NLP 2022-2023 Homeworkp

Mina Makar
Data Science / 1804475
makar.1804475@studenti.uniroma1.it

Alex Onofri
Data Science / 1795982
onofri.1795982@studenti.uniroma1.it

Emanuele Vincitorio
Cybersecurity / 1811290
vincitorio.1811290@studenti.uniroma1.it

Abstract

In today's digital age, social media and online platforms have become an integral part of our daily lives. While these platforms offer a space for open dialogue and sharing, they also come with challenges, especially in managing the proliferation of hate speech. The anonymity provided by these platforms often encourages users to share aggressive and damaging content without fear of immediate consequences. This problem harms individual users and also worries society as a whole, leading experts and website owners to look for ways to solve it.

This report dives into the world of hate speech detection. We explore the cutting-edge methods, tools, and strategies currently being employed in this domain. A significant part of our focus will be on techniques grounded in natural language processing and deep learning—two areas that have shown promising results in understanding and categorizing textual content.

As we navigate through this complex landscape, we'll take a closer look at some of the modern methods employed by researchers and developers and others in the field. Where existing code or implementations are missing, we'll take the initiative to recreate and test them ourselves. To ensure our findings are robust and applicable, we'll evaluate these methods across various datasets, providing readers with a comprehensive view of the current state of hate speech detection and potential paths forward.

1 Task description/Problem statement

The NLP task addressed in this paper is Text Classification, with a specific focus on the topic of Hate speech detection.

Text classification algorithms, using machine learning techniques and linguistic analysis, can learn to recognize patterns and features within text data and through them automatically assign predefined categories or labels to text documents [29].

The issue of hate speech (HS) detection has gained significant prominence, especially with the rise of social media and the vast expanse of the internet. According to Paz, Montero-Díaz, and Moreno-Delgado (2020), the growing emphasis on hate speech research is closely tied to its increased visibility on digital platforms and the escalating media coverage surrounding this phenomenon [31].

One of the challenges in tackling hate speech lies in its definition. Different scholars offer various perspectives on what constitutes hate speech. Tsesis (2002) for instance, notes that some definitions of hate speech emphasize identifying characteristics like race, color, religion, ethnicity, or nationality [39].

However, it's essential to understand that hate speech is not just a matter of terminology or targeted characteristics. Brown (2017) elucidates that hate, in the context of hate speech, signifies an emotional state or opinion, which is distinct from any manifested action [8]. This distinction is vital because it highlights the underlying emotional triggers and sentiments that give rise to such expressions. To simplify, hate speech can be perceived as any emotional expression directed with animosity towards individuals.

Given the complexities in defining and understanding hate speech, there's a pressing need for tools and techniques that can effectively detect and manage it. Martins et al. (2018) emphasize the potential of automated techniques that can classify text as hate speech programmatically [25].

1.1 Examples

In this section, we illustrate examples of application of hate speech detection algorithms that take in input some text (e.g. tweets, comments, articles) and return back the predicted category between Hate Speech or Non-Hate Speech/Neutral.

The goal is to accurately differentiate between offensive or harmful language and neutral expressions.

Input text	Expected output
I can't believe they let those people attend. Disgusting!	Hate Speech
The event was a great success, thanks to everyone involved!	Non-Hate Speech
This group is responsible for all the problems. They should be eradicated	Hate Speech
The team worked collaboratively to overcome challenges	Non-Hate Speech

1.2 Real-world applications

Hate speech detection is becoming more important in text classification. With more people using online platforms to communicate, it's handy to have algorithms that automatically spot and flag harmful content. This helps to make online spaces safer and reduces the negative effects of hate speech. Beyond legal requirements, platforms like Facebook have been proactive in taking significant steps to limit online hate speech[3]. That's why we see hate speech detection being used a lot on social media, online communities, and places like content platforms.

2 Related work

Hate speech detection has become an increasingly crucial domain, especially considering the surge in online interactions on various platforms. As a response to this growing concern, there have been several advancements in utilizing sophisticated machine learning techniques for hate speech detection.

The landscape of hate speech detection is diverse, characterized by a myriad of models and feature embeddings tailored to the diversity of the task. Dinakar et al. (2012) utilized the Tf-idf feature embedding, a popular choice for its ability to capture the meaning of words in documents. This representation was further processed using the SVM algorithm, a robust classifier for text data [17]. Venturing into deeper architectures, Badjatiya et al. (2017) experimented with a suite of feature embeddings, including FastText, GloVe, Random Embedding, Tf-IDF, and BOW. These embeddings were paired with diverse algorithms, from traditional ones like Logistic Regression and SVM to neural network architectures such as CNN and LSTM, showcasing the breadth of approaches in this domain [2].

