IN SUPREMÆ DIGNITATIS
1343
UNIVERSITÀ DI PISA

Human Language Technologies

Project 2023/2024

Cyberbullying Classification by ChunkDevs

Arcangelo Franco (584174), Salvatore Ergoli (660527), Marco Sanna (660521)

Contents

1 Introduction

The project that has been presented for the Human Language Technologies course is the implementation of an NLP model which aims to solve a classification task on tweets from the social media X (before named Twitter), which can potentially be considered as cyberbullying acts or offenses.

1.1 Motivations

The purposes of this project choice are based on a practical application of the theoretical and methodological concept learned from the HLT course, the relevance from a social and psychological point of view of a phenomenon which has been drastically increased after the Covid-19 pandemic situation and on the objectives that are given by the authors of the dataset, which can represent a stimulating challenge for the group.

1.2 Dataset and goal of the project

The Dataset is a *.csv* file and it has been taken from the website *Kaggle* at the following link: Cyberbullying Classification Dataset.

It contains more than 47000 tweets labelled according to the class of cyberbullying. The data has been balanced to contain almost 8000 labels of each class. These tweets either describe a bullying event or are the offense themselves. This dataset has been labeled by humans.

The first phase of the projects is to create a **binary classification model** which must be able to identify a tweet as a cyberbullying act or not. Considering that the dataset labels are set for multi-classification task, it will be necessary to create a new binary feature which labels would be like 0 (False) for non-cyberbullying message and 1 (True) for cyberbullying detection and to consider the fact that the classes will be very umbalanced.

Therefore, the next step is to also detect the type of the discriminatory act and to deal with a **multiclassification task**, by associating the correct label to the respective message among the available ones:

- Age;
- Ethnicity;
- Gender;
- Religion;
- Other type of cyberbullying;
- Not cyberbullying.

2 Literature review

In order to obtain the necessary competencies needed for this project, we consulted different NLP theoretical and project resources, in addiction to the academical material released during the course of Human Language Technologies at the University of Pisa.

With respect to the NLP disciplinary environment, "Speech and Language Processing"[1] has been considered as the main theoretical reference point about the techniques, models and most modern challenges of this research field. Moreover, for the text processing phase, we have also taken into account "Testo e computer"[2].

Considering the practical aspects of this project, the first consulted paper was the one written by the authors of the dataset[3] in order to get a better insight among the data and the model that have been already implemented for this task, for example XGBoost (94% of accuracy), but taking in consideration the differences in the data preparation, as the Dynamic Query Expansion and the removal of the not cyberbullying class during the multi class classification task. Moreover, some project presented by people who participated at the Kaggle challenge have been analysed, like the one who got the best rate in the website and the best accuracy in multi class classification task (94%), using Roberta and LSTM, but removing the other cyberbullying label.

3 Data understanding and preparation

3.1 Data understanding

This stage includes collecting initial information, describing the dataset, exploring the content to identify patterns and verifying the quality to ensure it is suitable for further analysis. In the context of our cyberbullying classification task, understanding the information helps us to identify the key features and characteristics that will drive the model's ability to classify different types of cyberbullying accurately. This process includes analyzing textual information, such as tweets, and understanding the distribution and semantics of words and hashtags used within them.

Examining the wordclouds in the figure 1, each class reveals notable differences in the semantic connections to the concept of cyberbullying. Specifically, the classes related to gender, religion, age and ethnicity display words that are closely associated with cyberbullying acts. In contrast, the classes labeled as ”other cyberbullying” and ”not cyberbullying” do not exhibit such clear semantic connections. This observation suggests that distinguishing between ”other cyberbullying” and ”not cyberbullying” will likely be more challenging due to the lack of specific, identifiable language patterns in these categories.

