MUDES: Multilingual Detection of Offensive Spans

Tharindu Ranasinghe
University of Wolverhampton
Wolverhampton, UK
tharindu.ranasinghe@wlv.ac.uk

Marcos Zampieri
Rochester Institute of Technology
Rochester, NY, USA
mazgla@rit.edu

Abstract

The interest in offensive content identification in social media has grown substantially in recent years. Previous work has dealt mostly with post level annotations. However, identifying offensive spans is useful in many ways. To help coping with this important challenge, we present MUDES, a multilingual system to detect offensive spans in texts. MUDES features pre-trained models, a Python API for developers, and a user-friendly web-based interface. A detailed description of MUDES' components is presented in this paper.

1 Introduction

Offensive and impolite language are widespread in social media posts motivating a number of studies on automatically detecting the various types of offensive content (e.g. aggression (Kumar et al., 2018, 2020), cyber-bullying (Rosa et al., 2019), hate speech (Malmasi and Zampieri, 2018), etc.). Most previous work has focused on classifying full instances (e.g. posts, comments, documents) (e.g. offensive vs. not offensive) while the identification of the particular spans that make a text offensive has been mostly neglected.

Identifying offensive spans in texts is the goal of the SemEval-2021 Task 5: Toxic Spans Detection (Pavlopoulos et al., 2021). The organisers of this task argue that highlighting toxic spans in texts helps assisting human moderators (e.g. news portals moderators) and that this can be a first step in semi-automated content moderation. Finally, as we demonstrate in this paper, addressing offensive spans in texts will make the output of offensive language detection systems more interpretable thus allowing a more detailed linguistics analysis of predictions and improving the quality of such systems.

With these important points in mind, we developed MUDES: **Mu**ltilingual **De**tection of Offensive **S**pans. MUDES is a multilingual framework for offensive language detection focusing on text spans. The main contributions of this paper are the following:

1. We introduce MUDES, a new Python-based framework to identify offensive spans with state-of-the-art performance.

2. We release four pre-trained offensive language identification models: en-base, en-large models which are capable of identifying offensive spans in English text. We also release Multilingual-base and Multilingual-large models which are able to recognise offensive spans in languages other than English.

3. We release a Python Application Programming Interface (API) for developers who are interested in training more models and performing inference in the code level.

4. For general users and non-programmers, we release a user-friendly web-based User Interface (UI), which provides the functionality to input a text in multiple languages and to identify the offensive span in that text.

2 Related Work

Early approaches to offensive language identification relied on traditional machine learning classifiers (Dadvar et al., 2013) and later on neural networks combined with word embeddings (Majumder et al., 2018; Hettiarachchi and Ranasinghe, 2019). Transformer-based models like BERT (Devlin et al., 2019) and ELMO (Peters et al., 2018) have been recently applied to offensive language detection achieving competitive scores (Wang et al., 2020; Ranasinghe and Hettiarachchi, 2020) in recent SemEval competitions such as HatEval (Basile et al., 2019) OffensEval (Zampieri et al., 2020).

In terms of languages, the majority of studies on this topic deal with English (Malmasi and Zampieri,

WARNING: This paper contains text excerpts and words that are offensive in nature.

Proceedings of NAACL-HLT 2021: Demonstrations, pages 144–152
June 6–11, 2021.

Post	Offensive Spans
Stupid hatcheries have completely fucked everything	[0, 1, 2, 3, 4, 5, 34, 35, 36, 37, 38, 39]
Victimitis: You are such an asshole.	[28, 29, 30, 31, 32, 33, 34]
So is his mother. They are silver spoon parasites.	[]
You're just silly.	[12, 13, 14, 15, 16]

Table 1: Four comments from the dataset, with their annotations. The offensive words are displayed in red and the spans are indicated by the character position in the instance.

2017; Yao et al., 2019; Ridenhour et al., 2020; Rosenthal et al., 2020) due to the the wide availability of language resources such as corpora and pre-trained models. In recent years, several studies have been published on identifying offensive content in other languages such as Arabic (Mubarak et al., 2020), Dutch (Tulkens et al., 2016), French (Chiril et al., 2019), Greek (Pitenis et al., 2020), Italian (Poletto et al., 2017), Portuguese (Fortuna et al., 2019), and Turkish (Çöltekin, 2020). Most of these studies have created new datasets and resources for these languages opening avenues for multilingual models as those presented in Ranasinghe and Zampieri (2020). However, all studies presented in this section focused on classifying full texts, as discussed in the Introduction. MUDES' objective is to fill this gap and perform span level offensive language identification.