Mozafari, Farahbakhsh, and Crespi (2020) proposed a novel approach leveraging the capabilities of BERT (Bidirectional Encoder Representations from Transformers) for hate speech detection. They explored a transfer learning technique capitalizing on this pretrained language model, emphasizing its potential for the task [30].

In a study by Kim, Park, and Han (2022), the authors delve deeper into the challenges of implicit hate speech, which often lacks obvious lexical cues. While they acknowledge that fine-tuning on an implicit hate speech dataset yields satisfactory results on the same dataset, the performance drops notably when applied to a different dataset. To address this, they introduced a contrastive learning method named ImpCon. Their experiments demonstrated substantial improvements when using ImpCon with BERT and HateBERT on cross-dataset evaluations [23].

Caselli et al. (2021) introduced HateBERT, a variant of BERT retrained specifically for abusive language detection in English. This model was trained on a large-scale dataset of English Reddit comments sourced from communities known for offensive or abusive content. Their evaluations showed that HateBERT outperformed the general BERT model on various datasets for offensive language and hate speech detection tasks [10].

The potential of large language models, like ChatGPT, has also been explored for hate speech detection. Das, Pandey, and Mukherjee (2023) conducted a detailed evaluation of ChatGPT for detecting hate speech across 11 languages, emphasizing its multilingual capabilities [13]. Another study by Huang, Kwak, and An (2023) examined the feasibility of using ChatGPT to provide natural language explanations for implicit hateful speech detection, underscoring the model's potential in delivering meaningful insights into the detection process [21]. Additionally, Sood and Dandapat (2023) innovatively proposed a custom GPT-4 few-shot prompt annotation scheme, focusing on the identification of problematic webpages that promote hate and violence [38].

For a comprehensive understanding of this domain, it's essential to refer to extensive reviews in the literature. Schmidt and Wiegand (2017) offer a detailed survey on hate speech detection, emphasizing the challenges and the need for automated methods [37]. Similarly, Fortuna and Nunes (2018) provided an overview of automatic detection techniques for hate speech in text, shedding light on the various methods employed [18]. A more recent systematic review by Jahan and Oussalah (2023) focused on the last decade's advancements, specifically highlighting the role of natural language processing and deep learning technologies in hate speech detection [22].

A significant issue highlighted in many reviews is that numerous languages, particularly dialects and low-resource languages, struggle due to insufficient annotated data for training deep neural networks. This leads to inconsistent hate speech moderation and protection across different languages. [35, 22, 18, 37].

In hate speech detection, English has been at the forefront due to the abundance of data. But recently, researchers have started to adapt this English data in innovative ways. They're employing techniques like cross-lingual embeddings and transfer learning to address languages with less data, such as Bengali, Hindi, and Spanish. [34].

Researchers have explored cross-lingual transfer learning as an alternative to collecting new hate speech data. By leveraging existing data from higher-resource languages, they aim to enhance hate speech detection in languages with limited resources [5].

Additionally, researchers have developed language-specific BERT models to address specific linguistic challenges. For instance, "RoBERT" is a Romanian BERT model tailored to Romanian language tasks [26] , and "Spanish-BERT" is designed for the Spanish language [9].

Despite linguistic diversity, multilingual BERT models like "mBERT," trained on a multitude of languages, have exhibited impressive cross-lingual performance across various natural language processing tasks, bridging the gap in hate speech detection and other NLP applications [42]. These models contribute to more equitable language processing solutions.

3 Datasets and benchmarks

In this section, we'll be diving into the various datasets and benchmarks specifically designed for hate speech detection. These tools provide the foundation for understanding and addressing hate speech in digital spaces.

3.1 Datasets

From the datasets point of view, a salient observation is the dominance of English language datasets in the topic of hate speech detection and the variety of platforms and different way to classify as hate speech.

The "Hate Speech Dataset" from 2018, sourced from Stormfront, consists of 9,916 entries and has a ratio of 0.11. This indicates that about 11% of the entries in this dataset are categorized as hate speech. The data is classified into 'Hate', and 'Not Hate' categories [15]. The "Hate Speech Twitter annotations" from 2016 is derived from Twitter. It is composed of 4,033 annotations, with a focus on Racism and Sexism, having a ratio of 0.16 [40]. In 2017, a dataset titled "Hate Speech Detection Fox news comments" was curated from Fox News comments, containing 1,528 instances. This dataset adopts a binary classification system: Hate or not, with a notable ratio of 0.28 [19]. The "CONAN Multilingual Dataset of Hate Speech," although multilingual, offers data from 2019 focused on Islamophobia, with 1,288 entries sourced from Synthetic and Facebook platforms [11]. Another dataset from Reddit, titled "Online Hate Speech," from 2019, boasts a substantial 22,324 entries, categorizing data into Hate and Not hate with a ratio of 0.24 [33]. Finally, the "Personal Attacks" dataset from 2017, sourced from Wikipedia, stands out with an impressive 115,737 entries, dedicated to distinguishing Personal attacks from non-attacks, with a ratio of 0.12 [43].