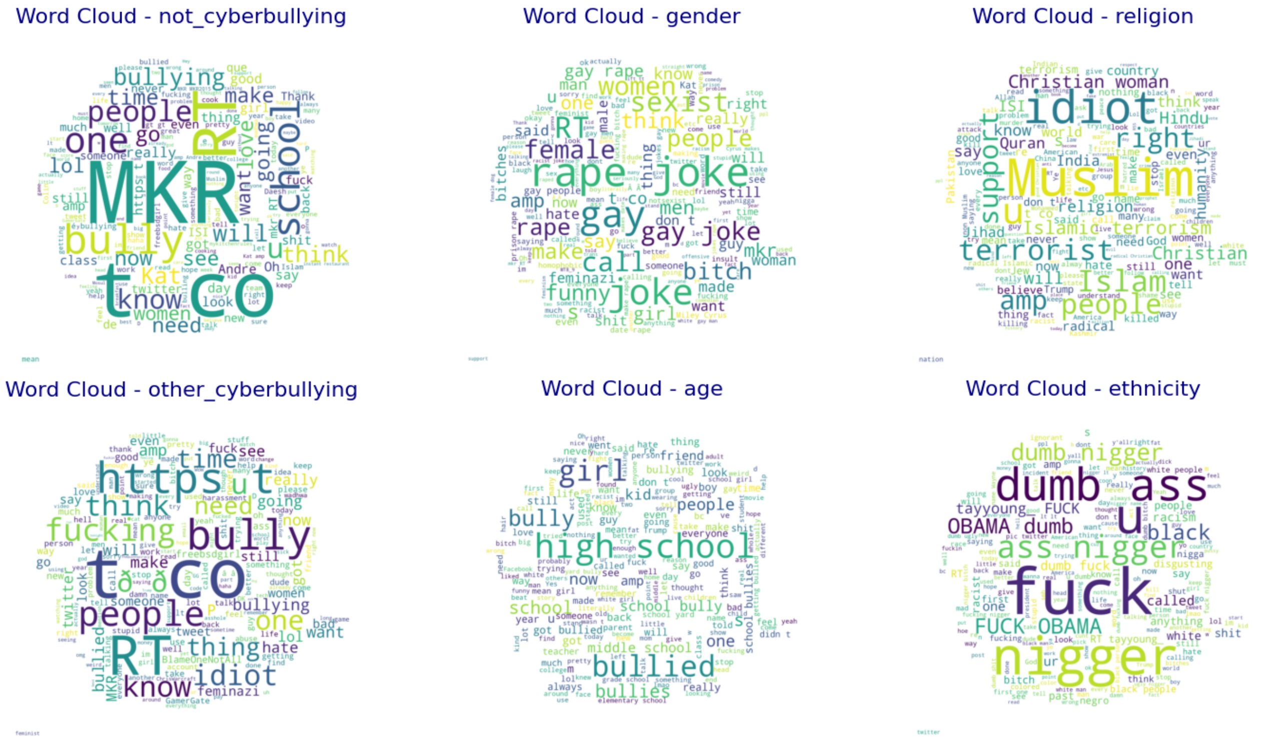


Figure 1: Word cloud for all the six labels

In addition, we attempted to treat hashtags as a separate feature for classification by extracting words with the hash symbol into a new column. However, we encountered two significant issues with this approach, highlighted by the data understanding phase:

1. The prevalence of hashtags in the tweets was much lower than expected;

2. Some hashtags, like ”MKR” (an acronym for ”My Kitchen Rules,” a British TV series), appeared across

multiple classes, including gender, other cyberbullying and not cyberbullying. This lack of specificity meant that these hashtags could not effectively differentiate between the classes.

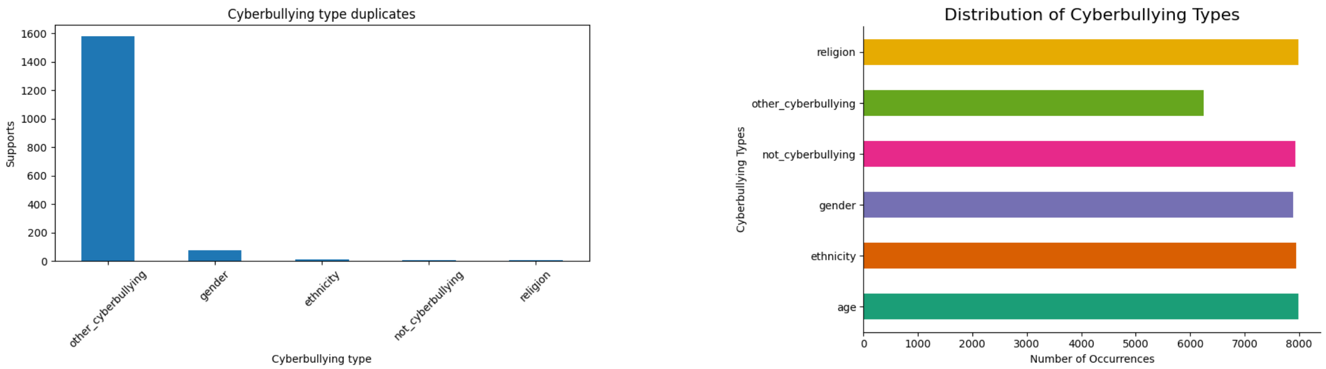
Given these challenges, we decided to abandon the use of hashtags as a feature for classification, as they did not provide the necessary discriminative power for our analysis.

3.2 Data preprocessing

In this phase, we decided to create two version of the original dataset, the first one with all the tweets, and the second one with only english texts. Both of them will be divided in dev-set and test-set, but only the tweets of the first one will be normalized during the text cleaning phase, the test set will be leaved totally untouched.

3.2.1 Duplicates analysis

The first step of the data processing phase was to check if there was the presence of duplicated tweet inside the dataset. In fact, we found that approximately 1600 duplicated messages were existing in the data collection, but with different cyberbullying type label (other cyberbullying and another one). Therefore, we removed them. However, after this process, the classes were no more balanced.