3 Data

The main dataset used to train the machine learning models presented in this paper is the dataset released within the scope of the aforementioned SemEval-2021 Task 5: Toxic Spans Detection for English. The dataset contains posts (comments) from the publicly available Civil Comments dataset (Borkan et al., 2019). The organisers have randomly selected 10,000 posts, out of a total of 1,2 million posts in the original dataset. The offensive spans have been annotated using a crowd-annotation platform, employing three crowd-raters per post. By the time of writing this paper, only the trial set and the training set have been released and the gold labels for the test set have not yet been released. Therefore, training of the machine learning models presented in MUDES was done on the training set which we refer to as *TSDTrain* and the evaluation was conducted on the trial set which we refer to as *TSDTrial* set. In Table 1 we show four randomly selected examples from the *TSDTrain* dataset with their annotations.

The general idea is to learn a robust model from this dataset and generalize to other English datasets which do not contain span annotation. Another goal is to investigate the feasibility of annotation projection to other languages.

Other Datasets In order to evaluate our framework in different domains and languages we used three publicly available offensive language identification datasets. As an off-domain English dataset, we choose the Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a), used in OffensEval 2019 (SemEval-2019 Task 6) (Zampieri et al., 2019b), containing over 14,000 posts from Twitter. To evaluate our framework in different languages, we selected a Danish (Sigurbergsson and Derczynski, 2020) and a Greek (Pitenis et al., 2020) dataset. These two datasets have been provided by the organisers of OffensEval 2020 (SemEval-2020 Task 12) (Zampieri et al., 2020) and were annotated using OLID's annotation guidelines. The Danish dataset contains over 3,000 posts from Facebook and Reddit while the Greek dataset contains over 10,000 Twitter posts, allowing us to evaluate our dataset in an off-domain, multilingual setting. As these three datasets have been annotated at the instance level, we followed an evaluation process explained in Section 5.

4 Methodology

The main motivation behind this methodology is the recent success that transformer models had in various NLP tasks (Devlin et al., 2019) including offensive language identification (Ranasinghe and Zampieri, 2020; Ranasinghe et al., 2019; Wiedemann et al., 2020). Most of these transformer-based approaches take the final hidden state of the first token ([CLS]) from the transformer as the representation of the whole sequence and a simple softmax classifier is added to the top of the transformer model to predict the probability of a class label (Sun et al., 2019). However, as previously men-

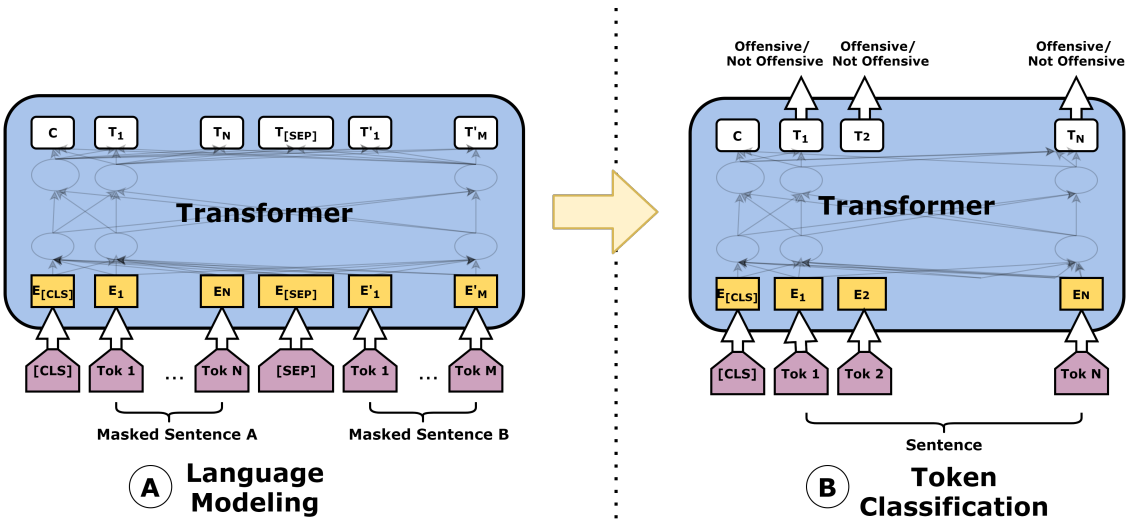


Figure 1: Model Architecture. Architecture consists of two parts. Part A is the language modelling and Part B is the token classification.

tioned, these models classify whole comments or documents and do not identify the spans that make a text offensive. Since the objective of this task is to identify offensive spans rather than classifying the whole comment, we followed a different architecture.