However, in the last few years some steps are being made to diversify this dataset landscape with the inclusion of other languages. For instance, the "Religious Hate Speech in the Arabic" dataset, sourced from Twitter in 2018, encapsulates 16,914 entries, focusing on Hate and Not hate categorization, specifically in Arabic [1]. The "GermEval 2018" dataset, curated in 2016, contains 8,541 entries from Twitter, emphasizing various categories such as Offense, Other, Abuse, Insult, and Profanity in the German language [41]. Lastly, ”CONAN HS French,” a dataset from 2019, with 17,119 entries, concentrating on distinguishing between Islamophobic and non-Islamophobic content, all in French [11].

Recent observations show a change in language research. Although many studies focus on English, there’s a growing interest in including more languages. Since we all connect globally on social media, this variety is important. If we want hate speech detection tools to work well everywhere, we need to look at different languages. This makes our studies more complete and helps make the online world safer for everyone.

3.2 Benchmark

In the evolving landscape of Natural Language Processing (NLP), various benchmarks have emerged to tackle this subject trying to increase the understanding of this topic. Qian et al. (2019) introduced an innovative approach to intervene in online hate speech, moving beyond just detecting it. Their focus was on generating automatic responses to conversations that contain hate speech, using datasets from Gab and Reddit as their primary source [33]. Meanwhile, Salminen and his team (2020) worked on a classifier that detects online hate across multiple platforms like YouTube, Reddit, Wikipedia, and Twitter. They found that their model, based on XGBoost, was effective across these diverse platforms [36]. Mathew et al. (2021) launched HateXplain, a dataset uniquely focusing on explainable hate speech detection. This dataset offers insights into the nature of posts, their targeted communities, and the reasons for their specific classifications [27]. Interestingly, Kulkarni et al. (2023) presented GOTHate, a vast dataset from Twitter, emphasizing its ”neutrally seeded” nature covering various languages and topics.[24]

4 Existing tools, libraries, papers with code

Hate speech detection has really grown thanks to new tech, tools, and practical solutions found in research papers. This section delves into these existing resources, providing an overview of the existing tools, libraries, frameworks and papers with code. The goal is to give a complete look at the tools and methods available to researchers.

4.1 Existing tools and libraries

In the rapidly evolving of Natural Language Processing (NLP), various tools and libraries have come, addressing diverse linguistic and computational challenges. Kili Technology offers a straightforward way to annotate and transform data, optimizing the creation of machine-learning datasets. It blends smoothly with common machine learning tools, ensuring a user-friendly experience. When it comes to comprehensive frameworks for language analysis, the Natural Language ToolKit (NLTK) is an essential tool for Python developers to manage and analyze human language data. In the same way, Stanford’s CoreNLP offers a multi-lingual, scalable option written in Java. Apache OpenNLP and SpaCy both emphasize user accessibility, with the latter being particularly known for its advanced tokenization capabilities. The big tech companies also offer formidable NLP solutions. Both Google Cloud’s Natural Language API and Amazon Comprehend tap into cloud capabilities, offering a wide array of pre-trained models and text analysis options. These platforms assist businesses in deriving meaningful insights from their extensive text data. For those focused on research and advanced text preprocessing, AllenNLP is a preferred choice, while GenSim provides robust capabilities for handling large linguistic datasets. And for a touch of innovation, OpenAI’s GPT-3/4 can be used for providing natural language explanations (NLEs) for implicit hateful speech detection. Other tools like Text Blob and CogCompNLP round out the selection, offering speedy machine learning solutions and a versatile approach to text data processing respectively. Finally for building and training hate speech detection models, libraries like Scikit-learn, TensorFlow, and PyTorch offer versatile frameworks. Additionally, pre-trained models such as BERT have been fine-tuned for hate speech detection tasks, providing efficient and effective solutions, and can be found from hubs like Hugging face, Tensorflow hub, Pytorch Hub, John Snow LABS(written in spark). In essence, the NLP space has a diverse set of tools, each catering to specific needs and functions, ensuring there's something for every task.

4.2 Paper with code

In the evolving landscape of research, many papers not only present findings but also share the source code to provide transparency and make community collaboration easier. This practice aids in reproducibility and validation of the proposed methodologies. Some of the most famous and implemented papers with code are the following: HateXplain[27], Automated Hate Speech Detection [14] and A BERT-based transfer learning approach [30]

5 State-of-the-art evaluation

Whenever we train a machine or deep learning model to detect hate speech in text, it's crucial to assess how well it's performing. This process is known as evaluation. In the world of text classification, there's a suite of metrics available to ensure our model is hitting the mark.