(a) Number of duplicates for each class

(b) Distribution of cyberbullying types

Figure 2: Duplicates and distribution after removal

We found that almost the totality of the duplicates were belonging to the other cyberbullying label. This is due to the fact that some tweet that were labeled as, for example, gender cyberbullying, were also labeled as other cyberbullying. In order to lose as less data as possible, the re-balancing of the classes will be done after the data splitting and cleaning.

3.2.2 Multilanguage dataset: splitting and dev-set cleaning

For the multilanguage dataset, we splitted in dev-set and test-set, using the 90%-10% proportion of the original dataset. After the separation of the data, we proceeded with the cleaning of the dev-set tweets.

X has become a significant platform for real-time communication, with users posting short and concise messages. The nature of this messages is unique due to several factors:

- **Character limit**: tweets are limited to 280 characters, encouraging users to be succinct and often leading to the use of abbreviations, acronyms and slang. However, we found that these tweets in the dataset have different lengths in terms of characters;
- **Hashtags and mentions**: users frequently employ hashtags (#) to categorize topics and mentions (@) to direct tweets to other users, they can be used as a part of the sentence, or at the end or beginning of the tweet. We believe that mentions to other users are not very informative to our type of task, and then we

decided to remove them, whereas the case of the hashtag is more particular when dealing with messages from twitter. The hashtag are used as words inside the sentences, so we decided to only remove the ”#” symbol while keeping the word after that;

- **Emojis and Emoticons**: emojis and emoticons are common, and can be useful for sentiment analysis but unfortunately we cannot deal with this kind of character. A possible solution would be to map the emojis with a special token, in order to try to keep information about the sentimental dimension of the message.
- **Informal language and other extralinguistic elements**: tweets often contain informal language, typos and non-standard grammar or slang terms. They can contain also elements as hyperlinks, non-ASCII symbols or other external elements that should be normalized.

Given these characteristics, a preprocessing pipeline has been adopted to try to improve the performance of the classification. This pipeline involves several steps:

1. **Removal of Hashtags, Mentions, Tabulation Characters, Non-ASCII Characters, Repeated Characters, Emojis, Emoticons, Hyperlinks using *regex***:
    - Regular expressions (regex) whit the python *re* library are employed to clean the text by removing:
        - **Hashtags symbols** and **mentions** to focus on the core content, as written before, we removed the entire mentions and only the hashtag symbol;
        - **Tabulation characters**, **non-ASCII characters** to standardize texts and avoid codify problems during the input transformation for the machine learning models, so we removed them;
        - **Repeated characters**, which can be a form of emphasis or error, are considered as different tokens by a ML model. Therefore, we normalized the repeated consecutive characters, by reducing to two the number of repeated letters if there is a pattern with more than two consecutive characters;
        - **Emojis** and **emoticons**: since we did not know how to deal with this specific kind of character, we decided to remove them, using the *emot* library;
        - **Hyperlinks** and **HTML tags** to eliminate uninformative content.
2. **Handling Contractions with a custom glossary**:
    - A custom glossary in form of a python dictionary is created to expand contracted forms (e.g., ”don’t” to ”do not”) but also common abbreviations used in social media (e.g., ”ppl” to ”people”) which helps in maintaining the clarity and meaning of the text.

After the cleaning process, we placed the cleaned tweets in a new column of the dataset and we found that some of the processed tweets where completely removed (i.e. tweets with only emojis or links). Therefore, they should be considered as missing values and they have been removed. Moreover, some of the processed tweets are become equal to other tweets. This is due to the fact that some of them were differentiated only by links, mentions emojis etc. Therefore, we decided to remove the duplicated messages.

After that, we check again if they were some tweets in common in the dev-set and the test-set, in order to avoid the presence of shared texts on both of the sets. Finally, we balanced again the classes of the dev-set and the test-set and saved them into two new *.csv* files.

Summarizing, we splitted the dataset in dev and test set (38000 tweet DEV, 4800 tweets for TEST), and then we procedeed in cleaning only the dev set. After that, we balanced again the dataset, in order to mantain the same distribution of the classes. In the end, the dataset has 3 new columns:

1. *cyberbullying_type_bin*: 0/1 labels (not cyberbullying/ cyberbullying) for binary classification,

2. *cyberbullying_type_multi*: numerical labels for the multi class classification task,
3. *tweet_text_cleaned* : the column with the cleaned text, only for the dev-set.

3.2.3 English dataset: language filtering

Starting again from the original dataset, we performed language detection on the text data, using three different libraries: 'langdetect', 'fasttext' and 'lingua'. In order to filter the english tweet we decided to keep only the tweets that are classified as english by every library. This step ensures the focus on English content for consistent analysis.

After the language filtering, we splitted the data into dev-set and test-set, cleaned only the first dataset, re-balanced again the classes and saved the dataset in two new files.

3.2.4 Basic text processing approaches

In addiction, we tried different **basic text processing approaches**, in order to create new features that could be used for classification and put in a new column inside the dataset. However, since they did not resulted as efficient properties for the models performances, we decided to not keep them.

