TÉCNICO LISBOA

Automatic Hate Speech Detection in Portuguese Social Media Text

Bernardo Cunha Matos

Thesis to obtain the Master of Science Degree in

Engenharia Informática e de Computadores

Supervisors: Prof. Paula Cristina Quaresma da Fonseca Carvalho
Prof. Ricardo Daniel Santos Faro Marques Ribeiro

Examination Committee

Chairperson: Prof. José Alberto Rodrigues Pereira Sardinha
Supervisor: Prof. Paula Cristina Quaresma da Fonseca Carvalho
Member of the Committee: Profª. Maria Luísa Torres Ribeiro Marques da Silva Coheur

October 2022

Acknowledgments

Primeiramente, gostaria de expressar o meu agradecimento à Fundação para a Ciência e a Tecnologia, que financiou o projeto HATE COVID-19.pt, e também a toda a equipa do projeto que foi crucial para a realização deste trabalho. Em particular, agradeço profundamente ao Ricardo Ribeiro, à Paula Carvalho e ao Fernando Batista pela sua importantíssima contribuição.

Em segundo lugar, agradeço também à minha família, que me motivou sempre ao longo destes anos e cujo apoio foi fundamental para que pudesse alcançar esta fase académica.

Por último, dirijo um agradecimento especial a todos os meus amigos, pela importância clara que tiveram ao longo deste meu percurso académico.

Abstract

Online Hate Speech (HS) has been growing dramatically on social media and its uncontrolled spread has motivated researchers to develop a diversity of methods for its automated detection. However, the detection of online HS in Portuguese still merits further research. To fill this gap, we explored different models that proved to be successful in the literature to address this task. In particular, we have explored models that use the BERT architecture. Beyond testing single-task models we also explored multitask models that use the information on other related categories to learn HS. To better capture the semantics of this type of texts, we developed HateBERTimbau, a retrained version of BERTimbau more directed to social media language including potential HS targeting African descent, Roma, and LGBTQI+ communities. The performed experiments were based on CO-HATE and FIGHT, corpora of social media messages posted by the Portuguese online community that were labelled regarding the presence of HS among other categories. The results achieved show the importance of considering the annotator's agreement on the data used to develop HS detection models. Comparing different subsets of data used for the training of the models it was shown that, in general, a higher agreement on the data leads to better results. HATEBERTimbau consistently outperformed BERTimbau on both datasets confirming that further pre-training of BERTimbau was a successful strategy to obtain a language model more suitable for online HS detection in Portuguese. The implementation of target-specific models, and multitask learning have shown potential in obtaining better results.

Keywords

Text Classification; Hate Speech Detection; Supervised Learning; Transfer Learning; Deep Learning; Multi-task learning.

Resumo

O discurso de ódio (DO) online tem crescido dramaticamente nas redes sociais e a sua disseminação descontrolada tem motivado os investigadores a desenvolver diversos métodos para sua deteção automática. No entanto, a deteção de DO online em português ainda é muito pouco estudada. Para preencher essa lacuna, exploramos diferentes modelos que fazem uso da arquitetura BERT e se mostraram adequados na literatura para abordar esta tarefa. Além de testar modelos monotarefa, também exploramos modelos multitarefa, que usam informação acerca de outras categorias para aprender DO. Desenvolvemos o HateBERTimbau, uma versão retreinada do BERTimbau mais direcionada para linguagem de redes sociais, que inclui potencial DO que visa as comunidades afrodescendentes, ciganas e LGBTQI+. As experiências realizadas foram baseadas no CO-HATE e no FIGHT, dois corpora constituídos por mensagens publicadas pela comunidade online portuguesa anotadas relativamente à presença de DO e de outras categorias relacionadas. Os resultados alcançados mostram a importância de considerar a concordância entre anotadores sobre os dados utilizados para desenvolver modelos de deteção de DO. Comparando diferentes subconjuntos de dados utilizados para o treino dos modelos, mostrou-se que, em geral, uma maior concordância nos dados leva a melhores resultados. O HATEBERTimbau superou consistentemente o BERTimbau em ambas as coleções de dados, confirmando que o pré-treino adicional do BERTimbau foi uma estratégia bem-sucedida em obter um modelo de linguagem mais direcionado para nossa tarefa. A aprendizagem simultânea de tarefas relacionadas com o DO permitiu obter melhores resultados e também foi demonstrado o potencial de modelos específicos para um grupo alvo.

Palavras Chave

Classificação de texto; Deteção de Discurso de Ódio; Aprendizagem Supervisionada; Transferência de

Aprendizagem; Aprendizagem Profunda; Aprendizagem multitarefa.

Contents

1 Introduction **1**

1.1 Context and Motivation . . . 3

1.2 Generic Methodological Approach of Automatic Hate Speech (HS) Detection . . . 4

1.3 Research Questions . . . 6

1.4 Goals of the Work . . . 6

1.5 Research Methodology . . . 7

1.6 Document Structure . . . 8

2 Fundamental Concepts **9**

2.1 Natural Language Processing . . . 11

2.2 Machine Learning . . . 11

2.3 Deep Learning . . . 12

2.3.1 Convolutional Neural Networks . . . 12

2.3.2 Transformer Networks . . . 13

2.3.2.A Attention Mechanisms . . . 13

2.3.2.B Positional Encoding . . . 14

2.4 Transfer Learning . . . 14

2.5 Word Embeddings . . . 15

2.6 BERT . . . 15

2.6.1 Contextualized Embeddings . . . 15

2.6.2 Self-supervised Learning . . . 16

2.6.3 Pretraining BERT . . . 17

2.6.4 Adapting BERT . . . 17

3 Related Work **19**

3.1 Systematic Literature Reviews on Hate Speech Detection . . . 21

3.2 Classical Methods . . . 25

3.3 Deep Learning Methods . . . 27

4 Data **33**
4.1 Annotation Guidelines . . . 35
4.2 CO-HATE Corpus . . . 37
4.2.1 Annotators Profile . . . 38
4.2.2 Annotation Results . . . 38
4.3 FIGHT Corpus . . . 40
4.3.1 Annotators Profile . . . 41
4.3.2 Annotation Results . . . 41
5 Modelling **43**
5.1 BERT-LinearLayer . . . 45
5.2 BERT-CNN . . . 45
5.3 BERT-Attention . . . 46
5.4 Pre-trained BERT Models . . . 47
6 Results **49**
6.1 Experimental Setup . . . 51
6.2 Results for the CO-HATE Corpus . . . 52
6.2.1 Single-task Experiments . . . 52
6.2.2 Multitask Experiments . . . 55
6.2.3 Data Agreement Impact On Results . . . 58
6.2.4 Target-Specific Model . . . 58
6.2.5 Error Analysis . . . 59
6.3 Results for the FIGHT Corpus . . . 61
6.3.1 Single-task Experiments . . . 61
6.3.2 Multitask Experiments . . . 64
6.3.3 Data Agreement Impact on Results . . . 67
6.3.4 Target-Specific Model . . . 68
6.3.5 Error Analysis . . . 68
6.4 Summary and Discussion . . . 70
7 Conclusion **73**
7.1 Conclusions . . . 75
7.2 Limitations and Future Work . . . 76
Bibliography **77**

List of Figures

1.1 Stages of Cross-Industry Standard Process for Data Mining (CRISP-DM) [1]. 7

2.1 Convolutional Neural Network (CNN) architecture for Text Classification [2]. 13
2.2 BERT processes input through attention layers to produce contextualized embeddings [3]. 16

3.1 Number of publications per year from 2000 to 2021 related to automatic HS detection . . 22
3.2 Statistics on the types of Machine Learning (ML) approaches used for HS detection (e.g., supervised, semi-supervised or unsupervised) [4]. 23
3.3 Statistics of algorithm types used for HS detection [4]. 23
3.4 Statistics of features employed by ML/Deep Learning (DL) algorithms [4]. 24

5.1 BERT-LinearLayer model structure [5]. 45
5.2 BERT-CNN model structure [6]. 46
5.3 BERT-Attention model structure [7]. 47

6.1 Confusion matrix of the best performing model on CO-HATE corpus: BERT-Attention-CS using **HATEBERTimbau** trained with all training data. 59
6.2 Confusion matrix of the best performing model on FIGHT corpus: BERT-LinearLayer using **HATEBERTimbau** trained with all training data. 68

List of Tables

4.1 Dimensions and attributes of CO-Hate and FIGHT corpora annotation scheme. 35
4.2 Definitions of the discourse type concepts adopted in the annotation process of CO-Hate and FIGHT corpora. 36
4.3 Distribution of CO-HATE corpus according to the mentioned target. 37
4.4 Proportion of messages classified as Hate Speech in CO-HATE training set for each annotator and for all annotators. 38
4.5 Inter-Annotator Agreement (IAA) of all discourse types for different groups of annotators on CO-Hate corpus. 39
4.6 Pairwise IAA of the Hate Speech class on CO-HATE corpus. 39
4.7 Distribution of the final FIGHT corpus according to the mentioned target. 41
4.8 Proportion of messages classified as Hate Speech in FIGHT training set for each annotator and for all annotators. 41
4.9 IAA of all discourse types for different groups of annotators on FIGHT corpus. 42
4.10 Pairwise IAA of the Hate Speech class on FIGHT corpus. 42

6.1 Performance of the single-task models using **BERTimbau** for the different sets of training data in the test set of CO-HATE corpus. 53
6.2 Performance of the single-task models using **HATEBERTimbau** and the best single-task result using **BERTimbau** for the different sets of training data in the test set of CO-HATE corpus. 54
6.3 Performance of the multitask models using **BERTimbau**, the best single-task model using **BERTimbau** and BERT-Attention using **BERTimbau** for the different sets of training data in the test set of CO-HATE corpus. 56
6.4 Performance of the multitask models using **HATEBERTimbau**, the best single-task model and BERT-Attention using **HATEBERTimbau** for the different sets of training data in the test set of CO-HATE corpus. 57

6.5 Results of all experiments using for training A+B+C and D+E in the test set of CO-HATE corpus. . . . 58
6.6 Results of all experiments using for training A+B+D+E and A+B+C+D+E in the test set of CO-HATE corpus. . . . 59
6.7 Results of the African descent-directed HS Model and the Generic HS Model in the African descent-directed messages of the CO-HATE test set. . . . 59
6.8 Performance of the single-task models using **BERTimbau** for the different sets of training data in the test set of FIGHT corpus. . . . 62
6.9 Performance of the single-task models using **HATEBERTimbau** and the best single-task model using **BERTimbau** for the different sets of training data in the test set of FIGHT corpus. . . . 63
6.10 Performance of the multitask models using **BERTimbau**, the best single-task model using **BERTimbau** and BERT-Attention using **BERTimbau** for the different sets of training data in the test set of FIGHT corpus. . . . 65
6.11 Performance of the multitask models using **HATEBERTimbau** (in gray), the best single-task model and BERT-Attention using **HATEBERTimbau** for the different sets of training data in the test set of FIGHT corpus. Best multitask model results using **BERTimbau** are also shown in the cases where they obtain better results. . . . 66
6.12 Results of all experiments using for training A+C and B+D+E in the test set of FIGHT corpus. . . . 67
6.13 Results of all experiments using for training B+C+D+E and A+B+C+D+E in the test set of FIGHT corpus. . . . 67
6.14 Results of the LGBTQI+-directed HS Model and Generic HS Model in the LGBTQI+-directed messages of the FIGHT corpus test set. . . . 68

Acronyms

AI	Artificial Intelligence
ALBERT	A LiteBidirectional Encoder Representations from Transformers
BERT	Bidirectional Encoder Representations from Transformers
BOW	Bag-of-Words
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CO-HATE	**C**ounter, **O**ffensive and **Hate** speech
CRISP-DM	Cross-Industry Standard Process for Data Mining
DL	Deep Learning
DM	Data Mining
DNN	Deep Neural Network
DT	Decision Trees
ELMO	Embeddings from Language Models
FIGHT	**FI**ndin**G** **H**ate Speech in **T**witter
GBDT	Gradient Boosted Decision Trees
GPT-2	Generative Pre-trained Transformer 2
GRU	Gated Recurrent Unit
HS	Hate Speech
IAA	Inter-Annotator Agreement
KNN	K-Nearest Neighbors
LR	Logistic Regression
LSTM	Long Short-Term Memory

ML Machine Learning

NB Naive Bayes

NLP Natural Language Processing

OHS Online Hate Speech

POS Part-of-Speech

RF Random Forest

RNN Recurrent Neural Network

RoBERTa A Robustly Optimized BERT Pretraining Approach

SVM Support Vector Machines

TF-IDF Term Frequency–Inverse Document Frequency

XLM-RoBERTa Cross-Lingual Language Robustly Optimized BERT Pretraining Approach

mBERT Multilingual BERT

1

Introduction

Contents

1.1 Context and Motivation 3
1.2 Generic Methodological Approach of Automatic Hate Speech (HS) Detection 4
1.3 Research Questions 6
1.4 Goals of the Work 6
1.5 Research Methodology 7
1.6 Document Structure 8

In this chapter, we present the context and motivation of this work (Section 1.1), the generic pipeline of automatic HS detection (Section 1.2), the research questions we intend to answer (Section 1.3), the main goals (Section 1.4), the methodology used (Section 1.5), and finally the structure of this document (Section 1.6).

1.1 Context and Motivation

Due to the easy access of the platforms, the potential anonymity of the users, and the increased willingness of people to express their opinions online, social media environments are fertile to disseminate aggressive and harmful content [4, 8, 9]. In particular, Online Hate Speech (OHS) is becoming a major concern in modern society and poses a big threat to democracy and human rights. To combat its recent rise, the Natural Language Processing (NLP) and Artificial Intelligence (AI) research communities have been developing methods and tools to automatically detect HS [8, 10]. By creating systems that automatically identify OHS, this phenomenon can be detected with a minimum of human intervention, faster and perhaps more efficiently, thus decreasing the risks of exposure to OHS. However, the non-existence of a unique and consensual definition of HS [8] makes its distinction from other related phenomena, such as Offensive Speech, difficult either for humans or algorithms [11]. For the purpose of this work, HS is defined according to the following coexisting conditions [12]:

- HS has a specific target that can be mentioned explicitly or implicitly, which corresponds to vulnerable or historically marginalized groups or individuals targeted for belonging to those groups;
- HS typically spreads or supports hatred, or incites violence against the targets, by disparaging, humiliating, discriminating, or even threatening them based on specific identity factors (e.g., religion, ethnicity, nationality, race, color, descent, gender, sexual orientation);
- HS can be expressed both explicitly (or overtly) and implicitly (or covertly).

The major difficulty of this task is related to the fact that most of the hateful comments in social media are covert or implicit, and their interpretation requires information on the social practice context [13]. Moreover, demographic features such as the first language, age, education, and social identity can result in subjective and biased annotations in the corpora used to produce OHS detection systems [14].

Recently, the Commissioner for Human Rights, in a memorandum related to Portugal, has noted that, despite the information being provided by civil society organisations indicates low rates of reporting of HS, there is a rise in the number of racially motivated hate crimes and HS [15]. This highlights the need of developing systems that automatically detect OHS propagated by the Portuguese community, which is what we propose to do in this work. Despite the great popularity of OHS detection, few studies have been specifically dedicated to the analysis and detection of European Portuguese OHS. In fact, there

is a lack of resources (particularly annotated corpora) specifically designed to support OHS detection in European Portuguese [8, 16].

This work was developed under the scope of the FCT-funded project *HATE COVID-19.PT - Detecting Overt and Covert Hate Speech in Social Media*,[1], focused on the analysis and detection of OHS in European Portuguese. Two large annotated corpora from social media were created in the scope of this project: **C**ounter, **O**ffensive and **Hate** speech (CO-HATE) and **FI**ndin**G** **H**ate Speech in **T**witter (FIGHT). Both corpora focus on the expression of afrophobia, romaphobia, and LGBTQIphobia by the Portuguese online community, since the Afro-descendant, Roma, and LGBTQI+ communities are among the most commonly reported targets of HS in Portugal [17]. The Inter-Annotator Agreement (IAA) obtained among the annotators on the HS class is relatively low, which reflects the plurality of subjective views on the HS concept. This motivated us to investigate how does the selection of different perspectives for training OHS detection models affects the results of OHS detection. We will also experiment to leverage other categories of the messages, provided by the annotation, to benefit the OHS detection task, which is also scarcely unexplored in this context, especially considering European Portuguese.

Recent work on OHS detection relies mostly on the use of Deep Learning (DL) methods for both feature extraction and training of classifiers [18–21]. As reported in literature [5], the use of models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), among others, suffers from the lack of labeled data. Transfer learning approaches can overcome this issue since they do not require large amounts of labelled data to train models. Furthermore, they are not so time-consuming, and can outperform all the remaining approaches [5, 22, 23]. In this work, we focus on different transfer learning approaches based on the Bidirectional Encoder Representations from Transformers (BERT) [24] architecture. In particular, we compare the results of three single-task models – BERT-LinearLayer, BERT-CNN, and BERT-Attention – and two multitask models – BERT-Attention-CS and BERT-Attention-SNT – on the task of detecting OHS using CO-HATE and FIGHT corpora. We also study the impact of retraining the BERT pre-trained model on the performance of the models and compare the performance between models that detect OHS against a particular target group and models that detect OHS in general.

1.2 Generic Methodological Approach of Automatic HS Detection

HS detection is normally modelled as a text classification task [18, 25, 26]. We describe the generic pipeline of a text classification task as specified in [27]. The initial input consists of a raw text dataset. Generally, text datasets can be considered as $D = \{X_1, X_2, \ldots, X_N\}$, where X_i refers to a data point with some number of sentences, such that each sentence includes some number of words with some

[1] https://hate-covid.inesc-id.pt

number of letters. Each point is classified with a label value from a set of different discrete value indices [28].

The HS classification task is decomposed in four essential parts:

Dataset Collection and Preparation It is the first step of the HS detection pipeline. Often, datasets are collected from social media platforms. Preprocessing is performed according to dataset structure and quality. Typically, this involves filtering and normalization steps of the textual inputs, which includes tokenization, stopwords removal, misspelling correction, noise removal, stemming, and lemmatization, among other tasks. The dataset may be provided initially so that collecting it is not required. As part of data preparation, training and testing parts of the dataset should be distinguished for the subsequent Machine Learning (ML) step.

Feature Engineering It is the next phase of the analysis where appropriate features are extracted from the textual inputs so that unstructured text sequences are converted into structured features. The most common techniques for feature extraction are Term Frequency–Inverse Document Frequency (TF-IDF), semantic, lexical, topic modeling, sentiment, Bag-of-Words (BOW), and word embedding (Word2Vec, GloVe, FastText) [26, 29, 30]. Sometimes, dimensionality reduction is also applied to reduce the time and memory complexity [31].

Model Choice and Training To effectively determine the most efficient model for a text classification application, we must have a complete conceptual understanding of each algorithm. Model training is one of the most crucial steps of the text classification pipeline where a ML/DL model is trained on the training dataset. Several classifiers can be tailored based on task requirements: Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), CNN, Recurrent Neural Network (RNN), BERT-based classifiers, etc. Word embeddings are often jointly used in a neural network model as an embedding layer, where textual data is converted into numerical vectors, which helps to enhance DL performance. The output of the ML/DL model can be either a binary decision (hate vs non-hate speech) or a multi-class output where the model discriminates various types of HS and non-HS.

Evaluation It is the final part of the text classification pipeline where the performance of the ML/DL model is estimated. Several evaluation metrics are used for this purpose: Accuracy, F1-score, Precision, Recall, Matthews Correlation Coefficient (MCC), receiver operating characteristics (ROC), and area under the ROC curve (AUC) are some of the possibilities.

1.3 Research Questions

This work aims at answering the following research questions:

Q1 How do the current state-of-art DL methods used for HS detection in other languages (such as English) perform in the detection of European Portuguese OHS?

Since there are not many HS detection works focusing on the European Portuguese language, it is not known how the most popular and successful methods being used for other languages perform in this particular language. With this study we aim to understand if OHS can be successfully detected by any of the current state-of-art DL methods for other languages.

Q2 Do HS targets influence the classification performance? More specifically, do HS detection models developed for a specific target group perform better than generic HS detection models?

We want to test if by having models that detect HS against a particular target group, the results achieved are better than having a general HS detection model. We hypothesize that the HS directed to each target may have its particular language characteristics and maybe having different models for different subsets of HS produces more accurate results.

Q3 Do the models trained for detecting OHS and other relevant properties associated with this concept (e.g. Sentiment Polarity) perform better than single-task OHS detection models?

We want to evaluate if multitask models can make use of the knowledge about other related categories to detect OHS more successfully than single-task OHS detection models.

1.4 Goals of the Work

As already stated, the detection of HS is of major importance. The goals here presented are moved by the need of systems that can successfully perform that task and are directly related to the research questions. The goals of the proposed project are as follows:

Goal 1 Identify which methods are being used to automatically detect HS and which ones demonstrate better performance. An overall overview of what is being done in this field is of major importance for the execution of this work.

Goal 2 Develop automatic models that can detect European Portuguese OHS. For this purpose, we will not only use existing generic language models, but we also intend to develop a language model more tailored to the European Portuguese social media and potential OHS domains. We are also interested in exploring OHS detection models for a specific target group.

Goal 3 Develop multitask learning models that handle not only the OHS detection task but also identify some other related category. The knowledge of that category will be applied to our target task (HS detection) and with these models we can evaluate the benefits of sharing that knowledge.