1. **Accuracy** [32]: Quite simply, this metric tells us the proportion of correctly classified instances out of the total instances. It's the most straightforward metric but can be misleading if the dataset is unbalanced.

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions}$$

2. **Precision** [32]: Describe how many of the texts our model labeled as 'hate speech' were actually hate speech. A high precision indicates that when our model flags something as hate speech, it's likely right.

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall** [32]: Measures the proportion of actual positive cases that the model correctly identifies. A high recall means our model captures most of the hate speech out there.

$$Recall = \frac{TP}{TP + FN}$$

4. **F1 Score** [32]: Sometimes, we need a balance between precision and recall, especially if one of them is much higher than the other. F1 score is the harmonic mean of precision and recall and gives a combined score.

$$F1Score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

5. **Matthews Correlation Coefficient (MCC)**[6]: This is a correlation coefficient between observed and predicted classifications. The MCC returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction, and -1 indicates an inverse prediction.

6. **Receiver Operating Characteristics (ROC) [20]**: It's a graphical plot that illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. Essentially, ROC gives us a graph of true positive rate versus false positive rate.

7. **Area Under the ROC Curve (AUC)** [7]: AUC quantifies the overall ability of the model to discriminate between positive and negative classes. If you have an AUC of 1.0, you're in the golden zone – your model has perfect classification skills. An AUC of 0.5 suggests no discrimination, i.e., as good as random guessing.

6 Comparative evaluation

In this section, we will present our dataset, which has been used to assess the performance of our model. We will begin by introducing the datasets we've selected and detailing the preprocessing steps for each one. Following that, we present the specific models we've developed and elaborate on the metrics chosen to assess their effectiveness. In conclusion, we conduct a detailed examination of our results, measuring the performance of various models and datasets, and comparing the Model re-implementation with the original one presented in Mozafari [30].

6.1 Datasets

The following are the datasets used in our comparison, each one with the description and relative pre-processing performed on it.

6.1.1 SemEval-2019

One of the datasets utilized in our study is presented in [4], which we found to be of significant interest. This dataset comprises data collected from Twitter, specifically English and Spanish tweets, spanning from July to September 2018, with a particular emphasis on content targeting women.

To amass this dataset, the authors employed various strategies:

- monitoring potential victims of hate accounts
- downloading the history of identified haters
- filtering Twitter streams with keywords

The comprehensive HatEval dataset encompasses 19,600 tweets: 13,000 in English and 6,600 in Spanish. For the purposes of our study, we exclusively utilized the English portion of the dataset.

For annotation, the authors solicited a minimum of three independent judgments per tweet from untrained crowdsourcing contributors. Additionally, they obtained annotations from two domain experts. The final label for each tweet was determined based on the majority votes from the crowd, Expert 1, and Expert 2.

Finally the dataset contains 3 different labels:

- HS (Hate Speech): a binary value indicating the uses of Hate Speech in the tweet. 1 if occurs, 0 if not
- TR (Target Range): if HS is 1, we have that the target range is 0 if the target is a generic group of people, 1 if it's a specific individual
- A (Aggressiveness): if HS occurs, a binary value will indicate if the tweet is aggressive with 1, 0 if not

Preprocessing

Preprocessing the dataset is a crucial step for several reasons: Noise reduction, Standardization, Tokenization, Stopwords Removal, Feature Extraction, Text Cleaning. In this way we can clean, structure, and prepare text data for analysis and model training. The preprocessing steps are specific to each dataset. This is why the dataset must be dissected and understood so as to be able to carry out the steps described above in the best possible way and also avoid taking steps that would be redundant.

In this case [4], the dataset is very specific to the usage of the social network Twitter, since the dataset is composed by hate tweets. This is why we have to take in consideration that most of the text data may contains informations such as:

- Hashtags
- Users tag
- External links
- Emojis
- Numbers

Since these patterns are pretty standard for tweets, we kept some labels to point out different information like:

- `<hashtag>` for hashtags
- `<user>` for users tag
- `<url>` for external links
- `<emoticon>` for emojis
- `<number>` for numbers

On top of it we also have to take in consideration the cleaning of HTML tags, special characters, or formatting issues. Cleaning these elements ensures that the text is readable and consistent.

6.1.2 Automated Hate Speech Detection and the Problem of Offensive Language

We utilized another dataset in our research, as detailed in [14]. This dataset contains a random selection of 25,000 tweets from an extensive collection of 85.4 million tweets. The categorization was performed manually by crowdsourcers, who were tasked with classifying each tweet into one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech. Rather than focusing solely on the words in a tweet, the crowdsourcers were advised to consider the context of its usage. They were also informed that the mere presence of an offensive word did not automatically categorize the tweet as hate speech. Each tweet received evaluations from at least three individuals, with the majority opinion determining its label.

