure increases or decreases the importance of words based on the times it appears in other documents/tweets of other types.
In future work, additional features can be taken into account to capture even more information on which the algorithms will be trained and tested. For example, the use of emojis and a deeper analysis of links contained in tweets can be taken into consideration if present in the tweets. Furthermore, features that look at writing style and topic features jointly, like N-Grams, could be introduced.

References

[1] Dominique Jackson. Know your limit: The ideal length of every social media post. https://sproutsocial.com/insights/social-media-character-counter/.

[2] Taranjeet Singh. Natural language processing with spacy in python. https://realpython.com/natural-language-processing-spacy-python/.

[3] Radim Řehůřek. corpora.dictionary – construct word-id mappings. https://radimrehurek.com/gensim/corpora/dictionary.html.

[4] Jason Brownlee. A gentle introduction to the bag-of-words model. https://machinelearningmastery.com/gentle-introduction-bag-words-model/.

[5] Inc. Catboost. Catboost - open-source gradient boosting library. https://catboost.ai.

[6] The XGBoost Contributors. xgboost - optimized distributed gradient boosting library. http://xgboost.readthedocs.io.

[7] Rachel M Green and John W Sheppard. Comparing frequency-and style-based features for twitter author identification. In *The Twenty-Sixth International FLAIRS Conference*, 2013.

[8] Luke Chen, Eric Gonzalez, and Coline Nantermoz. Authorship attribution with limited text on twitter, 2017.