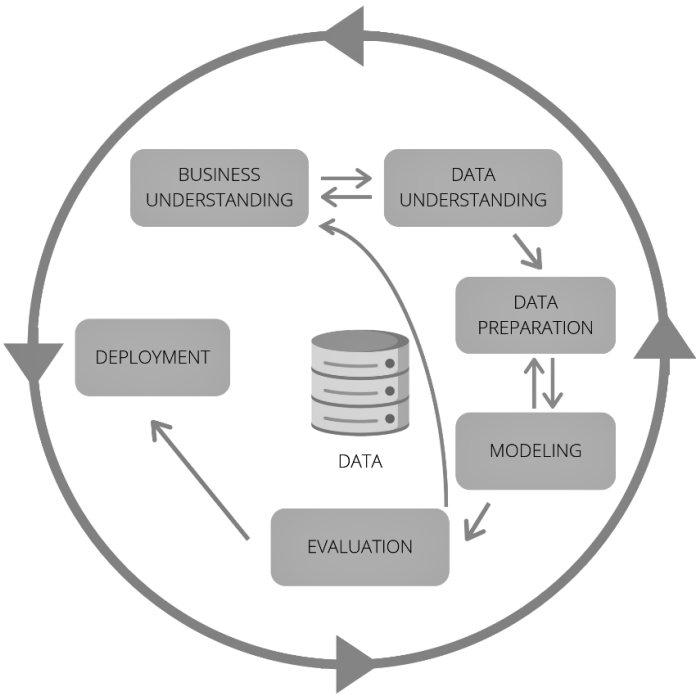

Figure 1.1: Stages of CRISP-DM [1].

1.5 Research Methodology

The research methodology that we follow in this work is inspired by Cross-Industry Standard Process for Data Mining (CRISP-DM), which is a Data Mining (DM) methodology and one of the most popular methodologies used in DM projects worldwide [32]. CRISP-DM is a cyclic process, which is divided into six phases that can be performed countless times, depending on the results of the evaluation. The main stages of the CRISP-DM life cycle are depicted in Figure 1.1.

In the first stage (Business Understanding), we start by contextualizing HS detection and formulating the research questions and goals of this work. Then we proceed to the identification of the most popular methods in literature concerning the detection of HS and related concepts, exploring both classical and DL approaches. After getting a deeper understanding of the methods being used, the research questions and goals are updated and refined.

In the second stage (Data Understanding), we collect, describe, and explore the datasets that will support our experiments, namely CO-HATE and FIGHT corpora, which are described in Chapter 4. We can return to Business Understanding if we need to better understand the scope of the problem.

We opted to skip the third stage (Data Preparation), in which the datasets should be cleaned and pre-processed, since some initial experiments showed that our OHS detection models did not benefit from these steps.

The fourth stage (Modelling) consists in the selection and training of the HS detection models. DL algorithms were used to train the classification models, a choice motivated by the promising results from previous similar studies. All the models developed are described in Chapter 5.

In the final step (Evaluation), the results of the models are evaluated and analysed (Chapter 6) and, if necessary, the previous steps are revisited with the aim of improving the results.

If the developed models are in accordance with the proposed goals, we pass to the final step (Deployment), where we document a summary and the main conclusions of this work and suggest future improvements (Chapter 7).

1.6 Document Structure

The structure of this document is described as follows:

- **Chapter 2** - Some concepts fundamental to better understand what is discussed in this work are presented and explained;
- **Chapter 3** - A literature review of works related to the automatic detection of HS and related concepts is performed;
- **Chapter 4** - The datasets that we will use in the development of our OHS detection models are presented;
- **Chapter 5** - The models that we propose to develop are described;
- **Chapter 6** - The evaluation of the models developed is performed through the analysis of the results;
- **Chapter 7** - The conclusions and future work of this work are stated.

2

Fundamental Concepts

Contents

2.1 Natural Language Processing 11
2.2 Machine Learning 11
2.3 Deep Learning 12
2.4 Transfer Learning 14
2.5 Word Embeddings 15
2.6 BERT 15

This chapter provides an introduction to some concepts useful for understanding what is discussed in this document: NLP in Section 2.1, ML in Section 2.2, DL in Section 2.3, Transfer Learning in Section 2.4, Word Embeddings in Section 2.5, and BERT in Section 2.6.

2.1 Natural Language Processing

NLP is a subfield of AI that concerns computational approaches for processing, understanding, and generating human languages. The word "natural" is used to contrast natural languages, which emerged naturally tens of thousands of years ago and have evolved organically ever since, with formal languages, which have strict syntax and semantics. In this sense, all the languages humans speak are natural while languages such as the Python programming language are formal. NLP includes a range of algorithms, tasks, and problems that take human-produced text as an input and produce as output some useful information, such as labels, semantic representations, and so on. The task addressed in this thesis is Text Classification.

2.2 Machine Learning

ML is a subfield of AI related to constructing computer programs that can learn from data without being explicitly programmed [33]. This includes learning a general function that maps inputs to outputs based on past experience (supervised learning) and drawing hidden patterns and structures from data (unsupervised learning).

More formally, the task of supervised learning is defined as follows: given a training set of example input-output pairs, $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, where each pair was generated by an unknown function $y = f(x)$, the goal is to discover a function h that approximates the true function f.

Since the data used in this work are labelled, we will focus on supervised ML, which is the main paradigm for training NLP models. For example, in our case the inputs will be text documents, each one accompanied by an output, called a label, saying "Hate" or "No Hate". The algorithm produces a function that, when given a new text, predicts the appropriate label.

In ML, to develop and evaluate models, it is common to separate the data into three different dataset splits — train, validation, and test sets. A train (or training) set is the main dataset used to train the models. A validation set (also called a dev or development set) is used for model selection. Model selection is a process where appropriate models are selected among all possible models that were trained using the train set. When a trained model fits the train set so well that it loses its generalizability, we say it is overfitting. The validation set gives a proxy for the model's generalizability. The validation set is also used for tuning hyperparameters, parameters about a ML algorithm or about a model that is

being trained. Finally, a test set is used to evaluate the model using a new set of data since it could be possible to overfit the model to the validation set.

2.3 Deep Learning

DL is a subfield of ML that usually uses deep neural networks. These neural network models are called "deep" because they consist of multiple layers. By having many stacked layers, deep neural networks can learn complex representations of data and can capture highly complicated relationships between the input and the output.

Neural networks receive feedback (how close the output is to the desired output) and adjust their internal parameters called weights so that they can produce more accurate outputs. This adjusting of parameters is done through loss functions and optimization.

A loss function is a function that measures how far an output of a ML model is from a desired one. The difference between an actual output and a desired one is called the loss. Neural networks change their internal parameters to make the loss smaller. They do this for each and every instance in the training data, so that they can produce more accurate predictions. This requires multiple cycles, called epochs, through the full training data.

The process where a neural network computes an output from an input using the current set of parameters is called the forward pass. The way the loss is fed back to the neural network is called backpropagation. The algorithm stochastic gradient descent is often used to minimize the loss. The process where the loss is minimized is called optimization, and the algorithm used to achieve this is called the optimizer.

DL models have achieved state-of-the-art results across many domains, including a wide variety of NLP task, such as Text Classification. In Section 2.3.1 we introduce CNN, and in Section 2.3.2 we introduce Transformer Networks. Both types of neural networks are used in the models we developed.

2.3.1 Convolutional Neural Networks

A CNN is a type of neural network that involves a mathematical operation called convolution, which detects local patterns that are useful for the Text Classification task. A CNN usually consists of one or more convolutional layers, which are called feature maps and can be stacked to provide multiple filters on the input, and pooling layers, which are responsible for aggregating the result of convolution. The most common pooling method is max pooling where the maximum element in the pooling window is selected. In order to feed the pooled output from stacked featured maps to the next layer, the maps are flattened into one column. The final layers in a CNN are typically fully connected/linear layers (used for compressing vectors). In general, during the back-propagation step of a CNN, both the weights

and the feature detector filters are adjusted. In Figure 2.1 it is illustrated a CNN architecture for Text Classification which contains a word embedding as input layer, 1D convolutional layers, 1D pooling layers, fully connected layers, and finally an output layer.

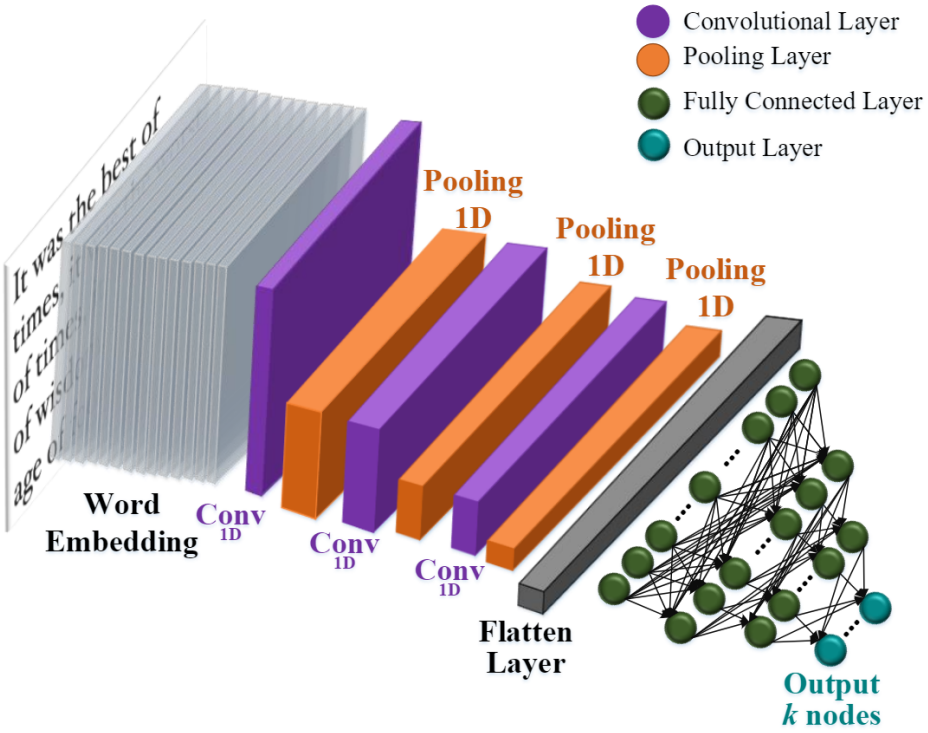

Figure 2.1: CNN architecture for Text Classification [2].

2.3.2 Transformer Networks

The Transformer [34] is a recent type of encoder-decoder neural network architecture based on the concept of self-attention. It is the most important NLP model since it appeared in 2017 [3]. It is a powerful model itself, but it is also used as the underlying architecture that powers numerous modern NLP pre-trained models, including Generative Pre-trained Transformer 2 (GPT-2) and BERT. The main ideas behind the Transformer architecture are presented in the next sections.

2.3.2.A Attention Mechanisms

Attention is a mechanism in neural networks that focuses on a specific part of the input and computes its context-dependent summary. It consists in having some sort of key-value store that contains all of the input's information and then looking it up with a query (the current context). The stored values are not just a single vector but usually a list of vectors, one for each token, associated with corresponding

keys. This effectively increases the size of the “memory” the decoder can refer to when it is making a prediction.

The Transformer uses a type of attention mechanism named self-attention. Each summary produced by self-attention takes all the tokens in the input sequence into consideration, but with different weights. Using an analogy, self-attention produces summaries through random access over the input. This is in contrast to RNNs, which allow only sequential access over the input, and is one of the key reasons why the Transformer is such a powerful model for encoding and decoding natural language text.

Multi-head attention is an extension of self-attention that computes multiple sets of attention weights to mix values that focus on different aspects of the input. The final embeddings are a combination of summaries generated this way.

2.3.2.B Positional Encoding

Self-attention operation is completely independent of positions but the meaning of a natural language sentence depends a lot on how its words are ordered. The Transformer model solves this problem by generating artificial embeddings named positional encoding that differ from position to position and adding them to word embeddings before they are fed to the layers. These embeddings are either generated by some mathematical function (such as sine curves) or learned during training per position.

2.4 Transfer Learning

In ML, Transfer Learning is a collection of related techniques to improve the performance of a ML model in a task using data and/or models trained in a different task. Transfer learning always consists of two or more steps — a ML model is first trained for one task (called pretraining), which is then adjusted and used in another (called adaptation). If the same model is used for both tasks, the second step is called fine-tuning, because the same model is being slightly tuned but for a different task.

Transfer learning has become the dominant way for building high-quality NLP models in the past few years for two main reasons. Firstly, thanks to powerful neural network models such as the Transformer and self-supervised learning, it became possible to bootstrap high-quality embeddings from an almost unlimited amount of natural language text. These embeddings take into account the structure, context, and semantics of natural language text to a great extent. Secondly, thanks to Transfer Learning, it is possible to incorporate these powerful pre-trained language models into their NLP applications. The advent of these new technologies (the Transformer, self-supervised learning, pre-trained language models, and Transfer Learning) moved the field of NLP to a completely new stage and pushed the performance of many NLP tasks to a near-human level.

Domain adaptation [35], a particular case of Transfer Learning, is a technique where a ML model is trained in one domain (e.g., news) and adapted to another domain (e.g., social media).

2.5 Word Embeddings

Word embeddings are vector representations of words that are learned so that semantically similar words share similar representations, that is, they have a close proximity in a high-dimensional space. These representations are trained on an independent, large textual corpus without any training signals, using algorithms such as Skip-Gram and CBOW, often collectively called Word2vec. After these word embeddings are trained, downstream NLP tasks can use them as the input to their models (which are often neural networks). Because these embeddings already capture semantic relationships between words, these tasks no longer need to learn how the language works from scratch, which gives them the upper hand in the task they are trying to solve. The model can focus on learning higher-level concepts that cannot be captured by word embeddings and the task-specific patterns learned from the given annotated data. Taking the outcome of one task (the training of the embeddings) and transferring the knowledge gleaned from it to another one (i.e., Text Classification, or any other NLP tasks) is a form of Transfer Learning.

2.6 BERT

BERT, a Transformer-based pre-trained language model, is by far the most popular and most influential pre-trained language model to date that revolutionized how people train and build NLP models. We will first introduce contextualized embeddings and why they are important, then move on to explaining self-supervised learning, which is an important concept in pretraining language models. We'll cover two self-supervised tasks used for pretraining BERT, namely, masked language models and next-sentence prediction, and cover ways to adapt BERT for NLP applications.

2.6.1 Contextualized Embeddings

Words in natural language may have more than one meaning, depending on their context. Word embeddings cannot take context into account since all the different meanings are compressed into a single vector. Due to this limitation, NLP researchers started exploring ways to transform the entire sentence into a series of vectors that consider the context, called contextualized embeddings. With these representations, the same word has different vectors assigned, helping downstream tasks disambiguate different uses of the word. The biggest breakthrough in contextualized embeddings was achieved by BERT.

The core idea of BERT is simple: it uses the Transformer (the Transformer encoder, to be precise) to transform the input into contextualized embeddings. The Transformer transforms the input through a series of layers by gradually summarizing the input. Similarly, BERT contextualizes the input through a series of Transformer encoder layers. This is illustrated in Figure 2.2.

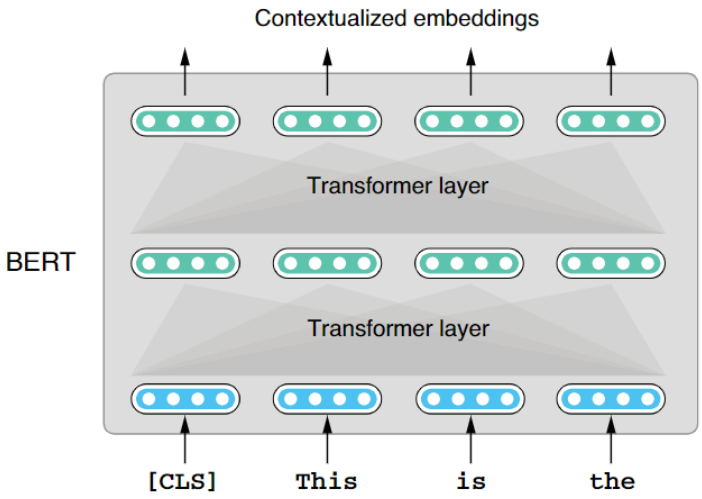

Figure 2.2: BERT processes input through attention layers to produce contextualized embeddings [3].

Because BERT is based on the Transformer architecture, it inherits all the strengths of the Transformer. Its self-attention mechanism enables it to "random access" over the input and capture long-term dependencies among input tokens. Unlike traditional language models that can make predictions in only one direction, the Transformer can take into account the context in both directions.

2.6.2 Self-supervised Learning

BERT is trained like word embeddings: surrounding words are predicted with word embeddings. The main interest is not the task performed but the product obtained, which are the word embeddings derived as the parameters of the model. This type of training paradigm where the data itself provides training signals is called self-supervised learning, or simply self-supervision, in modern ML. In self-supervised learning, the model is trained in such a way that it minimizes the loss function defined by the training signal. This training signal come from the data itself with no human intervention. With increasingly larger datasets and more powerful models, self-supervised learning has become a popular way to pretrain NLP models in the past several years.

Self-supervised learning works prosper because the knowledge required to predict the surrounding words ranges from simple collocation/association, syntactic and grammatical, and semantic. Also, there is virtually no limit on the amount of data used for self-supervision.

2.6.3 Pretraining BERT

BERT is a Transformer encoder that transforms the input into a series of embeddings that take context into account. BERT is trained on the masked language model (MLM) objective, where the words are dropped randomly in a given sentence and the model has to predict what the dropped word is. Specifically, after replacing a small percentage of words in a sentence with a special placeholder, BERT uses the Transformer to encode the input and then uses a feed-forward layer and a softmax layer to derive a probability distribution over possible words that can fill in that blank. By telling the model to solve this fill-in-the-blank type of task over a huge amount of textual data, the neural network model is trained so that it can produce contextualized embeddings that incorporate deep linguistic knowledge. BERT is pre-trained not just with the masked language model but also with another type of task called next-sentence prediction (NSP), where two sentences are given to BERT and the model is asked to predict whether the second sentence is the "real" next sentence of the first. The rationale behind this task is that by training with this objective, the model will learn how to infer the relationship between two sentences.

2.6.4 Adapting BERT

As previously mentioned, at the second stage of Transfer Learning, a pre-trained model is adapted to the target task so that the latter can leverage signals learned by the former. There are two main ways to adapt BERT to individual downstream tasks: fine-tuning and feature extraction. In fine-tuning, the neural network architecture is slightly modified so that it can produce the type of predictions for the task in question, and the entire network is continuously trained on the training data for the task so that the loss function is minimized. BERT "inherits" the model weights learned through pretraining, instead of being initialized randomly and trained from scratch. In this way, the downstream task can leverage the powerful representations learned by BERT through pretraining on a large amount of data. For downstream tasks to be able to extract representations for a sentence, BERT prepends a special token [CLS] (for classification) to every sentence at the pretraining phase. The hidden states of BERT can be extracted with this token and be used as the representation of the sentence. As with other classification tasks, a linear layer can compress this representation into a set of "scores" that correspond to how likely each label is the correct answer. This type of linear layer, which is plugged into a larger pre-trained model such as BERT, is often called a head. In other words, a classification head is being attached to BERT to solve a sentence-prediction task. The weights for the entire network (the head and BERT) are adjusted so that the loss function is minimized. This means that the BERT weights initialized with the pre-trained ones also are adjusted (fine-tuned) through backpropagation.

Another way to adapt BERT for downstream NLP tasks is feature extraction. Here BERT is used to extract features, which are simply a sequence of contextualized embeddings produced by the final layer

of BERT and these vectors are fed to another ML model as features.

In general, a better performance is obtained in the downstream task if the BERT parameters are fine-tuned because by doing so, BERT is also learning to get better at the task at hand.

3

Related Work

Contents

3.1 Systematic Literature Reviews on Hate Speech Detection 21

3.2 Classical Methods 25

3.3 Deep Learning Methods 27

This chapter focuses on the literature review, useful for the development of this work. To streamline the process of getting a general overview of the field of automatic HS detection, we start by focusing on the most relevant literature reviews on the topic (Section 3.1). Next, we present the research specifically related to the detection of HS and related concepts, such as cyberbullying, offensive language, and abusive language. Section 3.2 addresses classical methods and Section 3.3 addresses DL methods.

3.1 Systematic Literature Reviews on Hate Speech Detection

Systematic reviews use rigorous methodological approaches, guaranteeing transparency, greater breadth of studies included, greater objectivity, and reduction of implicit researcher bias [36]. For this reason, we only present in this section survey papers that follow a systematic review-based methodology.

Fortuna et al. [8] presented a critical overview of how the automatic detection of HS evolved over the past few years. They argued that the existing works regard this problem as a ML classification task. Due to the lack of standard datasets, they found that there is no particular approach proving to reach better results among the several articles. They observed that the majority of the studies only consider generic features and do not use particular features for HS. This can be problematic because HS is a complex social phenomenon in constant evolution and supported in language nuances. This review is very relevant, but it does not provide an updated overview of the approaches since it was carried out at the end of 2017.

Poletto et al. [10] systematically analyzed HS resources including their development methodology, topical focus, language coverage, and other factors. The authors mainly focused on HS corpora and not on the detection methods. Although it is not relevant for our review of the methods, this work mentions several works useful for our literature review.

Jahan et al. [4] presented a recent systematic literature review related to HS detection. The last search for completing the article collection was performed on 18 March 2021. After all the phases of their systematic review, 463 articles were considered for the analysis. From their analysis we found important the following points:

- As we can see in Figure 3.1, before 2016, the number of papers associated with HS detection was low and there was no work related to DL. However, since 2017 the number of published documents raised rapidly with a steady increase of the DL-based HS detection approach. A total of 96 documents were found from 2017 to 2021 using DL for HS detection, indicating a trend of almost doubling the number of DL-based work each year. The relatively small value in 2021 is due to the fact that the collection of new documents stopped in the beginning of 2021.