1. **Tokenization and Sequence Segmentation with *NLTK***:
   - *NLTK* (Natural Language Toolkit) is used for splitting the text into individual words or tokens and segmenting sequences. This step is fundamental for breaking down the text into manageable pieces for further processing.
2. **Lemmatization, Stemming, POS Taggings with *SpaCy***:
   - **Lemmatization**: reducing words to their base or root form to ensure consistency in word usage (e.g., "running" becomes "run");
   - **Stemming**: similar to lemmatization but involves cutting off word endings to reach the root form;
   - **POS Tagging (Part-of-Speech Tagging)**: identifying the grammatical parts of speech (nouns, verbs, adjectives, etc.) to understand eventually the syntactic structure of the text.

4 Classification

4.1 Models chosen and split of the dataset

For the task of classification, a range of models has been selected to ensure comprehensive evaluation and robust performance. Initially, simpler baseline models are employed, such as **Bernoulli Naive Bayes** for binary features and **Multinomial Naive Bayes** for multiple features. These models are known for their simplicity, efficiency and effectiveness with text data, serving as strong starting points for the classification task. **Logistic Regression** is also utilized as a fundamental and widely-used linear model, providing a solid benchmark due to its interpretability and competitive performance on text data.

Moving beyond the baseline, more advanced models like XGBoost, Long Short Term Memory (LSTM), and Support Vector Machines (LSTM) are introduced. **XGBoost** is particularly valued for its performance and efficiency, excelling in handling structured data and providing robust results even with complex datasets. **LSTMs**, with their ability to capture temporal dependencies and contextual information over sequences, are well-suited for text data where maintaining temporal relationships among words is crucial. **SVMs**, known for their effectiveness in high-dimensional spaces, offer robustness against overfitting and perform well in text classification tasks.

To leverage the latest advancements in natural language processing, transformer models like BERT and RoBERTa are employed. **BERT**, with its bidirectional context understanding, achieves high performance across various NLP tasks, making it ideal for the nuanced task of cyberbullying detection. **RoBERTa**, an optimized version of BERT, benefits from more extensive pre-training, leading to improved generalization and performance.

For the multilingual dataset, we used a **multi-language** version of **BERT** with a sustainable number of parameters, in order to not occupy too much memory resources

This diverse selection of models ensures a thorough exploration of the cyberbullying classification task, from simple yet effective baselines to advanced and state-of-the-art methods. By employing this range of models, we can evaluate the strengths and limitations of each approach, ultimately aiming to identify the most effective model for detecting cyberbullying.

As written before, the dataset was split into **dev set** (90%) and **test set** (10%), as a standard practice in machine learning. The development set, consisting of 26,238 tweets, provides ample data for model tuning and validation processes. This sizable dataset allows for robust experimentation and refinement of models before final evaluation on the test set. The test set is evenly balanced for multiclass evaluation, with 487 tweets per class. This balance ensures a fair assessment of model performance across all categories. However, for the binary cyberbullying classification task, the test set is imbalanced, comprising 2435 cyberbullying tweets and 487 not-cyberbullying tweets. This imbalance necessitates careful consideration during model training and evaluation to prevent biased predictions towards the majority class (cyberbullying).

4.2 Feature engineering for baseline and advanced models

In text classification, feature engineering is a crucial step that transforms raw text data into numerical representations that machine learning algorithms can understand. For both baseline models (Logistic Regression and Naive Bayes) and advanced models (LSTM, XGBoost, SVM), we used two popular techniques: Bag of Words (BoW) and TF-IDF.

The **Bag of Words** model is a simple and widely used method for text representation in natural language processing. It involves creating a vocabulary of all unique words present in the corpus (the collection of all text documents). Each document is then represented as a vector of word counts. The length of the vector is equal to the size of the vocabulary. Each position in the vector corresponds to a specific word and the value at each position is the count of that word in the document.

TF-IDF is an extension of the Bag of Words model that adjusts the word counts by the importance of each word in the corpus. It combines two measures:

1. **Term Frequency (TF)**: the number of times a word appears in a document, divided by the total number of words in that document. This gives a measure of how frequently a word occurs in a specific document;
2. **Inverse Document Frequency (IDF)**: the logarithm of the total number of documents divided by the number of documents that contain the word. This gives a measure of how important a word is in distinguishing documents from each other. Rare words across documents get higher scores, while common words get lower scores.

The TF-IDF score for a word in a document is the product of its TF and IDF scores:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

4.3 Hyperparameters

The Naive Bayes (Bernoulli and Multinomial NB) and Logicistic Regression models are used with different feature extraction methods (already introducted in section 4.1), such as Bag of Words and TF-IDF. Considering the dataset's characteristics — balanced for the multiclass task but imbalanced for the binary task—the hyperparameter choices for Logistic Regression and Naive Bayes are customed to address these specific condition effectively.

In the case of Naive Bayes, BernoulliNB for binary features (like that created by Bag Of Words) uses parameters like **alpha** for smoothing, addressing zero probabilities for certain features and **binarize** to convert continuous features into binary form, essential for text-based classification like cyberbullying detection. The **fit_prior** parameter provides flexibility to handle class imbalance by either learning the prior probabilities from the data or using uniform priors.