The dataset contains the following columns:

- count: indicates the number of crowdworkers who labeled the tweet
- hate_speech: denotes the number of crowdworkers who classified the tweet as hate speech
- offensive_language: represents the number of crowdworkers who considered the tweet as offensive
- neither: counts the crowdworkers who found the tweet neither hateful nor offensive
- class: a derived column with values 0 for hate speech, 1 for offensive language, and 2 for neither category
- tweet: the content of the tweet

Of the dataset, 6% is categorized as hate speech, 77% as offensive language, and the remaining 17% as non-offensive.

Text Processing Decisions

In this section, we delineate the procedures implemented for text processing.

1. Emojis were removed and replaced with text representing emotions. This decision was made because, unlike TF-IDF, LLMs (Large Language Models) can capture the context effectively. Removing emojis might alter or slightly change this context.

2. Any links were replaced with the tag `<url>`.

3. Usernames, indicated by `@user`, were replaced with `<user>`.

4. Numbers were substituted with the placeholder `<number>`.

5. Punctuation marks were removed.

6. Stopwords were retained as they play a crucial role in understanding the context of the text.

7. Any hashtags, represented as `#something`, were replaced with `<hashtag>`.

8. Leading and trailing white spaces in strings were removed.

9. Characters that are not part of the English alphabet were eliminated.

To facilitate this, the 'emoji' library was employed for emoji management, and the 're' library was utilized to manage regular expressions.

6.1.3 Toxic Comment Classification Challenge

Another dataset that we found interesting for our study is the Wiki_Toxic dataset that contains around 200 thousands of comments from Wikipedia forums which have been labeled by humans identifying presence or absence of toxicity. This is a reduced version of the following dataset [12] that has been created for the Toxic Comment Classification Challenge on Kaggle. It has been cleaned by crowd-source users and reduced. The cleaning consisted on removing and replacing some defined regex patterns by text comments. Also, in the original dataset there were six different types of toxicity labeled singularly with 0 or 1, while in this cleaned version these columns were unified into a binary classification. The value of this column is set to 1 when at least one of these types were labeled as 1:

- toxic
- severe_toxic
- obscene
- threat
- insult
- identity_hate

Additionally, the dataset was retrieved from Hugging Face where there are already three splits available: Train, Test and Validation subset. We took them all and then selected all the comments labeled as toxic (hate speech) and the same number has been selected randomly from all the available non-toxic comments. In this way we balanced the dataset since there was just a small percentage of toxic comments.

The final dataset containes around 45 thousands rows with the following 3 columns:

- **Id**: comment id
- **comment_text**: text of the comment
- **label**: label set to 1 if the text is classified as toxic, else 0.

Text Pre-processing

Before using this dataset we applied some pre-processing operation to make easier and more efficient the prediction with the various models tested. The pre-processing performed was done through NLTK library and consists in:

- Removing links, numbers and punctuation
- Tokenization
- Lower casing
- Removing stopwords
- Lemmatization

An important specification to do is that stopwords weren't removed in all the cases tested. In fact, while it's important to remove them in word embedding algorithms like Tf-Idf, in features embedding like BERT they have an important role in giving the context and improving comprehension of the sentence.

6.2 Models

In this section, we will discuss the chosen embeddings for each model. Embeddings play a crucial role in enhancing the performance of models, and our selection has been made with careful consideration to ensure optimal results.

6.2.1 HateBERT + MLP

BERT (Bidirectional Encoder Representations from Transformers) [16] is a popular and powerful pre-trained natural language processing (NLP) model developed by Google. Researchers and organizations often fine-tune BERT for specific NLP tasks, such as sentiment analysis, text classification, and question answering. HateBERT is a specialized version of BERT designed for hate speech detection or related tasks, it is a fine-tuned variant that has been trained on a specific dataset called RAL-E [10].

BERT-based Feature Extraction: We use HateBERT, a pre-trained BERT model, fine-tuned for binary classification, as our base architecture. Input text sequences are tokenized and processed through the BERT model to obtain contextualized embeddings.

Multi-Layer Perceptron (MLP): The extracted BERT embeddings are fed into an MLP architecture for classification. The MLP consists of three fully connected layers: The first layer (fc1) with input size matching the BERT output that is 768 and a configurable hidden size that is 128, followed by a ReLU activation function; the second layer (fc2) with 2 output classes (to determine if it is HS or not). The final layer applies a LogSoftmax activation to produce class probabilities.