- Figure 3.2 reveals that most of the works adopted supervised methods (73%). They observed that all three types of methods, supervised, unsupervised, and semi-supervised, can achieve high-

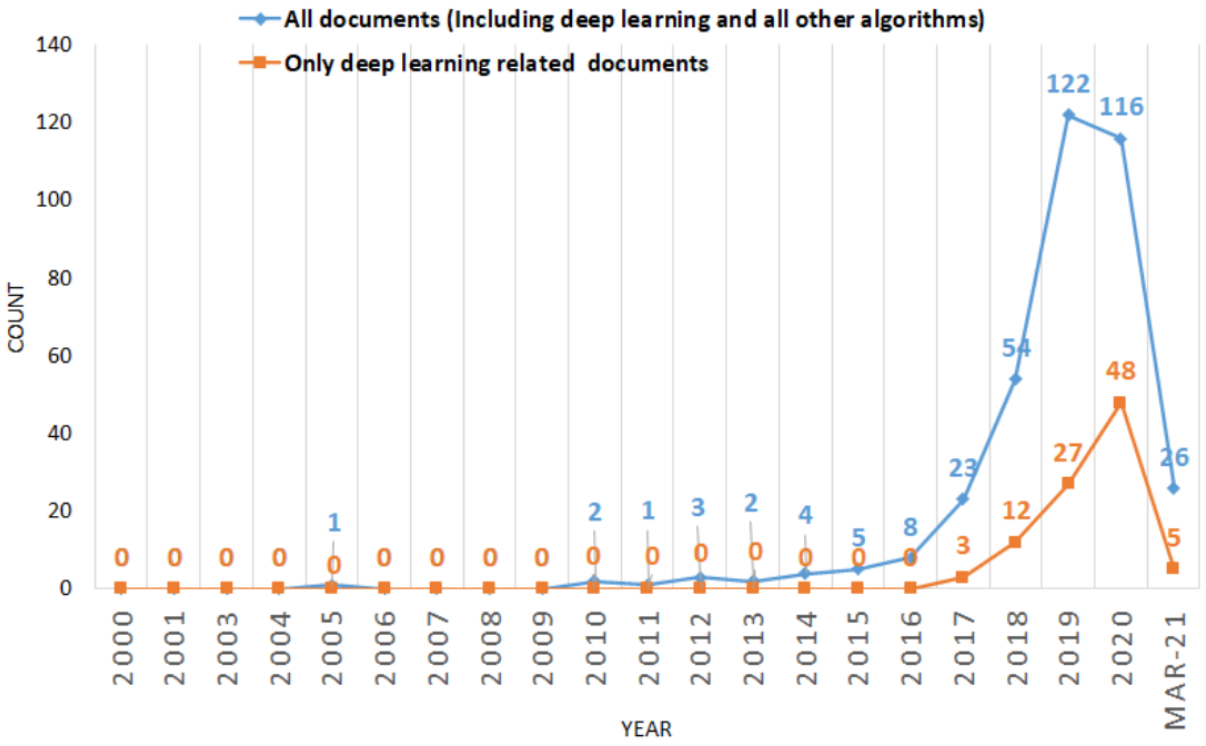

Figure 3.1: Number of publications per year from 2000 to 2021 related to automatic HS detection (the blue line represents all 463 documents including DL and other ML approaches, and the orange line represents 96 documents related to DL methods) [4].

performance Accuracy, and there is no substantial evidence to favour one over another, whereas only the context of data (e.g., availability and quality of training samples) can play a role in deciding about the suitability of one type of approach over another. Nevertheless, it is worth mentioning the popularity of the supervised approach over the others, which may be due to the multiplication of benchmarking datasets and ML/DL platforms that promote the supervised approach.

- In Figure 3.3, we can see that the Support Vector Machines (SVM) method emerges as the most popular HS detection model covering 29% of total records. The use of DL models started to rise from 2017 to quickly cover about 22% of total identified records. On the other hand, LR (20%) and NB (14%) were also among popular ML methods investigated by the researchers. They also noticed that in many DL related methods, non-DL models were often employed as baseline to compare the performance of the investigated DL model.

- When it comes to the features employed (Figure 3.4), TF-IDF based features cover 29% of the total records. However, word embedding models, which have widely been used in DL embedding layers, cover 33% of the entire records. The Part-of-Speech (POS) tagging (3%), topic modeling (3%), and sentiment (3%) features were the least used features. This suggests that the DL models and embedding features seem comparatively popular and widely used by the community.

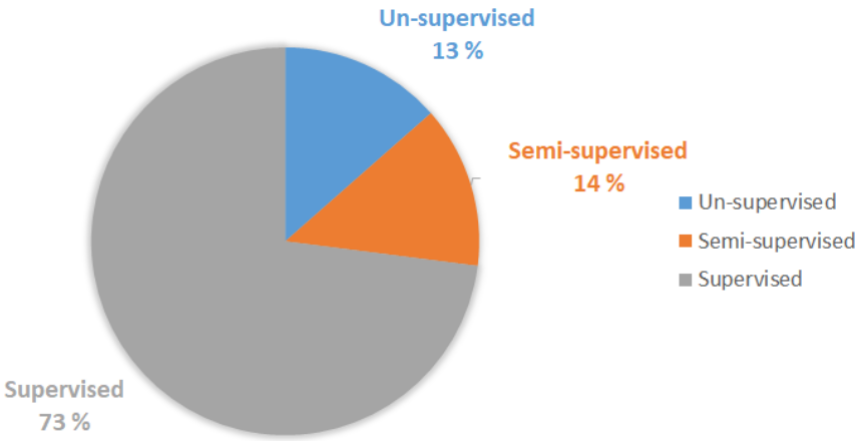

Figure 3.2: Statistics on the types of ML approaches used for HS detection (e.g., supervised, semi-supervised or unsupervised) [4].

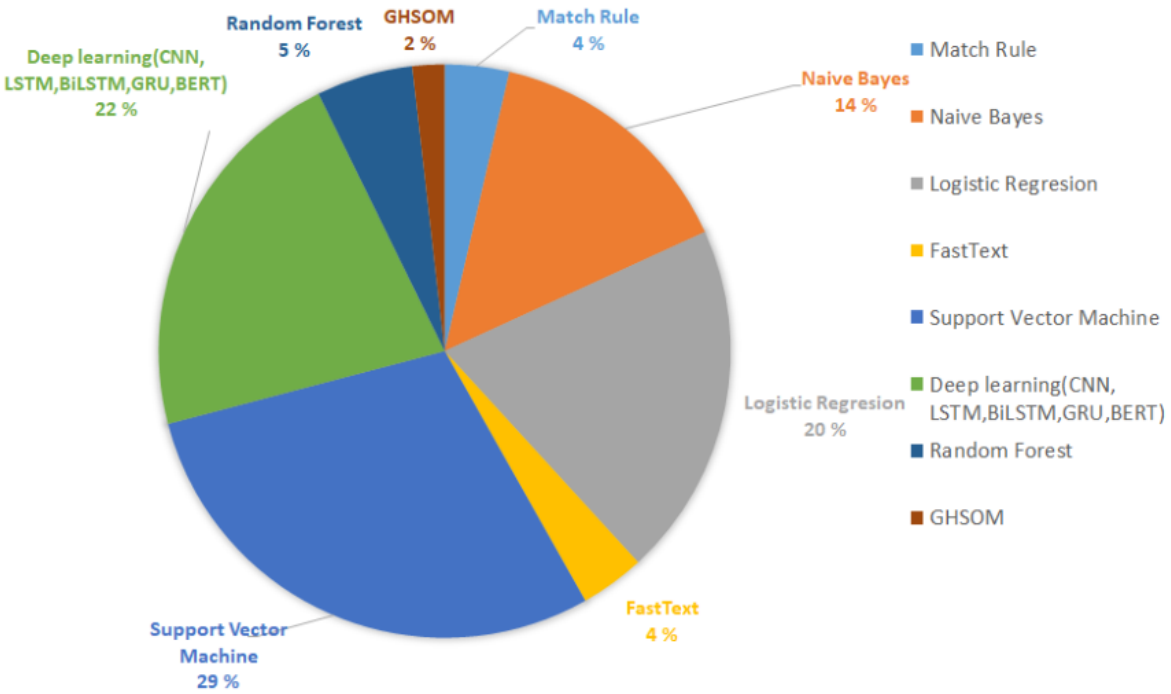

Figure 3.3: Statistics of algorithm types used for HS detection [4].

- The various DL algorithms were also analysed in terms of architecture and features employed. BERT (33%) is the most prevalent model, although it was only introduced recently, in 2019. The next most popular DL models are LSTM and CNN, which cover respectively 20% and 12% of total identified records. Most of the architectures used two steps: (i) word embedding layer employing models such as Word2Vec, FastText, GloVe; (ii) DL layer, where one distinguishes, among others, CNN, LSTM, and Gated Recurrent Unit (GRU) architectures.

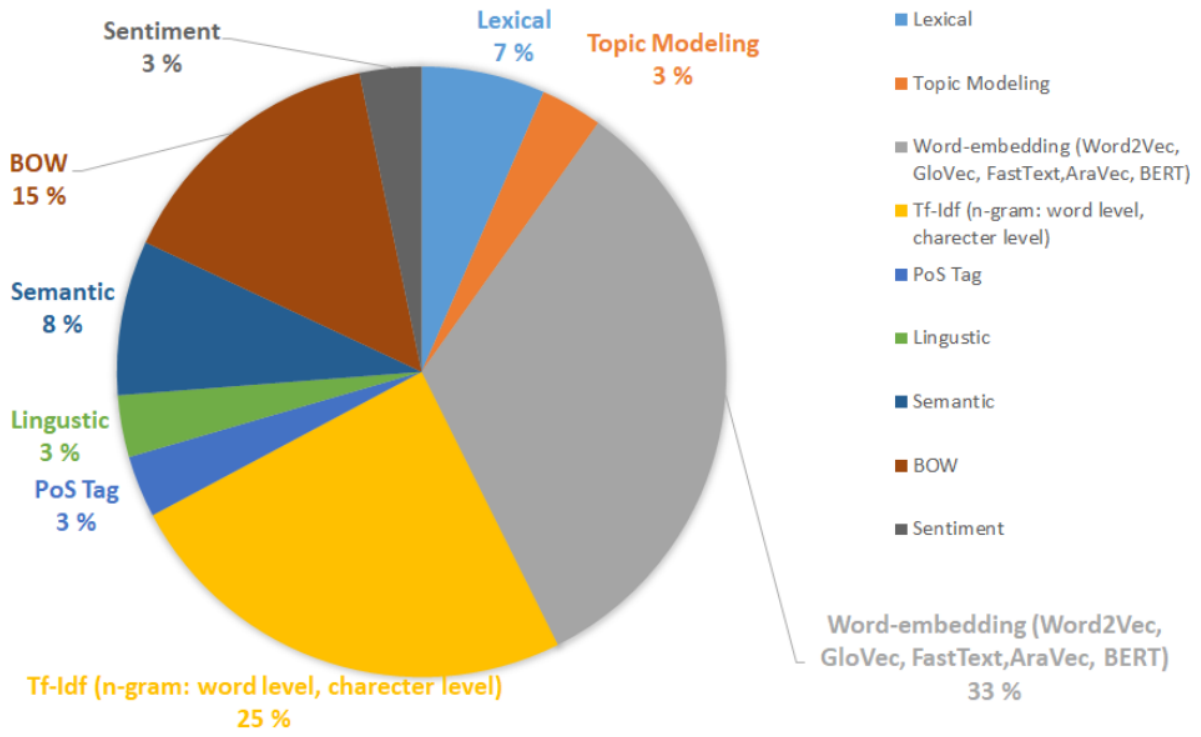

Figure 3.4: Statistics of features employed by ML/DL algorithms [4].

- DL models outperformed popular classifiers (NB, LR, RF, SVM) in most studies.
- Comparison between CNN, LSTM, Bidirectional Long Short-Term Memory (BiLSTM), and GRU models revealed mixed results in terms of which DL architecture performs best. Besides, the difficulty in data full reconstruction, the difficulty to reproduce exact preprocessing stages or use of distinct embeddings can make such comparison more challenging. One paper found that RNN (with GRU and LSTM layers) is more suited for the long-ranged context dependencies, while CNN seems better in extracting local features. They also revealed that GRU performs better in case of long sentences. However, several other papers suggested that the concatenation of two or more DL models achieve a better performance than a single DL model.
- Comparison between word embeddings revealed a lack of studies in this area. Word2Vec and FastText were regularly used with different DL architectures but this popularity does not entail systematically a better performance score. For instance, one study compared Word2Vec, GloVe, and Google NewsVec, and showed that Word2Vec performed only 1% better than the others. However, when compared to BERT model, BERT-Large showed the best Accuracy and F1-scores. Another work comparing FastText, GloVe, and GoogleNewsVec revealed no significant differences in the Accuracy scores between these embeddings. On the other hand, since Word2Vec, FastText, and GloVe vector representations capture semantic or meaning-related relationships as well as

syntactic or grammar-based relationships, this also bears an inherent limitation in the sense that this cannot capture polysemy relationship. That is, for the same word, even if it has different meanings in different contexts, the corresponding vector representation is unchanged. In that sense, Embeddings from Language Models (ELMO) word embedding model was designed to overcome the aforementioned shortcoming. However, in-depth comparisons of ELMO with other embedding models were not done yet. This leaves room for future experiments.

- The rise of BERT is a clear trend. This pre-trained model, when used as a classifier, is normally fine-tuned to the HS detection task. This Transfer Learning approach can be done in several ways and a popular one is by simply adding a classification layer on top of BERT [22]. Several works explored fine-tuning BERT in HS detection and almost all authors who compared a fine-tuned BERT model to other DL models concluded on the better performance of the BERT architecture. BERT model also achieved top performance in multilingual tasks. Some other language-specific BERT models developed over time for monolingual outperformed multilingual model Multilingual BERT (mBERT) but not every model has yet been tested for the HS domain.

3.2 Classical Methods

A popular approach for the detection of HS is the combination of feature extraction and classical ML algorithms, such as SVM, LR, and NB. Kwok et al. [37] implemented a NB classifier to distinguish between racist and non-racist tweets. They employed unigrams features (BOW) after finding that 86% of the time the reason a tweet was categorized as racist was because it contained offensive words. However, their model was insufficient to accurately classify anti-black speech, since the presence of offensive words led to the misclassification of tweets. Other studies applied a more sophisticated feature engineering. Warner et al. [38] detected HS, more specifically, anti-semitic content, using a SVM classifier, trained on word n-grams, brown clusters (grouping words into clusters that are assumed to be semantically related) and "the occurrence of words in a 10 word window". Surprisingly, unigram feature sets outperformed the full set, with the smallest feature set, comprised of only anti-semitic unigrams, performing the best. Chen et al. [39] used SVM with features including n-grams, automatically derived blacklists, manually developed regular expressions, and dependency parsing features to detect offensive language in YouTube comments. These dependency relationships have the potential benefit that non-consecutive words bearing a relationship can be captured in one feature. By not only using lexical features to detect offensive languages, but also incorporating style features, structure features and context-specific features they improved the traditional ML methods in terms of a much-reduced false negative rate. Burnap et al. [40] developed a supervised ML classifier for hateful and antagonistic content in Twitter. The authors also included syntactic features using typed dependencies as ML features

and reduced the false negatives by 7% over the baseline BOW features. Agarwal et al. [41] proposed several linguistic features such as presence of war, religious, negative emotions, and offensive terms to discriminate tweets promoting hate and extremism from other tweets. They employed single-class SVM and K-Nearest Neighbors (KNN) models for one-class classification task. Experimental results on a large and real-world dataset demonstrated that the proposed approach is effective with F-scores of 0.60 and 0.83 for the KNN and SVM classifier, respectively. Djuric et al. [25] proposed a two-step method for HS detection. First, they used paragraph2vec [42] for joint modeling comments and words, where they learn their distributed representations in a joint space using the continuous BOW (CBOW) neural language model.[1] Then, they used the embeddings to train a binary classifier to distinguish between hateful and clean comments. They pre-processed the text by lowercasing and removing stopwords and special characters, learned vector representations, trained the LR classifier and obtained a higher AUC than BOW models, while requiring less memory and training time to learn very effective HS detectors. Waseem et al. [43] analysed the impact of various extra-linguistic features in conjunction with character n-grams for HS detection using a LR classifier. They found that having the gender of the author as a feature can help improve HS classification but this information is often unavailable or unreliable on social media. Davidson et al. [26] trained a model to differentiate tweets between the following categories: HS, offensive language, or neither. They lowercased and stemmed each tweet and then used as features bigrams, unigrams, and trigrams, each weighted by its TF-IDF, POS tag unigrams, bigrams and trigrams, sentiment scores based on a sentiment lexicon and binary and count indicators for hashtags, mentions, retweets, and urls, as well as features for the number of characters, words, and syllables in each tweet. After testing different models, they found that the LR and SVM usually perform significantly better than the others. Their models misclassified some offensive language as HS but they were able to avoid the vast majority of these errors. They showed promising results but many of the tweets considered most hateful contained multiple slurs, meaning that their model will likely misclassify HS if it does not contain any curse words or offensive terms. Watanabe et al. [30] proposed an approach to classify tweets into hateful, offensive, and clean. Their approach was based on patterns, and unigrams along with sentiment and semantic features with SVM, Decision Trees (DT), and RF as classifiers. They showed that unigram features as well as the pattern features present the highest Accuracy, whereas semantic and sentiment features did not produce a good classification Accuracy. The combination of all features achieved an Accuracy of 87% for binary classification. However, their method depended on replacing hateful content with a specific pattern and unigram features, which led to the possibility of overlooking implicit hateful content with no clearly hateful pattern. Nobata et al. [29] addressed the detection of abusive language. The proposed model outperformed state-of-art models including a DL one. For building the model they developed a corpus for abusive language labels and applied a supervised approach using lexicon, lin-

[1] Although the feature extraction of this work makes use of DL technology we considered it a classical approach since the classifier was a classical one. The same applies to other works.

guistic, n-grams, syntactic, Word2Vec features and a regression model.

Malmasi et al. [44] developed methods to detect HS in social media, while distinguishing this from general profanity. They used supervised classification and a linear SVM with character n-grams, word n-grams, and word skip-grams as features and showed promising results with character 4-grams. Their results showed that distinguishing profanity from HS is a very difficult task. Similarly, Malmasi et al. [45] approached the problem of distinguishing general profanity from HS in social media. They employed SVM and ensemble classifiers along with a set of features that includes n-grams, skip-grams, and clustering-based word representations. Their analysis revealed that discriminating HS and profanity is not a simple task, which may require features that capture a deeper understanding of the text not always possible with surface n-grams.

Following a semi-supervised learning approach due to insufficient training instances and uncertain and imbalance feature distribution, Nahar et al. [46] applied a fuzzy SVM algorithm for cyberbullying detection. They implemented a number of lexical features (e.g., the number of swear words and capitalized words), sentiment features, and features based on metadata (e.g., the user's age and gender) and reported an F-score of 47%. Di Capua et al. [47] proposed an unsupervised approach to detect cyberbullying. Their model was based on a hybrid set of features: syntactic, semantic, sentiment and social features. They used the growing hierarchical self-organizing map (GHSOM) algorithm on different datasets taken from literature and achieved an F1-score of 0.4 on Twitter, an F1-score of 0.71 on FormSpring, and an F1-score of 0.74 on YouTube.

Gitari et al. [48] extracted sentences from sites that are considered to be generally offensive in United States. They annotated each of the sentences into one of three classes: strongly hateful, weakly hateful, and non-hateful. They developed a rule-based classifier using semantic features and grammatical patterns features and obtained an F1-score equal to 65.12%.

Wiegand et al. [49] addressed the detection of profane words by taking advantage of corpora and lexical resources. They used several features and general-purpose lexical resource to build their lexicon. While a lexicon of abusive words can only aid the detection of explicit abuse, its effectiveness was demonstrated on a task of cross-domain detection of abusive microposts, where their domain-independent lexicon outperformed previous supervised classifiers which suffer from overfitting to domain-specific features.

3.3 Deep Learning Methods

In recent years, DL methods have become more popular for both feature extraction and training classifiers.

Badjatiya et al. [18] performed the first experiments with DL architectures for the HS detection task,

to the best of their knowledge. They attempted to detect racism and sexism based on various DL architectures. As baselines, they tested char n-grams, TF-IDF, and BOW representations as features with different classifiers such as LR, RF, SVM, and Gradient Boosted Decision Trees (GBDT). The used DL architectures were the following: CNN, LSTM (to capture long-range dependencies in tweets, which may play a role in HS detection), and FastText (allows update of word vectors through back-propagation) initialized with either random embeddings or GloVe embeddings. All of these networks were fine-tuned using labeled data with back-propagation. Since these methods also learned task-specific word embeddings tuned towards the HS labels they also experiment by using these embeddings as features and various other classifiers like SVM and GBDT as the learning method. They found the Deep Neural Network (DNN) models to significantly outperform the baseline methods and the embeddings learned from the LSTM classifier initialized with random embeddings combined with GBDTs led to the best Accuracy values. Gambäck et al. [19] also used DL models to address HS on Twitter. Specifically, they proceed with CNNs and feature embeddings, such as one-hot encoded character n-gram vectors and word embeddings (Word2Vec). Although the character n-grams helped a little in improving Precision, the Word2Vec model without character n-grams achieved the best results of all the compared models. They outperformed the baseline LR in terms of Precision and F1-score, but not on Recall. Kamble et al. [20] explored HS detection in Hindi-English code-mixed tweets. They developed domain specific word embeddings from 255,309 Hindi-English tweets having hate and non-hate content which were collected using the Twitter API. They used the Word2Vec algorithm to train the word embeddings model. Using these embeddings as features, they conducted classification experiments with DL algorithms such as one-dimensional CNN (CNN-1D), LSTM, and BiLSTM. Among the three, CNN-1D resulted in the highest Precision, F1-score, and Accuracy, while BiLSTM gave the best Recall. Their models were able to better capture the semantics of HS along with their context which resulted in an improvement of about 12% in F1-score over a past work that used statistical classifiers. For the detection of cyberbullying incidents in social media platforms, Dadvar et al. [50] evaluated the performance of CNN, LSTM, BiLSTM, and BiLSTM with Attention as well as several Transfer Learning approaches. The best results were achieved with BiLSTM with Attention and Model Level Transfer Learning. They also mentioned that models like these can benefit from the integration of other sources of information. Sigurbergsson et al. [51] approached offensive language detection and developed four automatic classification systems, each designed to work for both the English and the Danish language. They experimented a LR classifier as a baseline, BiLSTM with random embeddings, BiLSTM with FastText embeddings, and BiLSTM with FastText embeddings and auxiliary features (n-grams, sentiment scores, linguistic features...). For English, their best model was BiLSTM with FastText embeddings, while for Danish LR worked best, maybe due to the low amount of data in the Danish dataset.