For MultinomialNB, the **alpha** parameter range ensures robust handling of varying feature frequencies, and **fit_prior** options help the model adjust based on class distributions.

Model	**alpha**	**fit_prior**	**binarize**
Bernoulli with BOW (binary)	1.0	True	0.0
Bernoulli with BOW (multiclass)	0.5	True	0.0
Multinomial with TF-IDF (binary)	0.5	False	Not added
Multinomial with TF-IDF (multiclass)	0.1	False	Not added

Table 1: Hyperparameters for Naive Bayes

For Logistic Regression in binary classification, the hyperparameters include various **solvers** like 'liblinear' with both L1 and L2 penalties and 'newton-cg', 'lbfgs' and 'sag' with L2 penalty, ensuring flexibility in handling different data complexities and regularization needs. The range of **C** values from 0.001 to 100 allows for fine-tuning between underfitting and overfitting, while the **class_weight** parameter includes the 'balanced' option to adjust weights inversely proportional to class frequencies, crucial for addressing the imbalance by ensuring minority class examples (non-cyberbullying) receive appropriate attention.

For the multiclass task, where the dataset is balanced, the hyperparameter space is streamlined with **solvers** like 'newton-cg', 'lbfgs', and 'sag', maintaining the same range for **C** and including **class_weight** to ensure robust performance across varied data complexities.

Model	**c**	**class_weight**	**penalty**	**solver**
Binary with BOW	1	None	L1	liblinear
Multiclass with BOW	0.1	None	L2	newton-cg
Binary with TF-IDF	10	None	L2	lbfgs
Multiclass with TF-IDF	1	Balanced	L2	lbfgs

Table 2: Hyperparameters for Logistic Regression

For SVM (Support Vector Machines), the hyperparameters are carefully selected to optimize performance across high-dimensional feature spaces typical in text classification tasks. The choice of **kernel** functions such as 'linear', 'sigmoid', 'poly', and 'rbf' provides flexibility in capturing different types of decision boundaries. The regularization parameter **C**, ranging from 0.1 to 100, controls the trade-off between maximizing the margin and minimizing classification errors. This range allows tuning for both underfitting and overfitting scenarios, crucial for handling complex datasets effectively. Additionally, parameters like **gamma**, which influences the kernel's influence range in 'rbf', 'poly', and 'sigmoid' kernels, are set to values like 0.0, 0.1, and 0.5 to explore different kernel coefficients.

For XGBoost, a highly regarded ensemble method, the hyperparameters are chosen to leverage its strengths in handling structured data and achieving robust performance:

- **n_estimators**: set to values like 100, 300, and 500, determines the number of boosting rounds, influencing model complexity and training time.
- **max_depth**: tuned to values such as 10, 20, and 30, controls the maximum depth of each tree in the boosting process, managing model complexity and potential overfitting.
- **learning_rate**: ranges like 0.05 and 0.25 determine the step size at each iteration while moving towards a minimum of the loss function, impacting both convergence speed and final model performance.
- **min_child_weight**: choices of 1 and 2 specify the minimum sum of instance weight (hessian) needed in a child, providing regularization to avoid overfitting.

- **gamma**: values of 0.0, 0.1, and 0.5 represent the minimum loss reduction required to make a further partition on a leaf node of the tree, controlling model complexity.

Model	**Best Hyperparameters**
SVM	**C**: 10 (binary) - 1 (multiclass), **kernel**: RBF, **max_tier**: 100000
XGBoost	**n_estimators**: 500, **max_depth**: 30, **learning_rate**: 0.05, **min_child_weight**: 1, **gamma**: 0.1

Table 3: Best Hyperparameters for SVM and XGBoost Models

4.4 Metrics for classification

Selecting appropriate evaluation metrics is crucial in evaluating the effectiveness of a classification model, especially in contexts such as cyberbullying detection, where nuanced insights into model performance are essential.

In our evaluation, we have chosen to employ a combination of metrics tailored to the nature of our tasks. For the multiclass classification task, where the goal is to categorize messages into multiple classes, **accuracy** emerges as a natural choice. Accuracy provides a comprehensive measure of the model's overall correctness in predicting all classes accurately. This choice is particularly suitable when each class carries equal importance and the dataset has been meticulously balanced to ensure fair representation across all categories. However, it's vital to acknowledge that accuracy may not fully capture model performance in scenarios with imbalanced class distributions. In such cases, additional metrics like recall, precision and F1-score become more reliable, offering deeper insights into how well the model identifies instances from minority classes.

For the binary task of cyberbullying detection, where messages are classified as either positive (cyberbullying) or negative (non-cyberbullying), choosing **recall** as the primary metric for the non-cyberbullying class is appropriate. This choice is justified by the clear interest in accurately identifying non-cyberbullying cases to avoid unnecessary censorship and ensure fair representation of all users. In other words, it's preferable to misclassify some true cyberbullying messages as non-cyberbullying rather than incorrectly flagging non-cyberbullying messages as cyberbullying.