As shown in Figure 1, the complete architecture contains two main parts; Language Modeling (LM) and Token Classification (TC). In the LM part, we used a pre-trained transformer model and retrained it on the *TSDTrain* dataset using Masked Language Modeling (MLM). In the second part of the architecture, we used the saved model from the LM part and we perform a token classification. We added a token level classifier on top of the transformer model as shown in Figure 1. The token-level classifier is a linear layer that takes the last hidden state of the sequence as the input and produce a label for each token as the output. In this case each token can have two labels; offensive and not offensive. We have listed the training configurations in the Appendix.

We experimented with several popular transformer models like BERT (Devlin et al., 2019), XLNET (Yang et al., 2019), ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019) etc. From the pre-trained transformer models we selected, we grouped the large models and base models separately in order to release two English models. A large model; en-large which is more accurate, but has a low efficiency regarding space and time. The base model; en-base is efficient, but has a comparatively low accuracy than the en-large model. All the experiments have been executed for five times with different random seeds and we took the mode of the classes predicted by each random seed as the final result (Hettiarachchi and Ranasinghe, 2020).

Multilingual models - The motivation behind the use of multilingual models comes from recent works (Ranasinghe and Zampieri, 2020, 2021; Ranasinghe et al., 2020) which used transfer learning and cross-lingual embeddings. These studies show that cross-lingual transformers like XLM-R (Conneau et al., 2019) can be trained on an English dataset and have the model weights saved to detect offensive language in other languages outperforming monolingual models trained on the target language dataset. We used a similar methodology but for the token classification architecture instead. We used XLM-R cross-lingual transformer model (Conneau et al., 2019) as the Transformer in Figure 1 on *TSDTrain* and carried out evaluations on the Danish and Greek datasets. We release two multilingual models; multilingual-base based on XLM-R base model and multilingual-large based on XLM-R large model.

5 Evaluation and Results

We followed two different evaluation methods. In Section 5.1 we present the methods used to evaluate offensive spans on the TSDTrial set. In Section 5.2 we presented the methods used to evaluate the other three datasets which only contained post level annotations.

5.1 Offensive Spans Evaluation

For the Toxic Spans Detection dataset, we followed the same evaluation procedure of the SemEval Toxic Spans Detection competition. The organisers have used F1 score mentioned in Da San Martino et al. (2019) to evaluate the systems. Let system A_i return a set $S^t_{A_i}$ of character offsets, for parts of the post found to be toxic. Let G_t be the character offsets of the ground truth annotations of t. We compute the F1 score of system A_i with respect to the ground truth G for post t as mentioned in Equation 1, where $|\cdot|$ denotes set cardinality.

$$F_1^t(A_i, G) = \frac{2 \cdot P^t(A_i, G) \cdot R^t(A_i, G)}{P^t(A_i, G) + R^t(A_i, G)} \quad (1)$$

$$P^t(A_i, G) = \frac{\left|S^t_{A_i} \cap S^t_G\right|}{\left|S^t_{A_i}\right|} \; R^t(A_i, G) = \frac{\left|S^t_{A_i} \cap S^t_G\right|}{\left|S^t_G\right|}$$

We present the results along with the baseline provided by the organisers in Table 2. The baseline is implemented using a spaCy NER pipeline. The spaCy NER system contains a word embedding strategy using sub word features and Bloom embedding (Serrà and Karatzoglou, 2017), and a deep convolution neural network with residual connections. Additionally, we compare our results to a lexicon-based word match approach mentioned in Ranasinghe et al. (2021) where the lexicon is based on profanity words from online resources[1,2].

Model Name	Base Model	F1 score
en-large	roberta-large	0.6886
en-base	xlnet-base-cased	0.6734
multilingual-large	XLM-R-large	0.6338
multilingual-base	XLM-R-base	0.6160
spaCy baseline	NA	0.5976
Lexicon word match (Ranasinghe et al., 2021)	NA	0.3378

Table 2: Results ordered by F1 score for TSD Trial.