Park et al. [52] explored a two-step approach of performing classification on abusive language and

then classifying into specific types, and compared it with one-step approach of doing one multiclass classification for detecting sexism and racism on public English Twitter corpora. The authors proposed three models: a CNN with characters as input features (CharCNN), a CNN with Word2Vec embeddings as input features (WordCNN), and a CNN with both types of input features (HybridCNN). As baseline, they used character n-gram LR, SVM, and FastText [53] classifiers. In the one-step classification HybridCNN performed the best, giving an improvement over the result from WordCNN. They showed the potential in the two-step approach: combining two logistic regression classifiers in the two-step approach performed about as well as the one-step HybridCNN and using HybridCNN on the first step to detect abusive language and logistic regression on the second step worked better than just using HybridCNN.

Some authors have opted for an ensemble approach, a technique used for improving the performance of a single model. Pitsilis et al. [54] focused on classifying tweets as racist, sexist, or neutral. They modeled the tweets using word-based frequency vectorization, used additional features concerned with the users' tendency towards hatred behavior, and employed an ensemble of LSTM-based classifiers. The best results were achieved when both text and features related to user's behavior were used, the performance was improved by using an ensemble instead of a single classifier and their approach outperformed the state-of-the-art approaches. Zimmerman et al. [55] also proposed an ensemble of DL models for HS detection and also Sentiments Analysis of tweets. The authors ensembled 10 CNN models by summing softmax results from the underlying models and then averaging it. Considering the average softmax score of all models, the class with highest average was assigned to the given tweet. All tweets were tokenized and all urls, mentions, and numbers were normalized to URL, MENTION, and NUMBER, respectively. Utilizing an embedding model, the classifier was evaluated on two datasets, namely, abusive speech and SemEval 2013, obtaining average F1-scores of 77.83 and 70.36, respectively. They concluded that with 99% confidence, their ensemble approach will significantly improve F1-scores 98% of the time compared to results from a single model.

Hybrid architectures started to be applied to take advantage of multiple models. Zhang et al. [56] used a CNN with GRU network, combined with word embeddings, to detect HS on Twitter. They used a GRU instead of an LSTM because GRU is faster to train, generalises better on small data and achieves comparable results to LSTM. As baselines they used SVM with two different feature sets (the first includes surface, linguistic, and sentiment features and the second is an extension of the first with additional surface based features), a modification of their CNN+GRU model with dropout, global max pooling layers and elastic net regularisation removed, and a CNN. These last two baselines allowed to evaluate the impact of the respective modifications. For the preprocessing, the authors removed some punctuation characters, applied lowercase, stemming, and removed any tokens with a document frequency less than five. Further, they normalized hashtags into words because hashtags are often used to compose sentences. The proposed method was tested on various datasets, in order to either discriminate among

racism, sexism, and neither (or both), or between hate and non-hate tweets. Compared to existing methods, the CNN+GRU model proposed by the authors achieved the highest F1-score in all datasets and outperformed state of the art in most cases. Similarly, Van Huynh et al. [57] presented three different models to solve the problem at the Vietnamese Language and Speech Processing (VLSP) HS Detection on Social Networks 2019 Shared Task: a CNN, a Bidirectional Gated Recurrent Unit CNN, and a BiGRU-BiLSTM-CNN (BiGRU corresponds to a Bidirectional GRU). Regarding the word embeddings, they used FastText. Results of this task showed that BiGRU-BiLSTM-CNN achieved the best performance among these models with an F1-score of 0.705. Kapil et al. [58] proposed the use of CNN, LSTM, BiLSTM, Character-CNN, and some combinations among them with various word embeddings, namely Word2Vec, GloVe, and FastText for detecting several types of HS. LSTM and BiLSTM models performed best for all the datasets used and the addition of a Character-CNN layer improved the overall Accuracy and F1-score. Rosa et al. [59], to detect cyberbullying, also experimented hybrid architectures, particularly the following: CNN, CNN-LSTM, and CNN-LSTM-DNN. They used the Formspring dataset and three word embeddings trained using Google-News, Twitter, and Formspring. The experimental results showed that these models outperformed SVM and LR models, with the best result being an F1-score of 0.444 for the class that contains offensive content. For offensive language detection, Ong [21] experimented CNN, LSTM, BiLSTM, GRU, BiGRU, and combinations among these models. The author used GloVe word vectors, some pre-trained using Twitter with 100 and 200 dimensions, and others pre-trained with Common Crawl with 300 dimensions. Synthetic Minority Over-sampling TEchnique (SMOTE) and Class Weights were used to balance the data among classes. The author concluded that the architecture that gave the highest macro average F1-score was BiLSTM-CNN.

New architectures, namely BERT-based ones, started to outperform these hybrid architectures. Ranasinghe et al. [22] presented a multilingual DL model to identify HS and offensive language in social media, in their submission to the sub-task A of the HASOC 2019 shared task. To make the system portable to all languages in the dataset, they used only minimal preprocessing methods, such as removing usernames, removing urls, and, depending on the architecture, converting all tokens to lowercase. They experimented multiple DNNs architectures: pooled GRU, LSTM+GRU+attention, two-dimensional CNN (CNN-2D)+Pooling, GRU+capsule and LSTM+capsule+attention, using FastText as word embeddings. Furthermore, they also experimented fine-tuning BERT, which outperformed every above mentioned DNN for all three languages (German, English and Hindi). Mozafari et al. [5] used two Twitter datasets that were annotated for racism, sexism, hate, and offensive content and experimented different combinations of BERT with other models, such as CNN and LSTM. The evaluation results indicated that BERT-CNN outperformed previous works by profiting from the syntactical and contextual information embedded in different transformer encoder layers of the BERT using a CNN-based fine-tuning strategy. Kovács et al. [60] tackled the automatic detection of HS in social media data also using

the HASOC 2019 dataset. For the text preprocessing they removed extra spaces, replaced words that start with @ and urls with @USER, URL, respectively, and removed hash characters and emoticons. They found that emojis (including facial emojis) were not correlated with hatefulness/offensiveness tweet scores and based on that they also removed all emojis. First, they used a CNN-LSTM model and then they conducted experiments with A Robustly Optimized BERT Pretraining Approach (RoBERTa). They also experimented with RoBERTa as a feature extractor for classical ML methods (KNN, AdaBoost, linear discriminant, LR, RF and SVM) and carried out experiments using the FastText classification model. Using a combination of RoBERTa and FastText they have attained results that were state-of-the-art in the sub-task A of the HASOC 2019 competition.

Several shared tasks have been focusing on the detection of HS related concepts. OffensEval shared tasks conducted in the scope of SemEval[2] focused in the detection of offensive language. In OffensEval 2019, Zampieri et al. [23] mentioned that among the top-10 teams, seven used BERT with variations in the parameters and in the preprocessing steps. The top-performing team [61] achieved a macro average F1-score of 0.829, using pre-trained BERT with finetuning on the OLID dataset, and hashtag segmentation and emoji substitution as preprocessing. In OffensEval 2020, it was emphasized in [62] that many teams used context-independent embeddings from Word2Vec or GloVe and some works resorted to other transfer learning approaches or multitask learning. In terms of models, the ones used were BERT, mBERT, RoBERTa, Cross-Lingual Language Robustly Optimized BERT Pretraining Approach (XLM-RoBERTa), A LiteBidirectional Encoder Representations from Transformers (ALBERT) and GPT-2. Most of the teams performed some of the following preprocessing: conversion of emojis into word representations, hashtag segmentation, abbreviation expansion, bad word replacement, spell correction, lowercase conversion, stemming, and lemmatization. Other techniques included the removal of user mentions, urls, hashtags, emojis, e-mails, dates, numbers, punctuation, consecutive character repetitions, offensive words, and stop words. The best team had a macro average F1-score of 0.92, using an ensemble of ALBERT models of different sizes and the OLID to train. HASOC competitions[3] also focused in the identification of HS and offensive content. Over 40 research groups participated in HASOC 2020 [63]. Most participants used DL models and in particular transformer-based architectures were popular. The best submission for Hindi-HS detection, used a CNN with FastText embeddings as input [64]. The best performance for German HS detection task was achieved using fine-tuned versions of BERT, DistilBERT, and RoBERTa [65]. The top performance in English language HS detection was based on a LSTM architecture with GloVe embeddings as input [66]. HaSpeeDe is also a HS detection shared task and it is organized within EVALITA.[4] Lavergne et al. [67] obtained the best macro F1-scores on the in-domain test set for both HS and stereotype detection of the HaSpeeDe 2 shared task organised

[2] https://semeval.github.io/
[3] https://hasocfire.github.io/
[4] https://www.evalita.it/

within EVALITA2020. The chosen classification approach was to fine-tune a BERT-based language model. They decided to add a simple linear layer with a softmax on top of it, for simplicity and because it is efficient enough since the other layers are fine-tuned. They evaluated two multilingual models, mBERT and XLM-RoBERTa, and three Italian monolingual models, AlBERTo, UmBERTo, and PoliBERT. The usage of multitask learning between the HS detection and stereotype detection was also evaluated. They removed emoticons and hashtags and replaced urls and user names with associated tags as done in the evaluation data. They used as baselines developed systems, such as TF-IDF bag of words and a BiLSTM with trainable word vectors inputs. Multilingual models performed worse than monolingual models even when additional data was used to train them, although they achieved results better than the baseline models. The system with multitasking learning performed much better on the in-domain data for the HS detection task. This may be because the multitask model has learned to discard some data that do not have the characteristics of stereotypes and are therefore unlikely to contain HS.

4

Data

Contents

4.1 Annotation Guidelines 35
4.2 CO-HATE Corpus 37
4.3 FIGHT Corpus 40

In this chapter we describe the datasets that will support our experiments, namely for training, validating, and testing our models. In detail, we will use two Portuguese datasets that were developed in the scope of the project *HATE COVID-19.PT - Detecting Overt and Covert Hate Speech in Social Media*[1]: CO-HATE corpus (Section 4.2) and FIGHT corpus (Section 4.3). Both annotated corpora focus on the expression of OHS within the Portuguese context on three specific target groups, namely the Afro-descendant, Roma, and LGBTQI+ communities. These communities were chosen since they are among the most commonly reported targets of both offline and online HS in Portugal [68].

4.1 Annotation Guidelines

In order to perform the annotation, the senior members of the team created and discussed guidelines. The annotation process of both datasets considers four dimensions of analysis and the categories and subcategories assigned to each dimension are represented in Table 4.1.

Table 4.1: Dimensions and attributes of CO-Hate and FIGHT corpora annotation scheme.

Discourse Type	Target Group	Rhetorical Strategy	Sentiment Intensity
Explicit Hate Speech	Afro-descendants	Fear appeal	Very Negative
Implicit Hate Speech	Roma	Call to action	Negative
Offensive Speech	LGBTQI+	Personal attack	Neutral
Counterspeech	Racism	Stereotype	Positive
	Xenophobia	Irony/Sarcasm/Humor	Very Positive
	Other	Rhetorical Question	
		Other	

Concerning the discourse type dimension it is relevant to distinguish HS from Offensive Speech (Example 3), Counterspeech (Example 4) and also identify the type of HS, Direct (Example 1) or Indirect (Example 2). The definitions underlying these concepts are highlighted in Table 4.2.

1. *Que se f@da o racismo! Se não fossem esses parasitas da sociedade que não querem fazer nada, Portugal era um paraíso.*
   F@ck the racism! If it were not those social parasites that don't want to do anything, Portugal was a paradise.

2. *Pura verdade. E se for ver os exemplos de países mais evoluídos como Holanda e França, já nem ciganos lá existem. Foram corridos de lá para fora.*
   Pure truth. And if you look at the examples of more developed countries like the Netherlands and France, there aren't even Roma there anymore. They were kicked out.

[1] https://hate-covid.inesc-id.pt/

Table 4.2: Definitions of the discourse type concepts adopted in the annotation process of CO-Hate and FIGHT corpora.

Concept	Definition
Hate Speech (HS)	Any act of communication that incites, attacks or supports hatred against a person or group of people belonging to a vulnerable or marginalized group, diminishing or discriminating it, based on one or more specific characteristics of that group (for example, race, ethnicity, nationality, religion, gender or sexual orientation).
Direct HS	Hate speech is expressed directly or explicitly, often using derogatory or offensive words or expressions against the targeted group.
Indirect HS	Indirect or covert attack on an individual or target group to which that individual belongs, often using rhetorical figures (eg, irony, sarcasm, humor, analogy, comparison, metaphor, rhetorical question, etc.)
Offensive Speech	Aggressive, offensive, insulting, prejudiced or discriminatory speech, but not targeting a vulnerable or marginalized person or group.
Counterspeech	Any direct response to Hate Speech with the aim of fighting it.

3. *É tudo a mesma bosta, todos esses vermes são racistas e xenofóbicos.*
   It's all the same crap, all these worms are racist and xenophobic.

4. *Nao, nao é racismo. Só é racismo se o indivíduo branco for agredido ou maltratado por causa da cor da sua pele. No caso aqui relatado, é sobre a constante actuacao abusiva da polícia contra cidadaos negros.*
   No, it is not racism. It is only racism if the white individual is attacked or mistreated because of the color of his skin. The case reported here is about the constant abusive action of the police against black citizens.

In case of not meeting any of the described discourses types, the comment is considered as *Non Relevant*. Besides the three previously mentioned targets, the annotation also includes racism and xenophobia to represent a more generic target. The sentiment and the rhetorical strategies used in the comments were also annotated. With the exception of sentiment, the remainder categories are not mutually exclusive. This means that annotators can assign as many labels as considered relevant. As a consequence, the number of labels per comment may differ.

4.2 CO-HATE Corpus

In order to create the CO-HATE corpus [12], the project team started by selecting a set of YouTube videos whose topic could potentially generate polarized content and hatred against the Afro-descendant, Roma, and LGBTQI+ communities. Specifically, the selection was based on a set of words and expressions typically used to mention the targets considered combined with controversial topics or events. Some titles of the extracted videos are illustrated in Examples 1, 2 and 3:

1. *Violência policial racista em Portugal*
   Racist police violence in Portugal

2. *A comunidade cigana vive numa bolha de impunidade*
   The Roma community lives in a bubble of impunity

3. *Preferias ter um filho homossexual ou ladrão? Experiência Social*
   Would you rather have a son that was homosexual than a thief? A Social Experiment

This selection was restricted to videos posted by Portuguese authors (making a total of 39 videos) since we are particularly interested in analyzing this phenomenon within the Portuguese context. The comments associated with the selected videos were not filtered in order to both assess, for each video, the real distribution of HS, and investigate other related phenomena, in particular, Counterspeech and Offensive Speech.

This corpus is composed of 20,590 written messages (795,111 tokens), posted by 8,485 different online users. The average number of comments per video is 528, and the number of comments associated with the selected videos ranges from 116 to 1,708. The distribution of the comments according to the target groups is represented in Table 4.3. The community most represented is the Afro-descendant community corresponding to 40% of the retrieved comments, followed by the LGBTQI+ community (31%), and lastly, the Roma community with 28% of the comments.

Table 4.3: Distribution of CO-HATE corpus according to the mentioned target.

Target	**Number of messages**
Afro-descendants	8,278 (40%)
Roma	5,862 (28%)
LGBTQI+	6,450 (31%)
Total	**20,590**

The corpus was then subdivided into five subsets, each containing approximately 4,000 messages, relative to seven YouTube videos, in average. Each subset was randomly assigned to a different annotator. Additionally, all the annotators were assigned to a common subset comprehending 534 messages,

relative to two additional videos. This subset was used to measure the agreement among the annotators and also used as the test set in our experiments. For the purpose of understanding the entire context of each comment analyzed, the annotators were asked to first watch the video, and then read carefully each comment, following the messages' order presented in the corpus.

4.2.1 Annotators Profile

The corpus annotation was performed by five recruited annotators, who were enrolled in a bachelor's or a master's degree in Communication or in Political and Social Sciences. The average age of the annotators is 23.6, and three annotators are female. The annotation team is composed by both individuals belonging to the communities monitored in this study (A, B, and C), and by annotators that do not belong to any potentially minority group (D and E). More specifically, the annotation team includes Portuguese youth as follows: a female of African descent, a White male who identifies himself as part of the LGBTQI+ community, a female of Roma descent, a White cisgender hetero male, and a White cisgender heterofemale.

The team has intentionally included both individuals belonging to the target groups monitored in our research and individuals that do not belong to any potential marginalized group in order to create a resource representing the multiplicity of perspectives of human subjects, particularly the ones directly involved in our study.

4.2.2 Annotation Results

Table 4.4 presents the distribution on the training set of messages classified as conveying HS by each annotator individually, and by all the annotators. About 35% of the comments were classified as *HS*, being *Implicit HS* more frequent than *Explicit HS*. Around 23% of the comments were annotated as *Offensive Speech* and 17% as *Counterspeech*.

Table 4.4: Proportion of messages classified as Hate Speech in CO-HATE training set for each annotator and for all annotators.

Annotators	**Number of messages**	**HS** (%)
A	4,008	25
B	4,011	36
C	4,017	29
D	4,014	39
E	4,006	48
Total	**20,590**	**35**

The IAA was measured using Krippendorff's alpha [69]. In order to assess whether HS is perceived

differently by individuals with different social identities, we calculated and present in Table 4.5 the IAA for (i) all the annotators (ALL), (ii) annotators belonging to the target groups (A-B-C), and (iii) annotators that do not belong to those groups (D-E). We also present in Table 4.6 the pairwise IAA on the HS class. The IAA between all the annotators for the classification of HS was considerably low (0.478), despite providing the annotators with detailed guidelines. This is also verified for the remaining attributes, demonstrating the subjectivity and difficulty of this task. As expected, Indirect HS is harder to classify than Direct HS. Besides, the Offensive Speech seems to be even harder to identify, especially between annotators A, B, and C, due to its similarity with HS. Directly comparing the two groups, the one composed by the annotators not belonging to the communities targeted reached a good agreement in the majority of the attributes and a higher agreement than the group composed by the annotators A, B, and C. This may reflect the idea that the HS detection highly depends on the personal perception and can be affected by a number of variables, including the individual's social identity.

Table 4.5: IAA of all discourse types for different groups of annotators on CO-Hate corpus.

Attribute	All	ABC	DE
Hate Speech	0.478	0.360	0.735
Explicit Hate Speech	0.416	0.383	0.548
Implict Hate Speech	0.237	0.145	0.421
Offensive Speech	0.143	0.005	0.472
Counterspeech	0.419	0.358	0.762

Table 4.6: Pairwise IAA of the Hate Speech class on CO-HATE corpus.

	A	B	C	D	E
A	1	0.57	0.221	0.659	0.643
B	0.57	1	0.262	0.609	0.531
C	0.221	0.262	1	0.235	0.211
D	0.659	0.609	0.235	1	0.735
E	0.643	0.531	0.211	0.735	1

Given the task subjectivity, and assuming that the profile of human annotators may influence the data annotation, the final labels for the golden set messages were obtained considering a class true if it was labelled as such by at least two annotators. With this voting strategy, the test set is composed of 50% HS messages. We did not consider the messages containing a unique positive vote in order to discard unintentional errors introduced by the annotator; the possibility of two annotators making a mistake would be a more unlikely scenario.

4.3 FIGHT Corpus

The FIGHT corpus [70] was obtained by collecting data from the Twitter API and also from an existing database composed of tweets that have been extracted daily since 2015. By combining both sources of information, we obtained an updated and robust dataset, which includes the information currently available on Twitter, and information on tweets previously collected that are no longer available, because they were deleted or the Twitter account where they were posted was removed. The data selection was restricted to a time span of about 3 years, from January 1, 2018 to November 31, 2021 and only the tweets posted by the Portuguese community were retrieved. We have created a **target lexicon** composed of 259 words and expressions often used to mention the targets we are interested in monitoring, particularly the African descent, Roma, and LGBTQI+ communities. This lexicon includes the unambiguous forms associated with each semantic category, corresponding to a total of 174 entries (e.g. *Africans*) and the ambiguous forms, such as *preto* ('black'), which can be used in a variety of contexts with a different meaning (e.g. *Eu adoro esse casaco preto*, 'I love that black coat'). In addition, we have created an **offensive lexicon** including approximately 800 inflected forms that are often used to insult or offend the previously mentioned targets. FIGHT-Target is composed of 63,931 tweets containing unambiguous terms described in the **target lexicon** (3,951 related to Roma, 34,111 to LGBTQI+, and 25,869 to Afro-descendants). FIGHT-Offensive is composed of 11,214 tweets containing terms from both the **target** (including both ambiguous and unambiguous forms) and the **offensive lexicons** (435 related to Roma, 3,190 to LGBTQI+, and 7,589 to Afro-descendants). This selection allowed to retrieve potentially offensive and hateful messages targeting each protected community (e.g. *É o **preto** mais **burro** que já vi mano*, 'It's the dumbest nigga I've ever seen bro').