To enhance the comprehensiveness of our evaluation, we have also incorporated the **F1-score** as a complementary metric. The F1-score provides a harmonic mean of precision and recall, offering a balanced assessment of the model's performance that considers both false positives and false negatives. This additional metric allows for a more nuanced understanding of the model's effectiveness in correctly identifying non-cyberbullying instances.

In this binary task, cyberbullying tweets are already largely recognized due to the dataset imbalance. Therefore, our goal is to maximize the recall for the non-cyberbullying class to minimize false positives, ensuring the model accurately identifies non-cyberbullying instances and reduces the risk of unfairly censoring benign content.

5 Results for classification

In this section, we present the comprehensive results of our classification experiments. For both the baseline and the advanced models (with the exception of LSTM), we employed a methodical 90% **development** and 10% **test** split, as outlined in 4.1. To optimize the performance of these models, we conducted a grid search with 5-fold cross-validation. This approach allowed us to systematically explore a range of hyperparameters and select the optimal configurations that maximized model performance. The 5-fold cross-validation, in particular, ensured that each model was trained and validated on multiple subsets of the data, thus enhancing the robustness and reliability of our results.

However, for the LSTM and Transformer models, our approach was slightly different. Given the complexity and computational intensity of these models, we defined their architectures without engaging in a grid search. Instead, we further subdivided the development set, allocating 80% to **training set** and retaining 20% as a **validation set**. After running the initial training set for a certain number of epochs, we proceeded with a subsequent phase that involved both the training and validation sets, using the number of epochs suggested by the initial training stage. This approach was employed to maximize the usage of available data, utilizing the development set to refine model performance. By doing this, we ensured that the models could learn from the entire dataset, thus improving their ability to generalize and perform well on the test set.

5.1 Baseline and advanced models

Following our approach introduced in section 4.4, we made the decision to focus on different metrics for both the binary and multiclass tasks. For the binary task, given its inherent imbalance and the near certainty of achieving high accuracy, we opted to prominently feature precision, recall, and f1-score in our tables. These metrics give a detailed view of how well the model finds positive examples while accounting for missed positives and false alarms, offering a comprehensive perspective beyond mere accuracy. Conversely, for the multiclass task, characterized by a balanced distribution across classes, we prioritized accuracy as the primary metric of evaluation.

Our baseline exhibit different strengths in identifying cyberbullying and not cyberbullying tweets. Firstly, in the binary task the Bernoulli Naive Bayes model with Bag-of-Words achieves 56% precision for non-cyberbullying (0) and 90% for cyberbullying (1), with a similar pattern in recall and F1-score. Meanwhile, the Multinomial Naive Bayes with TF-IDF scores higher in precision for non-cyberbullying (60%) but lower in recall, emphasizing its strengths and weaknesses in differentiating cyberbullying. Moving to logistic regression models, both with Bag-of-Words and TF-IDF, they maintain consistent precision and recall scores, showcasing their stability across different text representations.

Model	**Precision**		**Recall**		**F1-score**	
	0	**1**	**0**	**1**	**0**	**1**
Bernoulli with BOW	0.55	0.90	0.50	0.92	0.53	0.91
Multinomial NB with TF-IDF	0.59	0.88	0.38	0.95	0.46	0.92
Binary LR with BOW	0.59	0.91	0.54	0.92	0.56	0.92
Binary LR with TF-IDF	0.63	0.90	0.49	0.94	0.55	0.92

Table 4: Results of binary task for baseline models

Moving to to the multiclass task, we've seen strong performances. Bernoulli Naive Bayes with BOW and Multinomial Naive Bayes with TF-IDF achieve accuracies of 79% and 75%, respectively. Multiclass Logistic Regression excels with an 83% of accuracy in both BOW and TF-IDF setups.

Model	**Accuracy**
Bernoulli NB with BOW	0.79
Multinomial NB with TF-IDF	0.75
Multiclass LR with BOW	0.83
Multiclass LR with TF-IDF	0.82

Table 5: Results of multiclass task for baseline models

Turning to the results for advanced models in the binary task, we observe notable improvements in the metrics. The XGBoost model demonstrates a significant strength with a precision of 65% for non-cyberbullying (0) and 90% for cyberbullying (1), along with a recall of 48% for non-cyberbullying and an impressive 95% for cyberbullying. The resulting F1-scores are 55% and 92% for non-cyberbullying and cyberbullying, respectively. Similarly, the SVM model closely follows with precision scores of 63% for non-cyberbullying and 90% for cyberbullying, and recall scores of 49% and 94%. The F1-scores here are also robust at 55% and 92%. The Bidirectional LSTM model, while slightly lower, still performs well with precision at 60% for non-cyberbullying and 89% for cyberbullying, recall at 42% and 94%, and F1-scores at 49% and 92%. These results highlight the advanced models' ability to effectively identify cyberbullying tweets, particularly in maintaining high precision and recall for the positive class. When focusing on the binary task, it's notable that despite the advancements in models like XGBoost and SVM, Logistic Regression with Bag-of-Words stands out as the top performer for classifying non-cyberbullying (class 0) tweets.