The HateBERT model is trained on labeled datasets for hate speech detection tasks. The training process involves optimizing the model's parameters to minimize a suitable loss function, typically the cross-entropy loss. In our case we decided to use the Negative Log Likelihood Loss.

6.2.2 BERT + CNN [30]

We started with a codebase from a 2019 paper. Given the rapid advancements in software, the first task was to update the code to match the current versions of various libraries. During this process, we realized that the original authors had a monolithic coding style. We decided to introduce more functions, making the code modular and easier to understand.

The paper's approach combines BERT base with a Convolutional Neural Network (CNN). BERT [16] is a powerful model known for its proficiency in understanding language nuances. It's been trained on vast datasets, including the English Wikipedia. There are two main versions: BERT$_{base}$, which has 12 layers, and BERT$_{large}$ with 24 layers.

We maintained the structural integrity of the original code, we made it cleaner and ensured it aligned with the initial architectural blueprint. At the begining we tested BERT$_{base}$ and we obtained more or less the same results as the original paper, then we tried to use BERT$_{large}$ but it didn't outperform BERT$_{base}$ and was more time-consuming.

The integration of CNN is crucial here. It excels in extracting the contextual features from the last layer of BERT, ensuring that the most relevant patterns are highlighted. Following this, a Multilayer Perceptron (MLP) processes these patterns to produce the final prediction.

6.2.3 Concatenated features embedding + XGBoost

This system is an original combination we have done reading different surveys and papers and trying putting these components together searching for interesting performances. In the following we will explain how this system is made, splitting it in two big parts: Features embedding algorithm and Classification model.

Regarding the former, we decided to use a combination of 3 different features/word embedding algorithms:

- **Tf-Idf with PCA**: The Tf-Idf was performed using the sklearn library setting a maximum number of features to 20'000 to cut down the computation needed. After that, to reduce the dimensionality we applied Principal Component Analysis method ending with an embedding of 100 values for each comment.

- **BERT**: a Pre-trained bert-base-cased model taken from HuggingFace Was used and we set the maximum sequence length to 32 and added special tokens

- **FastText**: Pre-trained model in English taken from the fasttext library that was trained on millions of comments taken from Common Crawl and Wikipedia. [28]

The previous algorithms where used to produce the respective embedding for each comment and then they were concatenated all together, having at the end a single embedding for comment.

After computing the embeddings we used XGBoost as classification model since we found that it's a prominent machine learning model to use in the topic of Hate speech detection. We performed hyperparameters optimization using the GridSearch method trying to find the best combination of parameters to use in the model. The parameters tested in the grid were: *gamma, learning_rate, subsample, max_depth, colsample_bytree, min_child_weight, reg_alpha, n_estimators, reg_lambda, max_depth, learning_rate* but tested separately and with just few combinations due to high computational time needed. After some tests we didn't find any combination that improves the performances so at the end we used the model with the default values set in the xgboost Python library.

6.3 Measures and the evaluation protocol

Evaluation is conducted using the following metrics takes from the ones described in Section 5: accuracy, precision, recall and F1-score. All these metrics are computed using the sklearn.metrics library. For the evaluating process, we split the datasets into 80% for Train, 10% for Validation and the remaining 10% for the Test, using the same seed to facilitate the replication. In this way we are sure to evaluate models on the same bunch of comments.

6.4 Results

Table 1: Accuracy

Model	**SemEval-2019**	**HHS**	**WikiToxic**
HateBERT	**0.76**	**0.91**	**0.92**
BERT + CNN	0.75	**0.91**	0.91
Concatenated + XGBoost	0.73	0.89	0.88

Table 2: Precision

Model	**SemEval-2019**	**HHS**	**WikiToxic**
HateBERT	**0.76**	**0.91**	**0.92**
BERT + CNN	0.75	0.90	0.91
Concatenated + XGBoost	0.72	0.87	0.88

Table 3: f1-score

Model	**SemEval-2019**	**HHS**	**WikiToxic**
HateBERT	**0.76**	**0.91**	**0.92**
BERT + CNN	0.75	0.90	0.91
Concatenated + XGBoost	0.72	0.88	0.88

Table 4: Recall

Model	**SemEval-2019**	**HHS**	**WikiToxic**
HateBERT	**0.76**	**0.91**	**0.92**
BERT + CNN	0.75	**0.91**	0.91
Concatenated + XGBoost	0.72	0.88	0.88

Across all metrics (accuracy, precision, recall, and F1-score) and datasets, the HateBERT model consistently performs the best. This is not surprising since it has been fine-tuned and optimized specifically for hate speech detection.

BERT + CNN is very close in performance to HateBERT, with small differences depending on the dataset and metric. This combination seems to be effective, this is due to the peculiarity of CNNs to be good at picking up on spatial patterns in data and so when combined with the rich embeddings from BERT, the model might be recognizing specific patterns or structures in the text that are indicative of hate speech.