The final collection was attained by first getting only the tweets that are no longer available from FIGHT-Target, suggesting they might contain hateful content. To these tweets we added the tweets of FIGHT-Offensive and the removed duplicate tweets. By following this process we were able to collect 19,148 different tweets, posted by 7,303 different users; from those tweets, 15,429 were extracted from the existing database and 3,719 were retrieved from the Twitter API. The distribution of the tweets according to the target groups is represented in Table 4.7. The most represented community is the Afro-descendant community corresponding to approximately 52% of the retrieved tweets, followed by the LGBTQI+ community (approximately 41%), and lastly, the Roma community with approximately 5% of the tweets. This collection was shuffled, 1,000 tweets were selected for the testing set and 18,148 tweets for the training set. The training set was subdivided into five subsets, and each subset was randomly assigned to a different annotator. Additionally, all the annotators were assigned to the test set.

Table 4.7: Distribution of the final FIGHT corpus according to the mentioned target.

Target	Number of Tweets
Only Afro-descendants	9,978 (52.11%)
Only Roma	1,003 (5.24%)
Only LGBTQI+	7,848 (40.99%)
Afro-descendants and LGBTQI+	254 (1.33%)
Afro-descendants and Roma	51(0.27%)
Roma and LGBTQI+	7 (0.04%)
Afro-descendants, Roma and LGBTQI+	7 (0.04%)
Total	**19,148**

4.3.1 Annotators Profile

Just like we did with CO-HATE we chose an inclusive annotation team, composed by: a female and a male who identify themselves as part of the LGBTQI+ community, a female of Roma descent, a White cisgender hetero male, and a White cisgender hetero female. We refer to the individuals belonging to the communities monitored in this study as B, D, and E annotators and to the annotators that do not belong to any potentially minority group as A and C annotators.

4.3.2 Annotation Results

Table 4.8 shows the percentage of messages classified as conveying HS on the training set by each annotator individually and all the annotators. About 29% of the tweets were classified as HS. Direct HS is more frequent (19%) than Indirect HS (9%), which is probably due to the retrieval method that consisted in a keyword approach based on potentially offensive words. Around 12% of the comments were annotated as *Offensive Speech* and 21% as *Counterspeech*.

Table 4.8: Proportion of messages classified as Hate Speech in FIGHT training set for each annotator and for all annotators.

Annotators	Number of messages	**HS** (%)
A	4,630	24
B	4,629	24
C	4,630	32
D	4,629	34
E	4,630	23
Total	**19,148**	**29**

In Table 4.9 we present the IAA regarding all discourse types for (i) all the recruited annotators (ALL),

(ii) the annotators belonging to the target groups (B-D-E), and (iii) the annotators that do not belong to those groups (A-C). We also present in Table 4.10 the pairwise IAA on the HS Class on FIGHT corpus. The IAA between all the annotators for the classification of HS was considerably low (0.362). For Indirect HS, the agreement is almost nonexistent. This may be due to the lack of context that makes it even harder to identify what could be Indirect HS. Directly comparing the two sub-groups, in this case, the group composed by the annotators belonging to the communities targeted (B, D, and E annotators) reached a higher agreement than the group composed by the annotators A and C.

Table 4.9: IAA of all discourse types for different groups of annotators on FIGHT corpus.

Attribute	**All**	**AC**	**BDE**
Hate Speech	0.362	0.189	0.439
Explicit Hate Speech	0.324	0.242	0.330
Implict Hate Speech	0.080	0.035	0.116
Offensive Speech	0.214	0.121	0.269
Counterspeech	0.268	0.138	0.312

Table 4.10: Pairwise IAA of the Hate Speech class on FIGHT corpus.

	A	B	C	D	E
A	1	0.196	0.189	0.167	0.137
B	0.196	1	0.538	0.512	0.415
C	0.189	0.538	1	0.594	0.433
D	0.167	0.512	0.594	1	0.381
E	0.137	0.415	0.433	0.381	1

For the same reasons presented in Section 4.2.2, we labelled the golden set using at least two votes type of voting. With this voting strategy, the test set is composed of 38% HS messages.

5

Modelling

Contents

5.1 BERT-LinearLayer 45

5.2 BERT-CNN 45

5.3 BERT-Attention 46

5.4 Pre-trained BERT Models 47

In this chapter, we describe the models we use for detecting OHS in Portuguese, based on the data previously described. We decided to test such models because they have already shown good performance in similar tasks [5, 6]. In Section 5.1 we describe the BERT-LinearLayer model, in Section 5.2 the BERT-CNN model, in Section 5.3 the BERT-Attention model and in Section 5.4 the pre-trained BERT models used.

5.1 BERT-LinearLayer

BERT-LinearLayer, which is illustrated in Figure 5.1, is inspired in [6] and [5] and it is the standard BERT model with an added single Linear Layer. In this architecture, only the [CLS] token output provided by BERT is used. The [CLS] token output of the 12^{th} transformer encoder, a vector of size 768, is given as input to a fully connected network without hidden layer. The Sigmoid activation function is applied to the hidden layer in order to make the prediction.

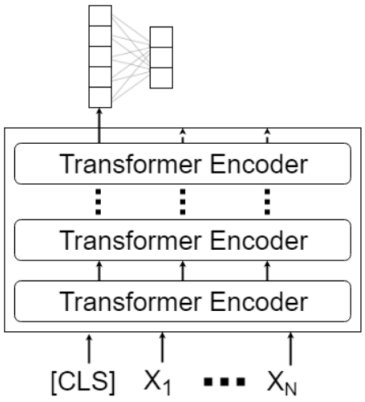

Figure 5.1: BERT-LinearLayer model structure [5].

5.2 BERT-CNN

BERT-CNN model is inspired in [6] and consists of two main components. The first one is the BERT model, in which the text is passed through 12 layers of self-attention to obtain contextualized vector representations. The other one is a CNN, which is used as a classifier.

First, the text is given as input to BERT, then the output of the last four hidden layers of the pre-trained BERT is concatenated to get vector representations as shown in Figure 5.2. Next, these embeddings are passed in parallel into 160 convolutional filters of five different sizes (768x1, 768x2, 768x3, 768x4, and 768x5), 32 filters for each size. Each kernel takes the output of the last four hidden layers of BERT as four different channels and applies the convolution operation on it. After that, the output is passed

through the ReLU Activation function and a Global Max-Pooling operation. Finally, the output of the pooling operation is concatenated and flattened to be later on passed through a dense layer and a Sigmoid function to get the final binary label.

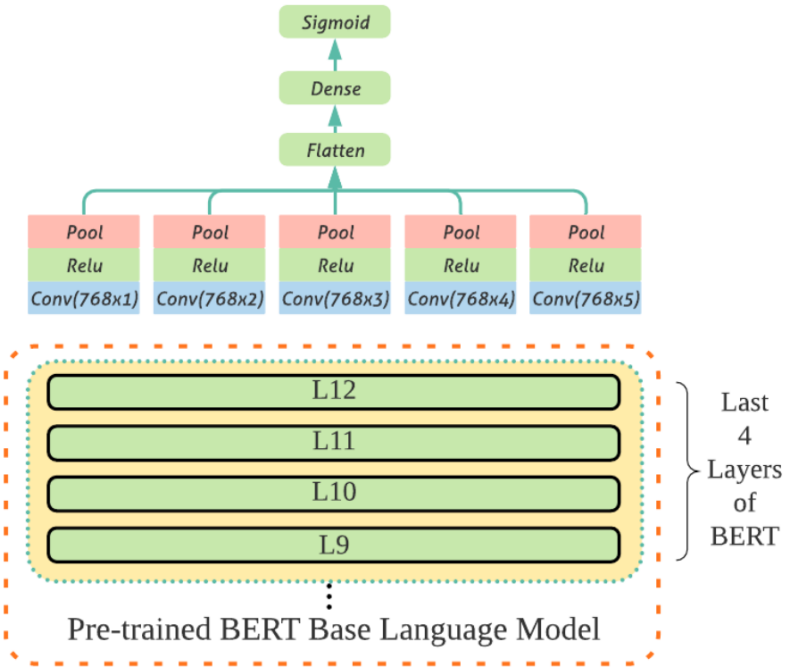

Figure 5.2: BERT-CNN model structure [6].

5.3 BERT-Attention

BERT-Attention model is inspired in [7], where the authors propose an architecture to detect misogyny and aggression using a multitask approach that, as shown in Figure 5.3, consists of the following modules: BERT Layer, Attention Layer, Fully-Connected Layers, and Classification Layer. First, the input sequence of tokens is passed to the BERT model to extract contextualized information. Then, the output of the BERT layer is fed to the Attention Layer. The output of the Attention Layer is passed to Fully Connected (linear) layers for dimension reduction. There are two linear layers with 500 and 100 neurons, respectively. Finally, the output of the Linear Layers is fed to two separate classification layers, each one for predicting a different class. For both cases, a Linear Layer is used with a Sigmoid activation on top, which gives a probability score to the classes. We will perform multitask experiments involving HS detection but we also developed a single-task version of this model, which will be helpful to measure the impact of learning two tasks at the same time on the results.

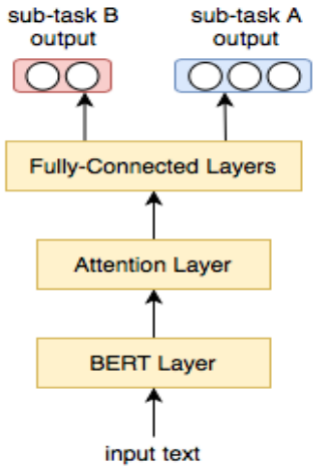

Figure 5.3: BERT-Attention model structure [7].

5.4 Pre-trained BERT Models

Every HS detection model we propose needs a pre-trained BERT model. Since we are dealing with Portuguese textual data we decided to use **BERTimbau** (brBERT) [71] trained with the brWac corpus [72], with a vocabulary size of 30k tokens and the usual training objectives – Masked Language Model (MLM) and Next-Sentence Prediction (NSP).

BERT achieves good performance on numerous NLP tasks, but when applied to less standard language varieties, such as social media data, results may fluctuate a lot. A competitive, effective, and fast solution to adapt pre-trained language models to new language varieties or domains is further pre-training a BERT-like model.

Inspired in [73], we retrained **BERTimbau** model by applying the Masked Language Model (MLM) objective. For this purpose we used 229,103 tweets, which are a combination of the tweets that were discarded from FIGHT-Target and all the tweets from the conversations associated with the FIGHT-Offensive tweets, making sure that no tweets from FIGHT corpus (Section 4.3) were used. We used the code they made available: `https://osf.io/tbd58/`. We retrained for 100 epochs in batches of 4 samples, including up to 512 sentence piece tokens and used Adam with learning rate 5e-5. We named this retrained **BERTimbau** model **HateBERTimbau**, a model that focuses on Portuguese social media language including potential HS and targeting African descent, Roma, and LGBTQI+ communities.

We also considered to use **Multilingual BERT** (mBERT),[1] but in previous experiments **BERTimbau** had an overall better performance than mBERT [74] so we discarded this option.

[1] `https://github.com/google-research/bert/blob/master/multilingual.md`

6

Results

Contents

6.1 Experimental Setup 51
6.2 Results for the CO-HATE Corpus 52
6.3 Results for the FIGHT Corpus 61
6.4 Summary and Discussion 70

In this chapter, we present and analyse the results from the experiments carried out in this dissertation. In Section 6.1 we present the experimental setup , in Section 6.2 we describe the experiments with CO-HATE corpus and in Section 6.3 the experiments with FIGHT corpus. The results are presented according to Macro Average and Positive Class being the main evaluation metrics reported Precision, Recall and F1-score. In all these experiments, the text was not pre-processed since in other experiments made with CO-HATE data, pre-processing did not improve the results [74] and we expect that linguistic clues like repetitions of punctuation signals and emojis could be helpful in capturing HS. The annotation of both CO-HATE and FIGHT corpus comprised other categories beyond HS. We take advantage of that in our multitask models where we not only detect HS but also another class. One of the categories that our annotation considered was sentiment intensity. HS detection and Sentiment Analysis are closely related NLP tasks, since HS messages usually convey a negative sentiment, and several approaches have been incorporating sentiment information to support HS classification [75]. In our annotation study, annotators were asked to classify each comment according to the following scale: 1 (very negative) - 2 (negative) - 3 (neutral) - 4 (positive) - 5 (very positive). We decided to adapt this scale to negative, positive and neutral classes and use this category in our multitask models. We also looked at the categories that achieved the best IAA in an attempt to use the least subjective one in our multitask models. For both CO-HATE and FIGHT corpora this category is Counterspeech (IAA of 0.419 in CO-HATE and IAA of 0.268 in FIGHT). As it was said in Chapter 5, BERT-Attention is the architecture that will be used to perform the multitask experiments: BERT-Attention-CS denotes the multitask model that identifies Counterspeech and HS, and BERT-Attention-SNT denotes the multitask model that identifies Sentiment Polarity and HS.

6.1 Experimental Setup

For all our experiments, we used the base version of BERT, which contains an encoder with 12 layers (transformer blocks), 12 self-attention heads, and 110 million parameters. The maximum sequence length of each text sample was set to 350 tokens to avoid overloading the GPU. A substantial amount of messages does not exceed that length and it does not degrade performance. The training data was split into 80% and 20% for training and development sets, respectively, preserving the same proportions of examples in each class. All the models were trained with Adam optimizer for 15 epochs and the model with the best Positive Class F1-score on the development set was selected.

We report both macro and Positive Class F1-scores, but when assessing the models' performance we give particular importance to the Positive Class F1-score, since it evaluates the models' performance on the class that we want to detect.

6.2 Results for the CO-HATE Corpus

In this section, the results of the detection of OHS using the CO-HATE corpus as training, development, and testing data are presented. This corpus was annotated by five different annotators and led to a low IAA (0.478), demonstrating the difficulty and subjectivity of this task, even for humans. Considering this heterogeneity of views on the HS concept, we have tested several combinations of data for the training of the models. Besides using the entire training data, i.e., all the messages annotated by annotators A, B, C, D, and E, we have also experimented using the data annotated by each user independently (A, B, C, D, and E). Since annotator C was the one having the worse IAA when compared to the remaining annotators (0.23 on average, while the others have at least 0.531, see Table 4.6), we tested the combination A+B+D+E. We decided to also test the combination A+B+C, composed by annotators that belong to the target communities and the combination D+E, composed by annotators who do not belong to any potential historically marginalized group. With these subsets of data, we may understand how the data agreement and annotators' profile may affect the performance of OHS detection.

6.2.1 Single-task Experiments

The results of all single-task models for the different sets of training data using **BERTimbau** are represented in Table 6.1. We present the results of our baseline, a dummy classifier that classifies all instances as HS. Considering the Positive Class F1-score as the benchmark metric, we can see that using the data of A, B, C and A+B+C always leads to worse results than the results obtained by the baseline model. Also, when using the data of all annotators (A+B+C+D+E), the best result (Positive Class F1-score of 0.672) was higher than the result of the baseline model but the difference was not that significant. On the other hand, when using the data of D, E, D+E, and A+B+D+E, the results outperformed the baseline. Analysing the models for each subset of data, their performance varies greatly: for A, D and A+B+C+D+E, BERT-LinearLayer achieved the best performance; for B and C and A+B+C+D+E, BERT-Attention achieved the best performance; and for E, D+E, A+B+D+E and A+B+C, BERT-CNN achieved the best performance. The best result was obtained using the data of D+E and the BERT-CNN model corresponding to a Positive Class F1-score of 0.709, surpassing the baseline by 6.3%.

We performed exactly the same experiences but using **HATEBERTimbau**, expecting that better results would be achieved since it is adapted to social media language that includes potential HS targeting African descent, Roma, and LGBTQI+ communities. The results of all single-task models for the different sets of training data using **HATEBERTimbau** are presented in Table 6.2. Now, using **HATEBERTimbau**, only with the data of B the performance is always inferior to the baseline model while all the other subsets of data can outperform the baseline. For every subset of data used are obtained better results than when using **BERTimbau**, suggesting that the domain-specific model obtained with the retraining obtains more

Table 6.1: Performance of the single-task models using **BERTimbau** for the different sets of training data in the test set of CO-HATE corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.5	1	0.667	0.25	0.5	0.333
A	BERT-CNN	0.59	0.689	0.636	0.608	0.605	0.602
	BERT-LinearLayer	0.592	0.708	0.645	0.615	0.61	0.607
	BERT-Attention	0.582	0.61	0.596	0.586	0.586	0.586
B	BERT-CNN	0.665	0.603	0.633	0.651	0.65	0.649
	BERT-LinearLayer	0.723	0.498	0.59	0.67	0.654	0.645
	BERT-Attention	0.574	0.708	0.634	0.597	0.592	0.586
C	BERT-CNN	0.586	0.584	0.585	0.586	0.586	0.586
	BERT-LinearLayer	0.577	0.629	0.602	0.585	0.584	0.583
	BERT-Attention	0.61	0.633	0.621	0.614	0.614	0.614
D	BERT-CNN	0.597	0.831	0.695	0.659	0.635	0.62
	BERT-LinearLayer	0.613	0.824	0.703	0.672	0.652	0.641
	BERT-Attention	0.577	0.772	0.66	0.616	0.603	0.591
E	BERT-CNN	0.655	0.76	0.704	0.685	0.68	0.678
	BERT-LinearLayer	0.651	0.742	0.694	0.676	0.672	0.671
	BERT-Attention	0.615	0.7	0.655	0.634	0.631	0.629
A+B+C	BERT-CNN	0.706	0.603	0.651	0.68	0.676	0.674
	BERT-LinearLayer	0.706	0.558	0.623	0.67	0.663	0.659
	BERT-Attention	0.647	0.603	0.624	0.637	0.637	0.636
D+E	BERT-CNN	0.618	0.831	**0.709**	0.681	0.659	0.649
	BERT-LinearLayer	0.64	0.745	0.689	0.667	0.663	0.661
	BERT-Attention	0.6	0.742	0.663	0.631	0.624	0.618
A+B+D+E	BERT-CNN	0.673	0.7	0.686	0.68	0.68	0.68
	BERT-LinearLayer	0.67	0.7	0.685	0.678	0.678	0.678
	BERT-Attention	0.686	0.607	0.644	0.667	0.665	0.664
A+B+C+D+E	BERT-CNN	0.658	0.678	0.668	0.663	0.663	0.663
	BERT-LinearLayer	0.681	0.663	0.672	0.676	0.676	0.676
	BERT-Attention	0.595	0.693	0.64	0.614	0.61	0.608

Table 6.2: Performance of the single-task models using **HATEBERTimbau** and the best single-task result using **BERTimbau** for the different sets of training data in the test set of CO-HATE corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.5	1	0.667	0.25	0.5	0.333
A	Best with BERTimbau (BERT-LinearLayer)	0.592	0.708	0.645	0.615	0.61	0.607
	BERT-CNN	0.596	0.64	0.617	0.604	0.603	0.602
	BERT-LinearLayer	0.622	0.603	0.612	0.618	0.618	0.618
	BERT-Attention	0.583	0.79	0.671	0.629	0.612	0.6
B	Best with BERTimbau (BERT-Attention)	0.574	0.708	0.634	0.597	0.592	0.586
	BERT-CNN	0.715	0.554	0.624	0.676	0.667	0.662
	BERT-LinearLayer	0.762	0.479	0.589	0.691	0.665	0.653
	BERT-Attention	0.654	0.667	0.66	0.657	0.657	0.657
C	Best with BERTimbau (BERT-Attention)	0.61	0.633	0.621	0.614	0.614	0.614
	BERT-CNN	0.643	0.599	0.62	0.634	0.633	0.633
	BERT-LinearLayer	0.624	0.547	0.583	0.61	0.609	0.607
	BERT-Attention	0.617	0.768	0.684	0.655	0.646	0.641
D	Best with BERTimbau (BERT-LinearLayer)	0.613	0.824	0.703	0.672	0.652	0.641
	BERT-CNN	0.617	0.828	0.707	0.678	0.657	0.647
	BERT-LinearLayer	0.598	0.846	0.701	0.668	0.639	0.622
	BERT-Attention	0.608	0.801	0.691	0.658	0.642	0.633
E	Best with BERTimbau (BERT-CNN)	0.655	0.76	0.704	0.685	0.68	0.678
	BERT-CNN	0.657	0.805	**0.724**	0.703	0.693	0.689
	BERT-LinearLayer	0.695	0.708	0.701	0.699	0.699	0.698
	BERT-Attention	0.612	0.861	0.715	0.689	0.657	0.642
A+B+C	Best with BERTimbau (BERT-CNN)	0.706	0.603	0.651	0.68	0.676	0.674
	BERT-CNN	0.676	0.648	0.662	0.669	0.669	0.668
	BERT-LinearLayer	0.653	0.663	0.658	0.655	0.655	0.655
	BERT-Attention	0.645	0.693	0.668	0.656	0.655	0.655
D+E	Best with BERTimbau (BERT-CNN)	0.618	0.831	0.709	0.681	0.659	0.649
	BERT-CNN	0.633	0.809	0.711	0.685	0.67	0.664
	BERT-LinearLayer	0.676	0.697	0.686	0.682	0.682	0.682
	BERT-Attention	0.622	0.809	0.704	0.675	0.659	0.651
A+B+D+E	Best with BERTimbau (BERT-CNN)	0.673	0.7	0.686	0.68	0.68	0.68
	BERT-CNN	0.679	0.738	0.707	0.696	0.695	0.694
	BERT-LinearLayer	0.662	0.749	0.703	0.687	0.684	0.682
	BERT-Attention	0.636	0.813	0.714	0.689	0.674	0.668
A+B+C+D+E	Best with BERTimbau (BERT-LinearLayer)	0.681	0.663	0.672	0.676	0.676	0.676
	BERT-CNN	0.703	0.682	0.692	0.697	0.697	0.697
	BERT-LinearLayer	0.678	0.663	0.67	0.674	0.674	0.674
	BERT-Attention	0.642	0.794	0.71	0.686	0.676	0.671

robust representations of the OHS language phenomenon even though it was trained with Twitter data and CO-HATE focuses on Youtube data, suggesting also the portability of **HATEBERTimbau**. Analysing the models for each subset of data, the best performances are obtained by BERT-Attention (A, B, C, A+B+C, A+B+D+E, A+B+C+D+E) or BERT-CNN (D, E, D+E). The best result was obtained using the data of E and the BERT-CNN model corresponding to a Positive Class F1-score of 0.724, surpassing the baseline by 8.5% and the best result with **BERTimbau** by 2.1%.