In the multiclass task, the advanced models also exhibit strong performance. The XGBoost model achieves the highest accuracy at 85%, indicating its effectiveness in handling the complexity of multiclass classification. The

Model	Precision		Recall		F1-score	
	0	1	0	1	0	1
XGBoost	0.65	0.90	0.48	0.95	0.55	0.92
SVM	0.63	0.90	0.49	0.94	0.55	0.92
Bidirectional LSTM	0.60	0.89	0.42	0.94	0.49	0.92

Table 6: Results of binary task for advanced models

SVM and Bidirectional LSTM models both follow closely with accuracies of 83%, showcasing their robustness and consistency in performance.

Model	Accuracy
XGBoost	0.85
SVM	0.83
Bidirectional LSTM	0.76

Table 7: Results of multiclass task for advanced models

5.2 Comparison with SOTA

In our cyberbullying classification project, we employed XGBoost, the state-of-the-art machine learning technique designed by the author of the dataset. To ensure our methodology was aligned with the best practices and previous research, we followed the approach adopted by the authors. Specifically, we removed the ”not cyberbullying” class from the dataset. This decision was motivated by the need to concentrate our analysis on the more critical and nuanced classes that represent different types of cyberbullying.

In this way, our XGBoost model achieved an impressive 95% accuracy rate in the multiclass classification task. This high level of accuracy underscores the model’s capability to distinguish between different types of cyberbullying effectively. Such a performance indicates that our model has an higher accuracy than the authors models. It is necessary to consider that the dimension of the train and test data are different, since we tested our model using more training data. Moreover, our better performances might be due to the language filtering and the text cleaning that we have done in the preprocessing step.

5.3 Transformers

In out task we employed state-of-the-art language models based on the Transformer architecture, which has become the standard for building large-scale NLP models.

Transformers, introduced by Vaswani et al. (2017), revolutionized the field of NLP by introducing the self-attention mechanism. This mechanism allows the model to capture long-range dependencies and contextual relationships between words, enabling it to build rich representations of word meanings that integrate information from surrounding words.

In our experiments, we evaluated the performance of three prominent Transformer-based language models:

1. **BERT** (Bidirectional Encoder Representation from Trasfomers): BERT is a pre-trained language model that has achieved state-of-the-art results on a wide range of NLP tasks. It is trained on a vast corpus of text data using a novel pretraining objective, enabling it to capture bidirectional contextual representations;

2. **RoBERTA** (Robustly Optimized Bert Pretraining Approach): RoBERTa is an optimized variant of BERT. It incorporates several improvements to the pretraining process, including larger batch sizes, longer training sequences, and more diverse pretraining data, resulting in enhanced performance on various NLP tasks;

3. **Bert Multilingual Uncased**: this model is a multilingual version of BERT, trained on a large corpus of text data from multiple languages. It is particularly useful for tasks involving multilingual or cross-lingual data, as it can effectively handle text in different languages without the need for separate models.

For the dimension of the **batch size**, we opted for a value of 16 examples per batch, which allows reducing the amount of data allocated in memory and consequently saving resources. The learning rate and eps (to improve numerical stability) hyperparameters were chosen to be the most commonly used values for defining the learning process. As the optimizer, we used **Adam**, a popular choice for its adaptive learning rate and efficient convergence. The loss function used was **cross-entropy**, a standard choice for classification tasks. Considering the **epochs** hyperparameter, the number of epochs was selected after visualizing the learning and accuracy curves to avoid potential overfitting situations, where the model starts to memorize the training data instead of generalizing well to unseen examples.

Model	**Precision**		**Recall**		**F1-score**	
	0	**1**	**0**	**1**	**0**	**1**
Bert	0.68	0.91	0.54	0.95	0.61	0.93
Roberta	0.52	0.95	0.78	0.86	0.63	0.90
Multilingual Bert	0.70	0.89	0.42	0.96	0.52	0.93

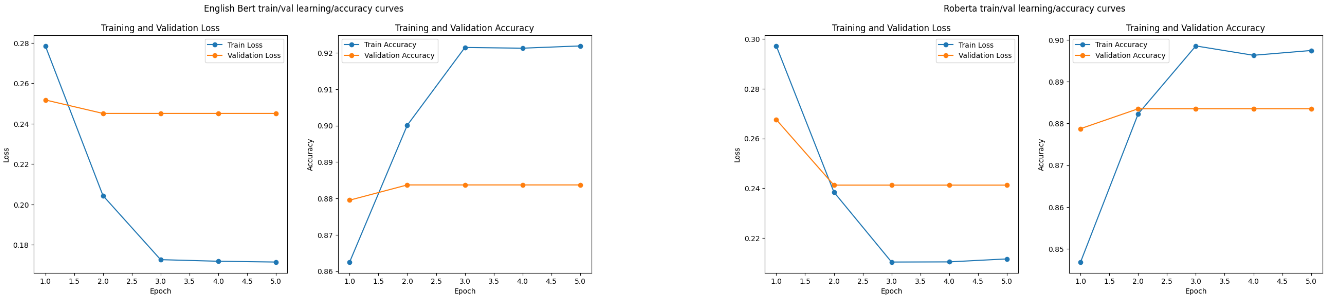
Table 8: Results of binary task for Transformers