The model which concatenates Fast text, BERT, and TF-IDF embeddings before feeding into XGBoost, performs slightly worse than the other two models across all metrics and datasets, although the difference isn't drastically lower. This combination tries to leverage the strengths of traditional embeddings (TF-IDF), modern word embeddings (Fast text), and contextual embeddings (BERT), with the goal of creating a rich, composite representation of text. However, the results show that it's not outperforming pure deep learning methods like HateBERT or BERT + CNN. One reason could be the challenge of merging different embeddings effectively; another could be the nature of XGBoost, which might not as powerful in navigating the high-dimensional space as neural architectures.

The performances underline the power of deep learning architectures, especially when fine-tuned, in processing textual data. While XGBoost is a powerful algorithm known in classification tasks, in the domain of hate speech where context and semantics are crucial, deep learning models like BERT tend to outperform. In fact, BERT's embeddings has the strength of capturing contextual relationships between words, making it particularly efficient in understanding texts. This might be why models using BERT as a base consistently outperform the concatenated embeddings model.

Summing up, while concatenating different embeddings can be a good idea to create a complete representation of the text, in the hate speech detection task, deep learning architectures seems to be more effective.

6.4.1 Comparison with Papers

For a deeper understanding and validation of used methodologies, we have replicated the code from two papers: one detailing the workings of HateBERT and the other focusing on BERT + CNN. This section aims to give a comparative analysis of our replicated results against the findings reported in the respective papers. Our goal is not just to see if we can match their results, but also to spot any differences or unique aspects from our implementation.

Table 5: BERT + CNN with Davidson Dataset

Models	**Precision**	**Recall**	**F1-score**
Our	0.90	0.91	0.90
Paper [30]	0.92	0.92	0.92

In the presented table, we compare the results obtained from our replication of the BERT + CNN model with those reported in the paper by Mozafari et al. (2020) [30]. It's evident that while we closely replicated their results, slight discrepancies exist between the two sets of values. Our model achieved precision, recall, and F1-score values of 0.90, 0.91, and 0.90, respectively. In contrast, the original model showcased slightly superior scores of 0.92 across all three metrics. These differences could be attributed to various factors not entirely outlined in the original paper, such as data preprocessing or specific training details. Nevertheless, it's commendable that we managed to achieve results very close to the original, showcasing the rigor of our replication approach.

6.5 Discussion

HateBERT is effective in detecting hate speech, but can encounter challenges with new hate speech types.

The BERT + CNN model, blending BERT's depth with CNN's pattern identification, is constrained by a 512-token limit, which can exclude vital context for longer texts. Its focus on short word sequences might also cause it to overlook broader text relationships, leading to potential misinterpretations.

The Concatenated + XGBoost model combines various embeddings but can overshadow key features from BERT. XGBoost's lack of sequence understanding could lead to errors, especially in complex hate speech contexts.

In short, while each model has its advantages, they also have their own set of challenges that need detailed error checking.

7 Conclusions

In conclusion, our research on Hate Speech detection seeks to illustrate the current state-of-the-art regarding models, techniques, and benchmarks. It underscores how such systems can be integrated across various domain. To achieve optimal performance across diverse scenarios, our study explored multiple English datasets. Furthermore, we have conducted an in-depth analysis of the selected datasets, aiming to perform a preporcessing step that could yield the best results.

State-of-the-art evaluation methods were employed, ensuring that the models' performance was rigorously assessed. In our comparative evaluations, we considered datasets like SemEval-2019 and explored models such as HateBERT + MLP and BERT + CNN. The text processing decisions, such as the removal of emojis and their replacement with textual representations, were crucial to improving model performance.

Our results indicate the effectiveness of models like HateBERT in detecting hate speech, but they also underscore the challenges these models face, especially with newer forms of hate speech. The BERT + CNN model, in particular, showcased the potential of combining deep learning architectures for enhanced performance.

Future research should prioritize refining these models, perhaps with a focus on addressing the challenges highlighted in our discussion. Furthermore, the incorporation of diverse datasets, both in terms of language and content, will be paramount.

Contributions. For almost all the tasks, we worked together, but each of us at the beginning chose one feature embedding - classification model combination and one dataset to test it on.