6.2.2 Multitask Experiments

The results of all multitask models for the different sets of training data using **BERTimbau** are represented in Table 6.3. Considering the Positive Class F1-score as the benchmark metric, we can see that using the data of A, B, C, and A+B+C always leads to worse results than the results obtained by the baseline model. In the context of each subset of data, the only situation in which the multitask architecture outperformed the best single-task architecture with **BERTimbau** was when using the data of B, in particular with BERT-Attention-CS. However, looking at the other subsets of data (A, C, D, E, A+B+C, D+E, A+B+D+E and A+B+C+D+E) and making a fairer comparison, the multitask architecture showed a better performance than its equivalent single-task version (BERT-Attention) when using the data of A, A+B+C, D+E, A+B+D+E and A+B+C+D+E. Comparing BERT-Attention-CS with BERT-Attention-SNT for each subset of data, their performance varies greatly: for A, E, and A+B+C, BERT-Attention-SNT is the best; for B, D, D+E, A+B+D+E and A+B+C+D+E, BERT-Attention-CS is the best; and for C, the performances of BERT-Attention-SNT and BERT-Attention-CS are similar. The best multitask result using **BERTimbau** was obtained using the data of D+E with BERT-Attention-CS and also using the data of A+B+C+D+E with BERT-Attention-CS model, with a Positive Class F1-score of 0.671, surpassing the baseline by 0.6% but not surpassing the best single-task result using **BERTimbau**.

The results of all multitask models for the different sets of training data using **HATEBERTimbau** are represented in Table 6.4. Now, using **HATEBERTimbau**, with the exception of using the data of B with BERT-Attention-CS, all the multitask results improved, with A, C, and A+B+C surpassing the baseline, once again confirming the importance of using better semantic representations. In the context of each subset of data, the multitask architecture outperformed the best single-task architecture when using the data of A, A+B+C, A+B+D+E, A+B+C+D+E, and looking at the other subsets (B, C, D, E, D+E) only when using the data of D the multitask architecture showed a better performance than its equivalent single-task architecture. Comparing BERT-Attention-CS with BERT-Attention-SNT for each subset of data, now, almost for every case BERT-Attention-CS is the best model (A, B, C, D, E, D+E and A+B+C+D+E). The best multitask result using **HATEBERTimbau**, which is the **best multitask result** and the **best result on CO-HATE corpus**, was obtained using the data of A+B+C+D+E and the BERT-Attention-CS model with a Positive Class F1-score of 0.727, surpassing the baseline by 9% and the best single-task result

Table 6.3: Performance of the multitask models using **BERTimbau**, the best single-task model using **BERTimbau** and BERT-Attention using **BERTimbau** for the different sets of training data in the test set of CO-HATE corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.5	1	0.667	0.25	0.5	0.333
A	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.592	0.708	0.645	0.615	0.61	0.607
	BERT-Attention	0.582	0.61	0.596	0.586	0.586	0.586
	BERT-Attention CS	0.527	0.723	0.61	0.543	0.537	0.521
	BERT-Attention SNT	0.517	0.798	0.627	0.537	0.526	0.489
B	Best single-task model w/ BERTimbau (BERT-Attention)	0.574	0.708	0.634	0.597	0.592	0.586
	BERT-Attention CS	0.572	0.757	0.652	0.607	0.596	0.585
	BERT-Attention SNT	0.601	0.667	0.632	0.614	0.612	0.611
C	Best single-task model w/ BERTimbau (BERT-Attention)	0.61	0.633	0.621	0.614	0.614	0.614
	BERT-Attention CS	0.537	0.712	0.612	0.554	0.549	0.536
	BERT-Attention SNT	0.509	0.768	0.612	0.518	0.513	0.479
D	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.613	0.824	0.703	0.672	0.652	0.641
	BERT-Attention	0.577	0.772	0.66	0.616	0.603	0.591
	BERT-Attention CS	0.543	0.824	0.655	0.589	0.566	0.534
	BERT-Attention SNT	0.525	0.801	0.634	0.552	0.537	0.503
E	Best single-task model w/ BERTimbau (BERT-CNN)	0.655	0.76	0.704	0.685	0.68	0.678
	BERT-Attention	0.615	0.7	0.655	0.634	0.631	0.629
	BERT-Attention CS	0.591	0.715	0.647	0.616	0.61	0.606
	BERT-Attention SNT	0.553	0.787	0.649	0.591	0.575	0.555
A+B+C	Best single-task model w/ BERTimbau (BERT-CNN)	0.706	0.603	0.651	0.68	0.676	0.674
	BERT-Attention	0.647	0.603	0.624	0.637	0.637	0.636
	BERT-Attention CS	0.622	0.603	0.612	0.618	0.618	0.618
	BERT-Attention SNT	0.588	0.723	0.649	0.615	0.609	0.603
D+E	Best single-task model w/ BERTimbau (BERT-CNN)	0.618	0.831	**0.709**	0.681	0.659	0.649
	BERT-Attention	0.6	0.742	0.663	0.631	0.624	0.618
	BERT-Attention CS	0.624	0.727	0.671	0.648	0.644	0.642
	BERT-Attention SNT	0.524	0.888	0.659	0.579	0.541	0.479
A+B+D+E	Best single-task model w/ BERTimbau (BERT-CNN)	0.673	0.7	0.686	0.68	0.68	0.68
	BERT-Attention	0.686	0.607	0.644	0.667	0.665	0.664
	BERT-Attention CS	0.598	0.757	0.668	0.633	0.624	0.617
	BERT-Attention SNT	0.608	0.738	0.667	0.637	0.631	0.627
A+B+C+D+E	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.681	0.663	0.672	0.676	0.676	0.676
	BERT-Attention	0.595	0.693	0.64	0.614	0.61	0.608
	BERT-Attention CS	0.627	0.723	0.671	0.65	0.646	0.644
	BERT-Attention SNT	0.636	0.693	0.663	0.649	0.648	0.647

Table 6.4: Performance of the multitask models using **HATEBERTimbau**, the best single-task model and BERT-Attention using **HATEBERTimbau** for the different sets of training data in the test set of CO-HATE corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.5	1	0.667	0.25	0.5	0.333
A	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.583	0.79	0.671	0.629	0.612	0.6
	BERT-Attention CS	0.574	0.846	0.684	0.64	0.609	0.585
	BERT-Attention SNT	0.565	0.801	0.663	0.611	0.592	0.573
B	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.654	0.667	0.66	0.657	0.657	0.657
	BERT-Attention CS	0.635	0.652	0.643	0.639	0.639	0.639
	BERT-Attention SNT	0.647	0.625	0.636	0.642	0.642	0.642
C	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.617	0.768	0.684	0.655	0.646	0.641
	BERT-Attention CS	0.595	0.798	0.682	0.644	0.627	0.616
	BERT-Attention SNT	0.532	0.869	0.66	0.587	0.552	0.503
D	Best single-task model (BERT-CNN)	0.617	0.828	0.707	0.678	0.657	0.647
	BERT-Attention w/ HATEBERTimbau	0.608	0.801	0.691	0.658	0.642	0.633
	BERT-Attention CS	0.599	0.828	0.695	0.66	0.637	0.623
	BERT-Attention SNT	0.586	0.757	0.66	0.621	0.61	0.602
E	Best single-task model (BERT-CNN)	0.657	0.805	0.724	0.703	0.693	0.689
	BERT-Attention w/ HATEBERTimbau	0.612	0.861	0.715	0.689	0.657	0.642
	BERT-Attention CS	0.628	0.82	0.711	0.684	0.667	0.659
	BERT-Attention SNT	0.58	0.816	0.678	0.635	0.612	0.596
A+B+C (IAA: 0.36)	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.645	0.693	0.668	0.656	0.655	0.655
	BERT-Attention CS	0.633	0.749	0.686	0.663	0.657	0.654
	BERT-Attention SNT	0.651	0.76	0.701	0.681	0.676	0.674
D+E (IAA: 0.735)	Best single-task model (BERT-CNN)	0.633	0.809	0.711	0.685	0.67	0.664
	BERT-Attention w/ HATEBERTimbau	0.622	0.809	0.704	0.675	0.659	0.651
	BERT-Attention CS	0.61	0.828	0.703	0.672	0.65	0.638
	BERT-Attention SNT	0.586	0.828	0.686	0.647	0.622	0.605
A+B+D+E (IAA: 0.625)	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.636	0.813	0.714	0.689	0.674	0.668
	BERT-Attention CS	0.63	0.79	0.701	0.674	0.663	0.657
	BERT-Attention SNT	0.632	0.843	0.722	0.698	0.676	0.667
A+B+C+D+E (IAA: 0.478)	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.642	0.794	0.71	0.686	0.676	0.671
	BERT-Attention CS	0.639	0.843	**0.727**	0.704	0.684	0.675
	BERT-Attention SNT	0.647	0.768	0.702	0.68	0.674	0.671

Table 6.5: Results of all experiments using for training A+B+C and D+E in the test set of CO-HATE corpus.

Model	BERT pre-trained model	Data of A+B+C used for training (IAA: 0.36)			Data of D+E used for training (IAA: 0.735)		
		Positive Class			Positive Class		
		Prec	Rec	F1	Prec	Rec	F1
BERT-CNN	BERTimbau	0,706	0,603	0,651	0,618	0,831	0,709
	HATEBERTimbau	0,676	0,648	0,662	0,633	0,809	0,711
BERT-LinearLayer	BERTimbau	0,706	0,558	0,623	0,64	0,745	0,689
	HATEBERTimbau	0,653	0,663	0,658	0,676	0,697	0,686
BERT-Attention	BERTimbau	0,647	0,603	0,624	0,6	0,742	0,663
	HATEBERTimbau	0,645	0,693	0,668	0,622	0,809	0,704
BERT-Attention-CS	BERTimbau	0,622	0,603	0,612	0,624	0,727	0,671
	HATEBERTimbau	0,633	0,749	0,686	0,61	0,828	0,703
BERT-Attention-SNT	BERTimbau	0,588	0,723	0,649	0,524	0,888	0,659
	HATEBERTimbau	0,651	0,76	0,701	0,586	0,828	0,686

by 0.4%.

6.2.3 Data Agreement Impact On Results

In this section, we compare the results obtained in all experiments between some subsets of data used for training. In Table 6.5 the comparison is between A+B+C and D+E, and in Table 6.6 the comparison is between A+B+D+E (without annotator C, the most discordant with all the others) and A+B+C+D+E.

Almost every Positive Class F1-score obtained by the annotators not belonging to the communities targeted in this study, D+E, is greater than the Positive Class F1-scores obtained by the annotators belonging to those communities, A+B+C (BERT-Attention-SNT with **HATEBERTimbau** is the exception). We believe that the fact that A, B and C tended to disagree more with each other (IAA of 0.36 vs IAA of 0.735) than D and E may have contributed to these results.

Using the data of A+B+D+E for training obtains better results than using the data of A+B+C+D+E with almost every model (BERT-Attention-CS with **HATEBERTimbau** and BERT-Attention-CS with **BERTimbau** are the exceptions). This suggests that a higher IAA of the training data tends to lead to higher performance and that the multitask architecture sometimes might take advantage from more training data.

6.2.4 Target-Specific Model

In this section, we report the results of a target-specific model experience. We trained a model with just the data targeting one specific target group and evaluated its performance only on the messages of the test set that target that target group. We chose to do this experiment with the model that obtained the best result in the previous experiences: BERT-Attention-CS with messages from all annotators and

Table 6.6: Results of all experiments using for training A+B+D+E and A+B+C+D+E in the test set of CO-HATE corpus.

Model	BERT pre-trained model	Data of A+B+D+E used for training (IAA: 0.625)			Data of A+B+C+D+E used for training (IAA: 0.478)		
		Positive Class			Positive Class		
		Prec	Rec	F1	Prec	Rec	F1
BERT-CNN	BERTimbau	0,673	0,7	0,686	0,658	0,678	0,668
	HATEBERTimbau	0,679	0,738	0,707	0,703	0,682	0,692
BERT-LinearLayer	BERTimbau	0,67	0,7	0,685	0,681	0,663	0,672
	HATEBERTimbau	0,662	0,749	0,703	0,678	0,663	0,67
BERT-Attention	BERTimbau	0,686	0,607	0,644	0,595	0,693	0,64
	HATEBERTimbau	0,636	0,813	0,714	0,642	0,794	0,71
BERT-Attention-CS	BERTimbau	0,598	0,757	0,668	0,627	0,723	0,671
	HATEBERTimbau	0,63	0,79	0,701	0,639	0,843	0,727
BERT-Attention-SNT	BERTimbau	0,608	0,738	0,667	0,636	0,693	0,663
	HATEBERTimbau	0,632	0,843	0,722	0,647	0,768	0,702

Table 6.7: Results of the African descent-directed HS Model and the Generic HS Model in the African descent-directed messages of the CO-HATE test set.

Model	Positive Class			Macro Avg		
	Prec	Rec	F1	Prec	Rec	F1
Generic HS Model	0.798	0.867	0.831	0.561	0.546	0.548
African descent-directed HS Model	0.812	0.85	0.831	0.589	0.578	0.582

HATEBERTimbau. We selected the African descent community since is the most represented one in both training and test sets of CO-HATE corpus. In Table 6.7 we report the results of this model, which we name African descent-directed HS Model, and the generic HS Model, which was trained with all the training data of CO-HATE corpus. As we can see, considering the Positive Class F1-score, the two models obtain similar performances, having no impact on the results creating a target-specific model in comparison to a generic one.

6.2.5 Error Analysis

This section discusses the major classification errors derived from our best model on CO-HATE corpus: BERT-Attention-CS with **HATEBERTimbau**, trained with the data from all annotators. In Figure 6.1, we present the confusion matrix of our best model in order to better visualize the classification errors. As

actual values		Non-HS	HS
	Non-HS	140	127
	HS	42	225
		Non-HS	HS
		predicted values	

Figure 6.1: Confusion matrix of the best performing model on CO-HATE corpus: BERT-Attention-CS using **HATEBERTimbau** trained with all training data.

we had already shown in the Table 6.4, we have interesting results in terms of Recall, but the Precision is not at the same level. We have inspected the 127 messages that were incorrectly classified as HS and found that some of these messages (Examples 4, 5 and 6) include words and expressions that are also highly frequently used in messages classified as conveying HS (e.g., *Rendimento Social de Inserção (RSI)*, *abonos*, and *subsidiodependência*; 'Social Integration Income', 'subsidies', and 'subsidy dependence'), which may have influenced the classifier in the training phase. Also, according to the annotation and using the at least two votes criteria, 98 of these messages (77.2%) correspond either to Counterspeech (Example 7) or Offensive Speech (Example 8), forms of speech that share a lot of the vocabulary with HS making also harder for the models to distinguish them.

4. *Nem direito tive ao rsi porque moro com um familiar que aufrere de 500€ de reforma.*
   I didn't even have the right to rsi because I live with a family member who has a pension of €500.

5. *Abonos de 4500€? Mentira mentira máximo 200€ segurança social já pagou mais mas n hoje.*
   €4500 allowances? Lie lie maximum €200 social security has already paid more but not today.

6. *Quem é que fez esta lei de. subsídiodepencia???, eles não são culpados, quem é culpado é quem fez esta lei*
   Who made this law of. subsidy dependence???, they are not guilty, the guilty is who made this law

7. *Triste triste PORTUGAL ser feito de pessoas como você nem todos os pretos são fruta podre por isso não englobe todos pois os brancos não são melhores que outras raças....*
   Sad sad PORTUGAL is made of people like you not all blacks are rotten fruit so don't encompass them all because whites are not better than other races....

8. *'Só português burro e que fica neste país e depois reclama lol'*
   Only dumb Portuguese stays in this country and then complains lol

We have also manually inspected the 42 messages that were incorrectly classified as Non-HS. More than two thirds of these messages (67%), according to the annotation and using the at least two votes criteria, correspond to Indirect HS (Examples 9, 10 and 11), which is hard to detect given that it often resorts figurative speech, including irony and rhetorical questions, and rhetorical strategies to attack or humiliate the HS targets. But when looking at these messages, we can observe that out of context (unlike the annotators had), they are difficult to interpret. In particular, Example 11 is an answer to another message that is "invisible" to the classifier, making almost impossible to understand it. Another observation that can be made is the fact that 17 (41%) of these messages (Example 9) were classified as HS by exactly two annotators, meaning that in the view of three annotators they are not HS, precisely the prediction of the model.

9. *Isto é um Crime a Protecao de Menores Não Faz Nada ? Afinal os filhos são um negocio*
   Is this a Crime Protection of Minors Does Nothing? After all, children are a business.

10. *Que Raiva este Video me está a dar mas as autoridades competentes nao fazem nada?*
    What rage is this video giving me do the competent authorities do nothing?

11. *@Alexandre Zua Caldeira so se estiveres a falar de africanos com cartão de cidadão oferecido*
    @Alexandre Zua Caldeira only if you are talking about Africans with a citizen card offered

6.3 Results for the FIGHT Corpus

In this section, the results of the detection of OHS using the FIGHT corpus as training, development, and testing data are presented. This corpus was annotated by five different annotators and led to a low IAA (0.362), reinforcing the difficulty and subjectivity of this task. Just like in Section 6.2, we have tested several combinations of data for the training of the models. Besides using the entire training data, i.e., all the messages annotated by annotators A, B, C, D, and E, we have also experimented using the corpus annotated by each user independently (A, B, C, D, and E). Since annotator A was the one having the worse IAA when compared to the remaining annotators (0.17 on average, while the others have at least 0.34, see Table 4.10), we tested the combination B+C+D+E. We also tested with the annotators C and A, which do not belong to any potential historically marginalized group, and with B, D and E, composed by annotators that belong to the target communities. With these subsets of data we may understand how the data agreement and annotators' profile may affect the performance of OHS detection.

6.3.1 Single-task Experiments

The results of all single-task models for the different sets of training data using **BERTimbau** are represented in Table 6.8. We present the results of our baseline, a dummy classifier that classifies all instances as HS. Considering the Positive Class F1-score as the benchmark metric, we only achieve worse results than the baseline when using exclusively the data of annotator A. Analysing the models for each subset of data, their performance varies greatly: for A, D, A+C, and B+C+D+E, BERT-LinearLayer is the best; for E, BERT-Attention is the best; and for B, C, B+D+E, and A+B+C+D+E, BERT-CNN is the best. The best result was obtained when using the data of B+D+E with BERT-CNN and the data of D with BERT-LinearLayer corresponding to a Positive Class F1-score of 0.665, surpassing the baseline by 20%.