In a binary classification context, the BERT model demonstrates the highest precision for class 0, accurately classifying the majority of non-cyberbullying tweets. However, it suffers from lower recall and F1 scores compared to RoBERTa and exhibits the false flagging issue, misclassifying almost 45% of non-cyberbullying tweets as harmful texts. The BERT multilingual uncased model also has a high precision value, but its low recall leaves a significant number of non-cyberbullying tweets incorrectly classified. On the other hand, RoBERTa tends to classify more tweets as non-cyberbullying due to its high recall value, making it less susceptible to the false flagging phenomenon, but it allows more harmful messages to be considered as non-cyberbullying.

Model	**Accuracy**
Bert	0.86
Roberta	0.86
Multilingual Bert	0.82

Table 9: Results of multiclass task for advanced models

In the multiclass task, BERT is among the models with the highest accuracy (86%) but shares the common weakness of confusing non-cyberbullying tweets with other types of cyberbullying tweets. Even with a multilingual dataset, this model makes similar errors as other transformers and maintains approximately the same accuracy. Similarly, RoBERTa is one of the models with the highest accuracy (86%) and it also shares the same weakness of confusing non-cyberbullying tweets with other cyberbullying tweets.



(a) Learning/accuracy curves Bert

(b) Learning/accuracy curves Roberta

Figure 3: Learning/accuracy curves for Transfomer models

Looking at the plot, we can see that the model slightly overfits in the second epoch. However, we still chose 2 as the definitive number of epochs. This decision is based on the observation that the validation loss plateaus and the validation accuracy stabilizes after the second epoch, suggesting that the model has reached a good balance between learning and generalization at this point.

Despite the slight overfitting observed, the overall performance on the validation set remains strong, with minimal difference between training and validation accuracy (about 2%). This indicates that the model maintains good generalization capabilities without significant degradation in performance on unseen data.

A possible alternative to mitigate overfitting even further would be to reduce the complexity of the model.

5.4 Ensemble

The motivation of the implementation of an ensemble model was to create a final model which is a combination of more models, where the first one has to be able to perform the binary classification task, and the following one has to distinguish the cyberbullying messages from the predictions of the precedent model. Since the multiclass task was well performed even by the baselines, we decided to focus oh the binary one, by trying an ensemble approach.

We implemented an ensemble approach combining BERT and RoBERTa models to optimize performance for classification tasks, since we understood the the multi class task is not as challenging. These models were chosen for their advanced capabilities in natural language understanding, as demonstrated by their performance in our specific task.

In our ensemble strategy, which is very powerful in classification tasks and usually used by Machine Learning competition winners, BERT and RoBERTa models contribute their predictions through a **arithmetic average**. This approach allows us to take advantage of the strengths of each model in making decisions. In instances where there is a tie in their predictions, we adopted a predefined strategy due to the imbalance in class distribution: assigning the label ”0” (not cyberbullying) to tweets. This decision ensures a consistent and fair classification approach while optimizing recall to non-cyberbullying instances.

Model	**Precision**		**Recall**		**F1-score**	
	0	**1**	**0**	**1**	**0**	**1**
BERT + RoBERTa	0.52	0.96	0.83	0.85	0.64	0.90

Table 10: Results of binary task for BERT + RoBERTa model

Looking at the table, we can see that the Ensemble is the model with the highest f1-score, and has high values of recall for both class 0 and 1, but has a low precision for class 0. However, this ensemble model is still not the best solution for our binary task.

6 Conclusions

During the development of this project, we analyzed the performances of several models for this cyberbullying classification task, putting into practise the theoretical and methodological concepts learned during the Human Language Technologies course.

Giving the conclusions, ML models applied to Natural language are very able to distinguish between the different type of cyberbullying acts, whereas is much more difficult for them to capture the intentions of the speaker (or the writer) when they must discriminate not cyberbullying tweets from harmful messages and one particular category of social group is cited in the text. Surely, the false flagging phenomena mentioned by Jurasky is still one of the most challenging and important topics of the NLP field, even for transformer models. Therefore, we believe that this will be a future test for the research in disambiguation of the context, comprehension of the intentions of the speakers and general help to people to prevent a dangerous social phenomena as the cyberbullying one.

References

[1] D. Jurafsky and J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, 2009.

[2] A. Lenci, S. Montemagni, and V. Pirrelli. *Testo e computer: elementi di linguistica computazionale*. Aula magna. Carocci, 2016.

[3] Jason Wang, Kaiqun Fu, and Chang-Tien Lu. Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1699–1708, 2020.