Table 6: contribution

Name	**Model Name**	**Dataset**
Emanuele	HateBERT	SemEval-2019
Alex	Concatenated + XGBoost	wiki toxic
Mina	BERT + CNN	Automated HS

References

[1] Nuha Albadi, Maram Kurdi, and Shivakant Mishra. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76. IEEE, 2018. 4

[2] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760, 2017. 3

[3] Chara Bakalis. Rethinking cyberhate laws. *Information & Communications Technology Law*, 27(1):86–110, 2018. 2

[4] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. 7, 8

[5] Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. Cross-lingual transfer learning for hate speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 15–25, Kyiv, April 2021. Association for Computational Linguistics. 4

[6] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6):e0177678, 2017. 6

[7] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997. 7

[8] Alexander Brown. What is hate speech? part 1: The myth of hate. *Law and Philosophy*, 36:419–468, 2017. 2

[9] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang,

and Jorge Pérez. Spanish pre-trained bert model and evaluation data. *arXiv preprint arXiv:2308.02976*, 2023. 4

[10] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online, August 2021. Association for Computational Linguistics. 3, 10

[11] Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. CONAN - COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy, July 2019. Association for Computational Linguistics. 4, 5

[12] Julia Elliott Lucas Dixon Mark McDonald nithum Will Cukierski cjadams, Jeffrey Sorensen. Toxic comment classification challenge, 2017. 9

[13] Mithun Das, Saurabh Kumar Pandey, and Animesh Mukherjee. Evaluating chatgpt's performance for multilingual and emoji-based hate speech detection. *arXiv preprint arXiv:2305.13276*, 2023. 3

[14] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515, 2017. 6, 8

[15] Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*, 2018. 4

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 10

[17] Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Trans. Interact. Intell. Syst.*, 2(3), sep 2012. 3

[18] Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018. 3, 4

[19] Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria, September 2017. INCOMA Ltd. 4

[20] John A Hanley et al. Receiver operating characteristic (roc) methodology: the state of the art. *Crit Rev Diagn Imaging*, 29(3):307–335, 1989. 7

[21] Fan Huang, Haewoon Kwak, and Jisun An. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *arXiv preprint arXiv:2302.07736*, 2023. 3

[22] Md Saroar Jahan and Mourad Oussalah. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232, 2023. 3, 4

[23] Youngwook Kim, Shinwoo Park, and Yo-Sub Han. Generalizable implicit hate speech detection using contrastive

learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6667–6679, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. 3

[24] Atharva Kulkarni, Sarah Masud, Vikram Goyal, and Tanmoy Chakraborty. Revisiting hate speech benchmarks: From data curation to system deployment. *arXiv preprint arXiv:2306.01105*, 2023. 5

[25] Ricardo Martins, Marco Gomes, José João Almeida, Paulo Novais, and Pedro Henriques. Hate speech classification in social media using emotional analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 61–66, 2018. 2

[26] Mihai Masala, Stefan Ruseti, and Mihai Dascalu. RoBERT – a Romanian BERT model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6626–6637, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. 4

[27] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14867–14875, 2021. 5, 6

[28] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. 11

[29] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021. 1

[30] Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. A bert-based transfer learning approach for hate speech detection in online social media. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pages 928–940. Springer, 2020. 3, 6, 7, 10, 13

[31] María Antonia Paz, Julio Montero-Díaz, and Alicia Moreno-Delgado. Hate speech: A systematized review. *Sage Open*, 10(4):2158244020973022, 2020. 2

[32] David Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011. 6

[33] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764, Hong Kong, China, November 2019. Association for Computational Linguistics. 4, 5

[34] Tharindu Ranasinghe and Marcos Zampieri. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5838–5844, Online, November 2020. Association for Computational Linguistics. 4

[35] María Luisa Ripoll, Fadi Hassan, Joseph Attieh, Guillen Collell, and Abdessalam Bouchekif. Multi-lingual contextual hate speech detection using transformer-based ensembles. In *Forum for Information Retrieval Evaluation (Working Notes)(FIRE). CEUR-WS. org*, 2022. 4

[36] Joni Salminen, Maximilian Hopf, Shammur A. Chowdhury, Soon-gyo Jung, Hind Almerekhi, and Bernard J. Jansen. Developing an online hate classifier for multiple social media platforms. *Hum.-Centric Comput. Inf. Sci.*, 10(1), jan 2020. 5

[37] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain, April 2017. Association for Computational Linguistics. 3, 4

[38] Ojasvin Sood and Sandipan Dandapat. Problematic webpage identification: A trilogy of hatespeech, search engines and GPT. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 126–137, Toronto, Canada, July 2023. Association for Computational Linguistics. 3

[39] Alexander Tsesis. *Destructive messages: How hate speech paves the way for harmful social movements*, volume 27. NYU Press, 2002. 2

[40] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics. 4

[41] Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. Overview of the germeval 2018 shared task on the identification of offensive language. 2018. 5

[42] Shijie Wu and Mark Dredze. Are all languages created equal in multilingual bert? *arXiv preprint arXiv:2005.09093*, 2020. 4

[43] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. 4