We performed exactly the same experiences but using **HATEBERTimbau** and those results are presented in Table 6.9, with the best result using **BERTimbau** being also shown. Using **HATEBERTimbau** and the data of A still cannot outperform the baseline model. Actuality, none of the results of A could beat

Table 6.8: Performance of the single-task models using **BERTimbau** for the different sets of training data in the test set of FIGHT corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.383	1	0.554	0.192	0.5	0.277
A	BERT-CNN	0.592	0.462	0.519	0.649	0.632	0.635
	BERT-LinearLayer	0.592	0.496	0.54	0.654	0.642	0.645
	BERT-Attention	0.519	0.546	0.532	0.614	0.616	0.614
B	BERT-CNN	0.656	0.582	0.617	0.707	0.696	0.7
	BERT-LinearLayer	0.722	0.496	0.588	0.73	0.689	0.696
	BERT-Attention	0.623	0.54	0.579	0.68	0.669	0.672
C	BERT-CNN	0.737	0.572	0.644	0.752	0.723	0.73
	BERT-LinearLayer	0.735	0.564	0.638	0.749	0.719	0.726
	BERT-Attention	0.612	0.572	0.591	0.678	0.673	0.675
D	BERT-CNN	0.699	0.569	0.627	0.729	0.708	0.714
	BERT-LinearLayer	0.619	0.718	**0.665**	0.713	0.722	0.714
	BERT-Attention	0.568	0.58	0.574	0.652	0.653	0.652
E	BERT-CNN	0.724	0.473	0.572	0.727	0.68	0.687
	BERT-LinearLayer	0.697	0.486	0.572	0.714	0.677	0.683
	BERT-Attention	0.631	0.577	0.603	0.691	0.684	0.687
A+C	BERT-CNN	0.695	0.54	0.608	0.722	0.696	0.703
	BERT-LinearLayer	0.697	0.582	0.634	0.731	0.713	0.718
	BERT-Attention	0.554	0.561	0.558	0.64	0.64	0.64
B+D+E	BERT-CNN	0.701	0.632	**0.665**	0.743	0.732	0.737
	BERT-LinearLayer	0.801	0.567	0.664	0.787	0.74	0.75
	BERT-Attention	0.691	0.509	0.586	0.715	0.684	0.69
B+C+D+E	BERT-CNN	0.823	0.533	0.647	0.792	0.731	0.742
	BERT-LinearLayer	0.796	0.569	0.664	0.784	0.739	0.75
	BERT-Attention	0.716	0.533	0.611	0.733	0.701	0.708
A+B+C+D+E	BERT-CNN	0.74	0.601	0.663	0.759	0.735	0.742
	BERT-LinearLayer	0.851	0.478	0.612	0.798	0.713	0.723
	BERT-Attention	0.707	0.504	0.588	0.723	0.687	0.694

Table 6.9: Performance of the single-task models using **HATEBERTimbau** and the best single-task model using **BERTimbau** for the different sets of training data in the test set of FIGHT corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.383	1	0.554	0.192	0.5	0.277
A	Best with BERTimbau (BERT-LinearLayer)	0.592	0.496	0.54	0.654	0.642	0.645
	BERT-CNN	0.519	0.522	0.521	0.611	0.611	0.611
	BERT-LinearLayer	0.568	0.402	0.471	0.627	0.606	0.607
	BERT-Attention	0.469	0.546	0.504	0.577	0.581	0.577
B	Best with BERTimbau (BERT-CNN)	0.656	0.582	0.617	0.707	0.696	0.7
	BERT-CNN	0.691	0.661	0.676	0.743	0.739	0.741
	BERT-LinearLayer	0.724	0.52	0.605	0.735	0.698	0.706
	BERT-Attention	0.618	0.614	0.616	0.69	0.689	0.69
C	Best with BERTimbau (BERT-CNN)	0.737	0.572	0.644	0.752	0.723	0.73
	BERT-CNN	0.783	0.603	0.681	0.784	0.75	0.759
	BERT-LinearLayer	0.76	0.619	0.682	0.774	0.749	0.756
	BERT-Attention	0.667	0.632	0.649	0.723	0.718	0.72
D	Best with BERTimbau (BERT-LinearLayer)	0.619	0.718	0.665	0.713	0.722	0.714
	BERT-CNN	0.662	0.655	0.659	0.725	0.724	0.724
	BERT-LinearLayer	0.731	0.603	0.661	0.754	0.733	0.739
	BERT-Attention	0.688	0.598	0.64	0.728	0.715	0.719
E	Best with BERTimbau (BERT-Attention)	0.631	0.577	0.603	0.691	0.684	0.687
	BERT-CNN	0.743	0.499	0.597	0.742	0.696	0.704
	BERT-LinearLayer	0.769	0.418	0.541	0.744	0.67	0.675
	BERT-Attention	0.703	0.551	0.618	0.729	0.703	0.71
A+C	Best with BERTimbau (BERT-LinearLayer)	0.697	0.582	0.634	0.731	0.713	0.718
	BERT-CNN	0.716	0.567	0.633	0.739	0.714	0.72
	BERT-LinearLayer	0.72	0.59	0.648	0.745	0.724	0.73
	BERT-Attention	0.609	0.663	0.635	0.694	0.7	0.696
B+D+E	Best with BERTimbau (BERT-CNN)	0.701	0.632	0.665	0.743	0.732	0.737
	BERT-CNN	0.727	0.681	0.704	0.768	0.761	0.764
	BERT-LinearLayer	0.835	0.606	0.702	0.813	0.766	0.777
	BERT-Attention	0.684	0.611	0.646	0.729	0.718	0.722
B+C+D+E	Best with BERTimbau (BERT-LinearLayer)	0.796	0.569	0.664	0.784	0.739	0.75
	BERT-CNN	0.854	0.598	0.704	0.822	0.767	0.78
	BERT-LinearLayer	0.819	0.637	0.717	0.81	0.775	0.785
	BERT-Attention	0.801	0.546	0.649	0.783	0.731	0.741
A+B+C+D+E	Best with BERTimbau (BERT-CNN)	0.74	0.601	0.663	0.759	0.735	0.742
	BERT-CNN	0.87	0.577	0.694	0.826	0.762	0.775
	BERT-LinearLayer	0.833	0.64	**0.724**	0.819	0.78	0.791
	BERT-Attention	0.775	0.54	0.637	0.768	0.722	0.731

the best model using **BERTimbau** and the same happened using the data of D. For every other subset of data (B, C, E, A+C, B+D+E, B+C+D+E, A+B+C+D+E) the obtained results were better than when using **BERTimbau**, again demonstrating the importance of using better semantic representations, which was indeed retrained with Twitter data. Then representations of the OHS language phenomenon seem to be more robust than those of **BERTimbau**. Analysing the models for each subset of data, BERT-CNN performs the best using the data of A, B, and B+D+E, BERT-LinearLayer performs the best using the data of C, D, A+C, B+C+D+E, and A+B+C+D+E, and BERT-Attention performs the best using the data of E. The best result was obtained using the data of A+B+C+D+E and the BERT-LinearLayer model with a Positive Class F1-score of 0.724, surpassing the baseline by 30.7% and the best single-task result with **BERTimbau** by 8.9%.

6.3.2 Multitask Experiments

The results of all multitask models for the different sets of training data using **BERTimbau** are represented in Table 6.10. Considering the Positive Class F1-score as the benchmark metric, we can see that for all subsets of data the multitask models overcome the baseline model. Only when using the data of A and E, the multitask architecture outperformed the best single-task architecture using **BERTimbau**. However, looking at the other subsets of data (B, C, D, A+C, B+D+E, B+C+D+E, A+B+C+D+E) and making a fairer comparison, the multitask architecture showed a better performance than its equivalent single-task version for all these cases. Comparing BERT-Attention-CS with BERT-Attention-SNT for each subset of data, BERT-Attention-SNT in most cases overcame BERT-Attention-CS: using the data of B, D, E, A+C, B+D+E, and A+B+C+D+E. The best multitask result using **BERTimbau** was obtained using the data of D and the BERT-Attention-SNT model with a Positive Class F1-score of 0.645, surpassing the baseline by 16.4% but not surpassing the best single-task result using **BERTimbau**.

The results of all multitask models for the different sets of training data using **HATEBERTimbau** are represented in Table 6.11. Now, using **HATEBERTimbau**, for most subsets of data the multitask results improved (B, C, D, E, A+C, B+D+E, B+C+D+E, and A+B+C+D+E) suggesting again the superiority of **HATEBERTimbau** over **BERTimbau**. Comparing the multitask models using **HATEBERTimbau** with the equivalent single-task model (BERT-Attention) using also **HATEBERTimbau** we can see that with the training the data of A, B, B+D+E, B+C+D+E, and A+B+C+D+E, the multitask architecture can overcome the single-task one. For almost every subset of data, the multitask architecture was not able to outperform the best single-task architecture (using the data of B, C, D, E, A+C, B+D+E, B+C+D+E and A+B+C+D+E). Comparing BERT-Attention-CS with BERT-Attention-SNT for each subset of data, now, in most cases, BERT-Attention-CS is the best model (A, B, D, B+C+D+E, A+B+C+D+E). The best multitask result using **HATEBERTimbau**, which is the best multitask result, was obtained using the data of B+D+E and the BERT-Attention-SNT model with a Positive Class F1-score of 0.66, surpassing the base-

Table 6.10: Performance of the multitask models using **BERTimbau**, the best single-task model using **BERTimbau** and BERT-Attention using **BERTimbau** for the different sets of training data in the test set of FIGHT corpus.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.383	1	0.554	0.192	0.5	0.277
A	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.592	0.496	0.54	0.654	0.642	0.645
	BERT-Attention	0.519	0.546	0.532	0.614	0.616	0.614
	BERT-Attention CS	0.457	0.836	0.591	0.624	0.61	0.554
	BERT-Attention SNT	0.465	0.668	0.548	0.591	0.595	0.576
B	Best single-task model w/ BERTimbau (BERT-CNN)	0.656	0.582	0.617	0.707	0.696	0.7
	BERT-Attention	0.623	0.54	0.579	0.68	0.669	0.672
	BERT-Attention CS	0.619	0.548	0.582	0.679	0.67	0.673
	BERT-Attention SNT	0.586	0.648	0.615	0.676	0.682	0.678
C	Best single-task model w/ BERTimbau (BERT-CNN)	0.737	0.572	0.644	0.752	0.723	0.73
	BERT-Attention	0.612	0.572	0.591	0.678	0.673	0.675
	BERT-Attention CS	0.545	0.713	0.618	0.662	0.672	0.657
	BERT-Attention SNT	0.622	0.593	0.607	0.688	0.685	0.686
D	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.619	0.718	**0.665**	0.713	0.722	0.714
	BERT-Attention	0.568	0.58	0.574	0.652	0.653	0.652
	BERT-Attention CS	0.578	0.616	0.597	0.665	0.669	0.666
	BERT-Attention SNT	0.588	0.715	0.645	0.692	0.702	0.692
E	Best single-task model w/ BERTimbau (BERT-Attention)	0.631	0.577	0.603	0.691	0.684	0.687
	BERT-Attention CS	0.486	0.752	0.591	0.627	0.63	0.601
	BERT-Attention SNT	0.602	0.608	0.605	0.679	0.679	0.679
A+C	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.697	0.582	0.634	0.731	0.713	0.718
	BERT-Attention	0.554	0.561	0.558	0.64	0.64	0.64
	BERT-Attention CS	0.58	0.606	0.593	0.664	0.667	0.665
	BERT-Attention SNT	0.536	0.702	0.608	0.653	0.662	0.648
B+D+E	Best single-task model w/ BERTimbau (BERT-CNN)	0.701	0.632	**0.665**	0.743	0.732	0.737
	BERT-Attention	0.691	0.509	0.586	0.715	0.684	0.69
	BERT-Attention CS	0.657	0.574	0.613	0.706	0.694	0.698
	BERT-Attention SNT	0.655	0.601	0.627	0.71	0.702	0.705
B+C+D+E	Best single-task model w/ BERTimbau (BERT-LinearLayer)	0.796	0.569	0.664	0.784	0.739	0.75
	BERT-Attention	0.716	0.533	0.611	0.733	0.701	0.708
	BERT-Attention CS	0.706	0.551	0.619	0.73	0.704	0.711
	BERT-Attention SNT	0.659	0.574	0.614	0.707	0.695	0.699
A+B+C+D+E	Best single-task model w/ BERTimbau (BERT-CNN)	0.74	0.601	0.663	0.759	0.735	0.742
	BERT-Attention	0.707	0.504	0.588	0.723	0.687	0.694
	BERT-Attention CS	0.684	0.559	0.615	0.719	0.699	0.705
	BERT-Attention SNT	0.662	0.582	0.619	0.71	0.699	0.703

Table 6.11: Performance of the multitask models using **HATEBERTimbau** (in gray), the best single-task model and BERT-Attention using **HATEBERTimbau** for the different sets of training data in the test set of FIGHT corpus. Best multitask model results using **BERTimbau** are also shown in the cases where they obtain better results.

Data used for training	Model	Positive Class			Macro Avg		
		Prec	Rec	F1	Prec	Rec	F1
	Dummy Classifier	0.383	1	0.554	0.192	0.5	0.277
A	Best single-task model (BERT-LinearLayer w/ BERTimbau)	0.592	0.496	0.54	0.654	0.642	0.645
	BERT-Attention w/ HATEBERTimbau	0.469	0.546	0.504	0.577	0.581	0.577
	BERT-Attention CS w/ BERTimbau	0.457	0.836	0.591	0.624	0.61	0.554
	BERT-Attention CS	0.438	0.718	0.544	0.574	0.573	0.539
	BERT-Attention SNT w/ BERTimbau	0.465	0.668	0.548	0.591	0.595	0.576
	BERT-Attention SNT	0.467	0.598	0.525	0.583	0.587	0.578
B	Best single-task model (BERT-CNN w/ HATEBERTimbau)	0.691	0.661	0.676	0.743	0.739	0.741
	BERT-Attention w/ HATEBERTimbau	0.618	0.614	0.616	0.69	0.689	0.69
	BERT-Attention CS	0.55	0.708	0.619	0.664	0.674	0.661
	BERT-Attention SNT w/ BERTimbau	0.586	0.648	0.615	0.676	0.682	0.678
	BERT-Attention SNT	0.604	0.621	0.613	0.682	0.684	0.683
C	Best single-task model (BERT-LinearLayer w/ HATEBERTimbau)	0.76	0.619	0.682	0.774	0.749	0.756
	BERT-Attention w/ HATEBERTimbau	0.667	0.632	0.649	0.723	0.718	0.72
	BERT-Attention CS	0.69	0.574	0.627	0.725	0.707	0.712
	BERT-Attention SNT	0.583	0.679	0.627	0.68	0.689	0.682
D	Best single-task model (BERT-LinearLayer w/ BERTimbau)	0.619	0.718	0.665	0.713	0.722	0.714
	BERT-Attention w/ HATEBERTimbau	0.688	0.598	0.64	0.728	0.715	0.719
	BERT-Attention CS	0.567	0.7	0.626	0.674	0.684	0.673
	BERT-Attention SNT w/ BERTimbau	0.588	0.715	0.645	0.692	0.702	0.692
	BERT-Attention SNT	0.597	0.611	0.604	0.676	0.677	0.677
E	Best single-task model (BERT-Attention w/ HATEBERTimbau)	0.703	0.551	0.618	0.729	0.703	0.71
	BERT-Attention CS	0.541	0.684	0.604	0.653	0.662	0.651
	BERT-Attention SNT	0.584	0.645	0.613	0.674	0.68	0.676
A+C (IAA: 0.189)	Best single-task model (BERT-LinearLayer w/ HATEBERTimbau)	0.72	0.59	0.648	0.745	0.724	0.73
	BERT-Attention w/ HATEBERTimbau	0.609	0.663	0.635	0.694	0.7	0.696
	BERT-Attention CS	0.536	0.757	0.628	0.667	0.675	0.654
	BERT-Attention SNT	0.622	0.648	0.634	0.698	0.701	0.7
B+D+E (IAA: 0.439)	Best single-task model (BERT-CNN w/ HATEBERTimbau)	0.727	0.681	0.704	0.768	0.761	0.764
	BERT-Attention w/ HATEBERTimbau	0.684	0.611	0.646	0.729	0.718	0.722
	BERT-Attention CS	0.663	0.642	0.653	0.723	0.72	0.721
	BERT-Attention SNT	0.675	0.645	0.66	0.73	0.726	0.728
B+C+D+E (IAA: 0.484)	Best single-task model (BERT-LinearLayer w/ HATEBERTimbau)	0.819	0.637	0.717	0.81	0.775	0.785
	BERT-Attention w/ HATEBERTimbau	0.801	0.546	0.649	0.783	0.731	0.741
	BERT-Attention CS	0.74	0.58	0.65	0.755	0.727	0.734
	BERT-Attention SNT	0.691	0.585	0.634	0.728	0.711	0.717
A+B+C+D+E (IAA: 0.362)	Best single-task model (BERT-LinearLayer w/ HATEBERTimbau)	0.833	0.64	**0.724**	0.819	0.78	0.791
	BERT-Attention w/ HATEBERTimbau	0.775	0.54	0.637	0.768	0.722	0.731
	BERT-Attention CS	0.636	0.661	0.648	0.71	0.713	0.711
	BERT-Attention SNT	0.765	0.554	0.642	0.764	0.724	0.733

Table 6.12: Results of all experiments using for training A+C and B+D+E in the test set of FIGHT corpus.

Model	BERT pre-trained model	Data of A+C used for training (IAA: 0.189)			Data of B+D+E used for training (IAA: 0.439)		
		Positive Class			Positive Class		
		Prec	Rec	F1	Prec	Rec	F1
BERT-CNN	BERTimbau	0,695	0,54	0,608	0,701	0,632	0,665
	HATEBERTimbau	0,716	0,567	0,633	0,727	0,681	0,704
BERT-LinearLayer	BERTimbau	0,697	0,582	0,634	0,801	0,567	0,664
	HATEBERTimbau	0,72	0,59	0,648	0,835	0,606	0,702
BERT-Attention	BERTimbau	0,554	0,561	0,558	0,691	0,509	0,586
	HATEBERTimbau	0,609	0,663	0,635	0,684	0,611	0,646
BERT-Attention-CS	BERTimbau	0,58	0,606	0,593	0,657	0,574	0,613
	HATEBERTimbau	0,536	0,757	0,628	0,663	0,642	0,653
BERT-Attention-SNT	BERTimbau	0,536	0,702	0,608	0,655	0,601	0,627
	HATEBERTimbau	0,622	0,648	0,634	0,675	0,645	0,66

Table 6.13: Results of all experiments using for training B+C+D+E and A+B+C+D+E in the test set of FIGHT corpus.

Model	BERT pre-trained model	Data of B+C+D+E used for training (IAA: 0.484)			Data of A+B+C+D+E used for training (IAA: 0.362)		
		Positive Class			Positive Class		
		Prec	Rec	F1	Prec	Rec	F1
BERT-CNN	BERTimbau	0,823	0,533	0,647	0,74	0,601	0,663
	HATEBERTimbau	0,854	0,598	0,704	0,87	0,577	0,694
BERT-LinearLayer	BERTimbau	0,796	0,569	0,664	0,851	0,478	0,612
	HATEBERTimbau	0,819	0,637	0,717	0,833	0,64	0,724
BERT-Attention	BERTimbau	0,716	0,533	0,611	0,707	0,504	0,588
	HATEBERTimbau	0,801	0,546	0,649	0,775	0,54	0,637
BERT-Attention-CS	BERTimbau	0,706	0,551	0,619	0,684	0,559	0,615
	HATEBERTimbau	0,74	0,58	0,65	0,636	0,661	0,648
BERT-Attention-SNT	BERTimbau	0,659	0,574	0,614	0,662	0,582	0,619
	HATEBERTimbau	0,691	0,585	0,634	0,765	0,554	0,642

line by 19.1% but not surpassing the best single-task result (Positive Class F1-score of 0.724 obtained by BERT-LinearLayer with the data of A+B+C+D+E using **HATEBERTimbau**), which is 9.7% higher.

6.3.3 Data Agreement Impact on Results

In this section, we compare the results obtained in all experiments between some subsets of data used for training. In Table 6.12 the comparison is between A+C and B+D+E, and in Table 6.13 the comparison is between B+C+D+E (without annotator A, the most discordant with all the others) and A+B+C+D+E.

Every single Positive Class F1-score obtained by the annotators belonging to the communities targeted in this study, B+D+E, is greater than the Positive Class F1-scores obtained by the annotators not belonging to those communities, A+C. It may have contributed to these results the fact that A and C disagree more with each other (IAA of 0.189 vs IAA of 0.439) than B, D, and E.

Table 6.14: Results of the LGBTQI+-directed HS Model and Generic HS Model in the LGBTQI+-directed messages of the FIGHT corpus test set.

Model	Positive Class			Macro Avg		
	Prec	Rec	F1	Prec	Rec	F1
Generic HS Model	0.908	0.742	0.817	0.756	0.791	0.76
LGBTQI+-directed HS Model	0.852	0.864	**0.858**	0.776	0.772	0.774

actual values	Non-HS	568	49
	HS	138	245
		Non-HS	HS
		predicted values	

Figure 6.2: Confusion matrix of the best performing model on FIGHT corpus: BERT-LinearLayer using **HATEBERTimbau** trained with all training data.

We were expecting that B+C+D+E would in general obtain a better performance than A+B+C+D+E due to the higher IAA between those annotators (IAA of 0.484 vs IAA of 0.362) and indeed for six of the ten models that happened.

6.3.4 Target-Specific Model

In this section, we describe and report the results of a target-specific model experiment, similarly to what is done in Section 6.2.4.

The LGBTQI+ community is the most represented one in both training and test sets of CO-HATE corpus so we tested an LGBTQI+-directed HS Model against a Generic HS Model, being the architecture selected the one that obtained the best result in the previous experiences: BERT-LinearLayer with messages from all annotators and **HATEBERTimbau**. In Table 6.14 we report the results of both models on the LGBTQI+-directed messages of FIGHT corpus test set. As we can see, the target-specific HS model produced a better result than the Generic HS Model (Positive Class F1-score of 0.858 against 0.817) suggesting that the detection of OHS might depend on the target group. The Recall improved by 16% and the Precision, contrary to what happened in Section 6.2.4 decreased (by 6%), but still remained high.

6.3.5 Error Analysis

This section discusses the major classification errors derived from our best model on FIGHT corpus, BERT-LinearLayer with **HATEBERTimbau**, trained with the data from all annotators. In Figure 6.2, we present the confusion matrix of our best model in order to better visualize the classification errors. As we had already shown in the Table 6.11, we have interesting results in terms of Precision (0.833), but the Recall (0.64) is not at the same level. We have inspected the 49 messages that were incorrectly

classified as HS and most of these messages are associated with the LGBTQI+ community (Examples 12 and 13), namely they contain words such as *gay* and *lésbica* ('lesbian'). We believe that the presence of these words might be misleading the classifier and indeed more than a third of the messages of the training data that contain HS and are directed to the LGBTQI+ community contain these words. Also, like in CO-HATE corpus, a large portion (49%) of these messages are, according to the annotation and using the at least two votes criteria, or Counterspeech (Example 14) or Offensive Speech (Example 15).

12. *Ok comunidade gay, já percebi mas tmb não sou a defensora da paz de ng, deixem-me em sossegada obg*
    Ok gay community, I get it but I'm not the defender of anyone's peace, leave me alone thank you

13. *Russia é um país bué homofóbico , será que há lésbicas ou bi lá?*
    Russia is a very homophobic country, are there lesbians or bisexuals there?

14. *A novidade é: há pessoas que simplesmente não querem ou não conseguem ter filhos. Não é a homossexualidade que vem dar cabo da natalidade*
    The novelty is: there are people who simply do not want or cannot have children. It's not homosexuality that ends the birth rate

15. *Mentiroso da merda!!*
    Fucking liar!!

We have also manually inspected the 138 messages that were incorrectly classified as Non-HS. Most of these messages (59%), according to the annotation and using the at least two votes criteria, correspond to Direct HS (Examples 16 and 17), which is the kind of HS that should be more easily detected. This is probably due to the fact that Direct HS is much more present in the test set than Indirect HS. Also, 67 (49%) of these messages (Example 18) were classified as HS by exactly two annotators, having the classifier made a good classification from the perspective of the other three annotators.

16. *N sujo o carro com ciganos*
    I don't dirty the car with gypsies

17. *Pensei que fosses apenas mais um hétero odioso, mas o facto de seres gay torna-te ainda mais escroto*
    I thought you were just another hateful straight guy, but the fact that you're gay makes you even more dickhead

18. *Mandar a carta do racismo ao ar a ver se pega. Funny tho*
    Send the racism letter to the air to see if it catches on. funny tho

6.4 Summary and Discussion

This section summarises the results of the experiments carried out in this work. For each corpus we tested 9 different subsets of data, 5 different models with 2 different BERT pre-trained models making a total of 180 different experiments, 90 for each corpus.

Varying the BERT pre-trained model used, for each configuration (Corpus, Data used for training, Model), it became evident that using **HATEBERTimbau** leads to better results than using **BERTimbau**. In 73 of the 90 configurations (81.1%), 36 with CO-HATE corpus and 37 with FIGHT corpus, the Positive Class F1-score achieved by **HATEBERTimbau** was higher than the one obtained by **BERTimbau**. This was an expected result because **HATEBERTimbau** was adapted to social media language that includes potential HS targeting African descent, Roma, and LGBTQI+. We were expecting to achieve better results in FIGHT corpus since **HATEBERTimbau** was trained with Twitter data and FIGHT is composed by Twitter data, but the performance was very similar to the one obtained with CO-HATE corpus. Although, **BERTimbau** could still outperform **HATEBERTimbau** in 17 scenarios, the best results obtained with CO-HATE corpus and FIGHT corpus were achieved with **HATEBERTimbau**.

Comparing the single-task models with the multitask models, the multitask architecture showed interesting results for some configurations (Corpus, Data used for training, BERT pre-trained model). In 25 of the 36 configurations (69.4%), the Positive Class F1-score achieved by the best multitask model was higher than the one obtained by BERT-Attention, the equivalent single-task architecture. This suggests that the multitask model can take advantage of the knowledge it has about the other class to detect the HS. In fact, in 8 of these 25 scenarios, the multitask model outperformed the best single-task model (for CO-HATE corpus it achieved the best result of all experiments), demonstrating the great potential of this multitask architecture. We also inspected these 25 scenarios to see which of the two classes (Counterspeech or Sentiment Polarity) allowed to achieve the best result. With CO-HATE, in 7 of the 11 scenarios (63.6%), Counterspeech was that class and with FIGHT, Counterspeech was that class in 7 of the 14 scenarios (50%). These results reveal that the leveraging of both classes (Counterspeech and Sentiment Polarity) for both corpora is relevant for the detection of HS.

The best result with CO-HATE corpus (Positive Class F1-score of 0.727) was achieved with the data of A+B+C+D+E, BERT-Attention-CS and **HATEBERTimbau**. We should also mention that the second best result, achieved with the data of E, BERT-CNN and **HATEBERTimbau**, had a pretty close result (Positive Class F1-score of 0.724) proving that transfer learning approaches can achieve good results without requiring large amounts of data. In this case, it was used five times fewer data and a similar performance was obtained.

The best result with FIGHT corpus (Positive Class F1-score of 0.724) was achieved with the data of A+B+C+D+E, BERT-LinearLayer and **HATEBERTimbau**.

We consider the best results to be quite fair, especially considering the low IAA obtained between all

the annotators on both corpora (0.478 on CO-HATE and 0.362 on FIGHT), which showed the difficulty of this task. The result obtained with FIGHT corpus is even more surprising since it has a much lower IAA than CO-HATE corpus and manages to obtain a similar performance. Also, even though the dummy classifier may not be the best model for comparison, the fact that it was outperformed by 9% in CO-HATE and by 30% in FIGHT proves that our models were able to learn about the OHS phenomenon.

7

Conclusion

Contents

7.1 Conclusions 75
7.2 Limitations and Future Work 76

In this chapter we present the main conclusions of this work, as well as its limitations and the directions for future work.

7.1 Conclusions

The proliferation of HS on social media needs to be filtered and automated tools are required. In the past years, DL approaches, for both feature extraction and training classifiers, have been the trend methods and have reached new state-of-the-art results. However, there are not much OHS detection works focusing on the European Portuguese language and it is not known how the most popular and successful methods being used for other languages perform in this particular language. To fill this gap, we explored different models that proved to be successful in the literature to address this task with two different European Portuguese corpora created recently: CO-HATE and FIGHT. Both corpora focus on the expression of HS by the Portuguese online community against the Afro-descendant, Roma, and LGBTQI+ communities, in particular, CO-HATE is composed by YouTube comments and FIGHT by Twitter posts. For each dataset we conducted several experiments to detect HS and also assessed how different factors affect the performance of that detection. We have tested different models for this task, all of them based on Transfer Learning, in particular, based on the existing BERT-like pre-trained model **BERTimbau**. We also developed **HateBERTimbau**, a retrained version of **BERTimbau** that is adapted to social media language including potential HS and targeting African descent, Roma, and LGBTQI+ communities. BERT-CNN, BERT-LinearLayer, and BERT-Attention models perform only the HS detection task while, in an attempt to leverage other aspects of the messages to benefit our main task, BERT-Attention-CS performs also the Counterspeech detection task and BERT-Attention-SNT performs also the Sentiment Polarity detection task. CO-HATE and FIGHT corpora have been manually labelled by five different annotators (regarding the presence of HS and other categories) and we have assessed the impact on the performance of OHS detection of using different subsets of annotations for the training of the models. We also tested if by having models that detect HS against a particular target group, we would achieve better results than having a general HS detection model. The results confirm that further pre-training ported **BERTimbau** to other language varieties. **HATEBERTimbau** consistently outperformed **BERTimbau** on both datasets and the best results were obtained with it. The learning of another task added to the HS detection demonstrated to be helpful in obtaining better results. Either Counterspeech information or Sentiment Polarity information allowed to achieve better results in diverse experiments. Comparing different subsets of data used for the training of the models, it was shown that, in general, a higher agreement on the data leads to better results. For the CO-HATE corpus, the target-specific model performance was equivalent to the performance of the generic model while for the FIGHT corpus the performance of the target-specific model was 5% better, revealing the potential of

having target-specific models. The error analysis performed based on both datasets showed that a great percentage of the messages misclassified as Non-HS would not be misclassified considering the view of the majority of the annotators. A high proportion of the messages misclassified as HS corresponds, in fact, to Counterspeech or Offensive Speech, and contain words that are frequently used in hatred content. We consider that our models were able to learn about the OHS phenomenon in Portuguese. The best result obtained in CO-HATE corpus (Positive Class F1-score of 0.727) and the best result obtained in FIGHT corpus (Positive Class F1-score of 0.724) were good performances considering the difficulty of this task, demonstrated by the low IAA obtained between all the annotators on both corpora.

7.2 Limitations and Future Work

As limitations, we believe the models started associating to HS the presence of specific words in the messages, which led to misclassifications, as it was shown in our error analysis. According to the definition we used, any message that supports hatred is considered HS. Concerning the CO-HATE corpus, the annotators performed their annotations considering the context of the conversation which allowed them to have a bigger picture of what the messages convey. Our models otherwise only "saw" the messages as they are, which made their task difficult and led to wrongly classified messages.

In terms of future directions, considering that a lot of misclassifications were Counterspeech or Offensive Speech, we believe that introducing these classes in the classification could be beneficial. For this purpose it would be important to have a bigger representation of these classes. It would be also interesting to explore strategies to provide the models the message context in order to capture semantic relations that require information not expressed in text and reduce the misclassifications.

Bibliography

[1] D. Ferreira, S. Silva, A. Abelha, and J. Machado, "Recommendation system using autoencoders," *Applied Sciences*, vol. 10, no. 16, 2020.

[2] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, "Text classification algorithms: A survey," *CoRR*, vol. abs/1904.08067, 2019. [Online]. Available: http://arxiv.org/abs/1904.08067

[3] M. Hagiwara, *Real-World Natural Language Processing: Practical Applications with Deep Learning*. Manning, 2021. [Online]. Available: https://books.google.pt/books?id=A92_zQEACAAJ

[4] M. S. Jahan and M. Oussalah, "A systematic review of hate speech automatic detection using natural language processing," *CoRR*, vol. abs/2106.00742, 2021. [Online]. Available: https://arxiv.org/abs/2106.00742

[5] M. Mozafari, R. Farahbakhsh, and N. 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, Lisbon, Portugal, December 10-12, 2019*, ser. Studies in Computational Intelligence, H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, and L. M. Rocha, Eds., vol. 881. Springer, 2019, pp. 928–940.

[6] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 2054–2059. [Online]. Available: https://www.aclweb.org/anthology/2020.semeval-1.271

[7] N. S. Samghabadi, P. Patwa, S. PYKL, P. Mukherjee, A. Das, and T. Solorio, "Aggression and misogyny detection using BERT: A multi-task approach," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020*, R. Kumar, A. K. Ojha, B. Lahiri, M. Zampieri, S. Malmasi, V. Murdock, and D. Kadar, Eds.

European Language Resources Association (ELRA), 2020, pp. 126–131. [Online]. Available: https://aclanthology.org/2020.trac-1.20/

[8] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Comput. Surv.*, vol. 51, p. 85:1–85:30, 2018.

[9] K. Müller and C. Schwarz, "Fanning the flames of hate: Social media and hate crime," *Journal of the European Economic Association*, vol. 19, pp. 2131–2167, 8 2021.

[10] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, and V. Patti, "Resources and benchmark corpora for hate speech detection: a systematic review," *Language Resources and Evaluation*, vol. 55, pp. 477–523, 2021.

[11] G. Kovács, P. Alonso, and R. Saini, "Challenges of hate speech detection in social media," *SN Computer Science*, vol. 2, no. 2, Feb. 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00457-3

[12] P. Carvalho, D. Caled, C. Silva, F. Batista, and R. Ribeiro, "The expression of Hate Speech against Afro-descendant, Roma and LGBTQ+ communities in YouTube comments," *submitted*, 2022.

[13] F. Baider, "Covert hate speech, conspiracy theory and anti-semitism: Linguistic analysis versus legal judgement," *International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique*, pp. 1–25, 2022.

[14] H. Al Kuwatly, M. Wich, and G. Groh, "Identifying and measuring annotator bias based on annotators' demographic characteristics," in *Proceedings of the Fourth Workshop on Online Abuse and Harms*. Online: Association for Computational Linguistics, Nov. 2020, pp. 184–190. [Online]. Available: https://aclanthology.org/2020.alw-1.21

[15] C. of Europe, "Portugal should act more resolutely to tackle racism and continue efforts to combat violence against women," Jun 2021. [Online]. Available: https://www.coe.int/en/web/commissioner/-/portugal-should-act-more-resolutely-to-tackle-racism-and-continue-efforts-to-combat-violence\-against-women

[16] P. Fortuna, J. Rocha da Silva, J. Soler-Company, L. Wanner, and S. Nunes, "A hierarchically-labeled Portuguese hate speech dataset," in *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 94–104. [Online]. Available: https://aclanthology.org/W19-3510

[17] E. U. Institute., *Monitoring media pluralism in the digital era: application of the media pluralism monitor in the European Union, Albania and Turkey in the years 2018 2019: country report Portugal.* Publications Office, 2020. [Online]. Available: https://data.europa.eu/doi/10.2870/292300

[18] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep learning for hate speech detection in tweets,” in *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, Eds. ACM, 2017, pp. 759–760.

[19] B. Gambäck and U. K. Sikdar, “Using convolutional neural networks to classify hate-speech,” in *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*, Z. Waseem, W. H. K. Chung, D. Hovy, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2017, pp. 85–90.

[20] S. Kamble and A. Joshi, “Hate speech detection from code-mixed hindi-english tweets using deep learning models,” *CoRR*, vol. abs/1811.05145, 2018. [Online]. Available: http://arxiv.org/abs/1811.05145

[21] R. Ong, “Offensive language analysis using deep learning architecture,” *CoRR*, vol. abs/1903.05280, 2019. [Online]. Available: http://arxiv.org/abs/1903.05280

[22] T. Ranasinghe, M. Zampieri, and H. Hettiarachchi, “BRUMS at HASOC 2019: Deep learning models for multilingual hate speech and offensive language identification,” in *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, ser. CEUR Workshop Proceedings, P. Mehta, P. Rosso, P. Majumder, and M. Mitra, Eds., vol. 2517. CEUR-WS.org, 2019, pp. 199–207. [Online]. Available: http://ceur-ws.org/Vol-2517/T3-3.pdf

[23] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, “Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval),” *CoRR*, vol. abs/1903.08983, 2019. [Online]. Available: http://arxiv.org/abs/1903.08983

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[25] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate speech detection with comment embeddings,” in *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, A. Gangemi, S. Leonardi, and A. Panconesi, Eds. ACM, 2015, pp. 29–30.

[26] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” in *Proceedings of the Eleventh International Conference on Web*

and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017. AAAI Press, 2017, pp. 512–515. [Online]. Available: https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15665

[27] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, p. 150, 2019.

[28] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*. Springer, 2012, pp. 163–222.

[29] C. Nobata, J. R. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks, and B. Y. Zhao, Eds. ACM, 2016, pp. 145–153.

[30] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13 825–13 835, 2018.

[31] R. Kshirsagar, T. Cukuvac, K. R. McKeown, and S. McGregor, "Predictive embeddings for hate speech detection on twitter," *CoRR*, vol. abs/1809.10644, 2018. [Online]. Available: http://arxiv.org/abs/1809.10644

[32] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying crisp-dm process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021.

[33] A. Samuel, "Eight-move opening utilizing generalization learning," *IBM J*, vol. 3, no. 3, pp. 210–229, 1959.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[35] A. Ramponi and B. Plank, "Neural unsupervised domain adaptation in NLP - A survey," *CoRR*, vol. abs/2006.00632, 2020. [Online]. Available: https://arxiv.org/abs/2006.00632

[36] R. Mallett, J. Hagen-Zanker, R. Slater, and M. Duvendack, "The benefits and challenges of using systematic reviews in international development research," *Journal of Development Effectiveness*, vol. 4, pp. 445–455, 9 2012.

[37] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue,*

Washington, USA, M. desJardins and M. L. Littman, Eds. AAAI Press, 2013. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6419

[38] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proceedings of the Second Workshop on Language in Social Media*. Association for Computational Linguistics, 6 2012, pp. 19–26. [Online]. Available: https://aclanthology.org/W12-2103

[39] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *2012 International Conference on Privacy, Security, Risk and Trust, PASSAT 2012, and 2012 International Confernece on Social Computing, SocialCom 2012, Amsterdam, Netherlands, September 3-5, 2012*. IEEE Computer Society, 2012, pp. 71–80.

[40] P. Burnap and M. L. Williams, "Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making," in *Internet, Policy & Politics*, 2014. [Online]. Available: http://orca.cardiff.ac.uk/id/eprint/65227

[41] S. Agarwal and A. Sureka, "Using knn and svm based one-class classifier for detecting online radicalization on twitter," in *Distributed Computing and Internet Technology - 11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings*, R. Natarajan, G. Barua, and M. R. Patra, Eds., vol. 8956. Springer, 2015, pp. 431–442.

[42] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[43] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. The Association for Computational Linguistics, 2016, pp. 88–93.

[44] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," *CoRR*, vol. abs/1712.06427, 2017. [Online]. Available: http://arxiv.org/abs/1712.06427

[45] ——, "Challenges in discriminating profanity from hate speech," *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 2, pp. 187–202, 2018.

[46] V. Nahar, S. Al-Maskari, X. Li, and C. Pang, "Semi-supervised learning for cyberbullying detection in social networks," in *Databases Theory and Applications - 25th Australasian Database Conference, ADC 2014, Brisbane, QLD, Australia, July 14-16, 2014. Proceedings*, H. Wang and M. A. Sharaf, Eds., vol. 8506. Springer, 2014, pp. 160–171.

[47] M. D. Capua, E. D. Nardo, and A. Petrosino, "Unsupervised cyber bullying detection in social networks," in *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*. IEEE, 2016, pp. 432–437.

[48] N. D. Gitari, Z. Zhang, H. Damien, and J. Long, "A lexicon-based approach for hate speech detection," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, pp. 215–230, 4 2015.

[49] M. Wiegand, J. Ruppenhofer, A. Schmidt, and C. Greenberg, "Inducing a lexicon of abusive words - a feature-based approach," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 1046–1056.

[50] M. Dadvar and K. Eckert, "Cyberbullying detection in social networks using deep learning based models; A reproducibility study," *CoRR*, vol. abs/1812.08046, 2018. [Online]. Available: http://arxiv.org/abs/1812.08046

[51] G. I. Sigurbergsson and L. Derczynski, "Offensive language and hate speech detection for danish," *CoRR*, vol. abs/1908.04531, 2019. [Online]. Available: http://arxiv.org/abs/1908.04531

[52] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on twitter," *CoRR*, vol. abs/1706.01206, 2017. [Online]. Available: http://arxiv.org/abs/1706.01206

[53] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016. [Online]. Available: http://arxiv.org/abs/1607.01759

[54] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Effective hate-speech detection in twitter data using recurrent neural networks," *Appl. Intell.*, vol. 48, no. 12, pp. 4730–4742, 2018.

[55] S. Zimmerman, U. Kruschwitz, and C. Fox, "Improving hate speech detection with deep learning ensembles," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. European Language Resources Association (ELRA), 2018. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2018/summaries/292.html

[56] Z. Zhang, D. Robinson, and J. A. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, ser. Lecture Notes in Computer Science,

A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., vol. 10843. Springer, 2018, pp. 745–760.

[57] T. Van Huynh, V. D. Nguyen, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate speech detection on vietnamese social media text using the bi-gru-lstm-cnn model," *CoRR*, vol. abs/1911.03644, 2019, withdrawn. [Online]. Available: http://arxiv.org/abs/1911.03644

[58] P. Kapil, A. Ekbal, and D. Das, "Investigating deep learning approaches for hate speech detection in social media," *CoRR*, vol. abs/2005.14690, 2020. [Online]. Available: https://arxiv.org/abs/2005.14690

[59] H. Rosa, D. M. de Matos, R. Ribeiro, L. Coheur, and J. P. Carvalho, "A "deeper" look at detecting cyberbullying in social networks," in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*. IEEE, 2018, pp. 1–8.

[60] G. Kovács, P. Alonso, and R. Saini, "Challenges of hate speech detection in social media," *SN Comput. Sci.*, vol. 2, no. 2, p. 95, 2021.

[61] P. Liu, W. Li, and L. Zou, "NULI at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers," in *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, J. May, E. Shutova, A. Herbelot, X. Zhu, M. Apidianaki, and S. M. Mohammad, Eds. Association for Computational Linguistics, 2019, pp. 87–91.

[62] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin, "Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020)," *CoRR*, vol. abs/2006.07235, 2020. [Online]. Available: https://arxiv.org/abs/2006.07235

[63] T. Mandl, S. Modha, A. K. M, and B. R. Chakravarthi, "Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german," in *FIRE 2020: Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020*, P. Majumder, M. Mitra, S. Gangopadhyay, and P. Mehta, Eds. ACM, 2020, pp. 29–32.

[64] R. Raja, S. Srivastavab, and S. Saumyac, "Nsit & iiitdwd@ hasoc 2020: Deep learning model for hate-speech identification in indo-european languages," in *Working notes of 2020 FIRE – CEUR workshop proceedings*, 2021, pp. 161–167. [Online]. Available: http://ceur-ws.org/Vol-2826/T2-8.pdf

[65] R. Kumar, B. Lahiri, A. K. Ojha, and A. Bansal, "Comma@ fire 2020: Exploring multilingual joint training across different classification tasks." in *Working notes of 2020 FIRE – CEUR workshop proceedings*, 2020, pp. 823–828. [Online]. Available: http://ceur-ws.org/Vol-2826/T10-3.pdf

[66] A. K. Mishraa, S. Saumyab, and A. Kumara, "Iiit_dwd@ hasoc 2020: Identifying offensive content in indo-european languages," in *Working notes of 2020 FIRE – CEUR workshop proceedings*, 2020, pp. 139–144. [Online]. Available: http://ceur-ws.org/Vol-2826/T2-5.pdf

[67] E. Lavergne, R. Saini, G. Kovács, and K. Murphy, "Thenorth@ haspeede 2: Bert-based language model fine-tuning for italian hate speech detection," in *Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. RWTH Aachen University, 2020. [Online]. Available: http://ceur-ws.org/Vol-2765/paper135.pdf

[68] and European University Institute, F. Rui Nunes Cádima, C. Baptista, L. Oliveira Martins, M. Torres da Silva, and R. Lourenço, *Monitoring media pluralism in the digital era : application of the media pluralism monitor in the European Union, Albania and Turkey in the years 2018-2019 : country report Portugal*. European University Institute, 2020.

[69] K. Krippendorff, "Measuring the reliability of qualitative text analysis data," *Quality and Quantity*, vol. 38, no. 6, pp. 787–800, Dec 2004. [Online]. Available: https://doi.org/10.1007/s11135-004-8107-7

[70] P. Carvalho, B. Matos, R. Santos, F. Batista, and R. Ribeiro, "Hate Speech Dynamics Against African descent, Roma and LGBTQI Communities in Portugal," *LREC*, 2022.

[71] F. Souza, R. Nogueira, and R. Lotufo, "Bertimbau: pretrained bert models for brazilian portuguese," in *Brazilian Conference on Intelligent Systems*. Springer, 2020, pp. 403–417.

[72] J. A. Wagner Filho, R. Wilkens, M. Idiart, and A. Villavicencio, "The brWaC corpus: A new open resource for Brazilian Portuguese," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1686

[73] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, "HateBERT: Retraining BERT for abusive language detection in English," in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 17–25. [Online]. Available: https://aclanthology.org/2021.woah-1.3

[74] B. C. Matos, R. B. Santos, P. Carvalho, R. Ribeiro, and F. Batista, "Comparing Different Approaches for Detecting Hate Speech in Online Portuguese Comments," in *11th Symposium on Languages, Applications and Technologies (SLATE 2022)*, ser. Open Access Series in Informatics (OASIcs),

J. a. Cordeiro, M. J. a. Pereira, N. F. Rodrigues, and S. a. Pais, Eds., vol. 104. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 10:1–10:12. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2022/16756

[75] A. Schmidt and M. 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*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1–10. [Online]. Available: https://aclanthology.org/W17-1101