The results show that all MUDES' models outperform the spaCy baseline and the lexicon-based word match. From all of the large transformer models we experimented roberta-large performed better than others. Therefore, we released it as en-large model in MUDES. From the base models we experimented, XLNet-base-cased model outperformed all the other base models so we released it as en-base model. We also released two multilingual models; multilingual-base and multilingual-large based on XLM-R-base and XLM-R-large respectively. All the pre-trained MUDES' models are available to download from HuggingFace model hub [3] (Wolf et al., 2020).

5.2 Off-Domain and Multilingual Evaluation

For the English off-domain and multilingual datasets we followed a different evaluation process. We used a pre-trained MUDES' model trained on *TSDTrain* to predict the offensive spans for all texts in the test sets of two non-English datasets (Danish, and Greek) and English off-domain dataset, OLID, which is annotated at the document level. If a certain text contains at least one offensive span we marked the whole text as offensive following the OLID annotation guidelines described in Zampieri et al. (2019a). We compared our results to the best systems submitted to OffensEval 2020 in terms of macro F1 reported by the task organisers (Zampieri et al., 2020). We present the results along with the majority class baseline for each dataset in Table 3. For English off domain dataset (OLID) we only used the MUDES en models while for Danish and Greek datasets we used the MUDES multilingual models.

Language	Model	M F1
Danish	Pàmies et al. (2020)	0.8119
	multilingual-large	0.7623
	multilingual-base	0.7143
	Majority Baseline	0.4668
English	Wiedemann et al. (2020)	0.9204
	en-large	0.9023
	en-base	0.8892
	Majority Baseline	0.4193
Greek	Ahn et al. (2020)	0.8522
	multilingual-large	0.8143
	multilingual-base	0.7820
	Majority Baseline	0.4202

Table 3: Results ordered by macro (M) F1 for Danish, English and Greek datasets

Results show that despite the change of domain and the language, MUDES perform well in all the datasets and compares favourably to the best systems submitted. It should be noted that the best

[1] https://www.cs.cmu.edu/~biglou/resources/bad-words.txt

[2] https://github.com/RobertJGabriel/Google-profanity-words

[3] MUDES' models are available on https://huggingface.co/mudes

systems have been predominantly trained on offensive languages identification task on post level while MUDES' objective is different. Yet MUDES come closer to the best systems in all the datasets.

From the results, it is clear that MUDES english models can perform in a different domain like Twitter. Also the results show that MUDES multilingual models are capable of identifying offensive spans in other languages too. Since XLM-R supports 104 languages, this approach will benefit all those languages without any training data at all.

6 System Demonstration

6.1 Application Programming Interface

MUDES is available as a Python package in the Python Package Index (PyPI)[4]. The package is related to MUDES GitHub repository[5]. Users can install it easily with the following command after installing PyTorch (Paszke et al., 2019).

```
$ pip install mudes
```

The Python package contains the following functionalities.

Get offensive spans with a pretrained model The library provides the functionality to load a pretrained model and use it to identify offensive spans. The following code segment downloads and loads MUDES' en-base model in a CPU only environment and identifies offensive spans in the the text; *"This is fucking crazy!!"*. If the users prefer a GPU, the argument *use_cuda* should be set to True.

Listing 1 English Inference Example

```
from mudes.app.mudes_app
                  import MUDESApp

sentence = "This is fucking crazy!!"

app = MUDESApp("en-base",
                  use_cuda=False)
app.predict_toxic_spans(sentence)
```

Train a MUDES model The library provides the functionality to train a MUDES model from scratch using the code segment present next. It takes a Pandas dataframe in the format of *TSDTrain*, formats it for the token classification task and train a MUDES model from scratch. MUDES support popular transformer types as bert, xlnet, roberta etc. as the MODEL_TYPE and name of the model as appear in Hugging Face (Wolf et al., 2020) model repository. [6]

Listing 2 Training Example

```
from mudes.algo.mudes_model
                  import MUDESModel
from mudes.algo.preprocess
                  import read_datafile,
                         format_data

train_df = format_data(train)
tags = train_df['labels']
             .unique().tolist()

model = MUDESModel(MODEL_TYPE,
             MODEL_NAME, labels=tags)
model.train(train_df)
```

6.2 User Interface

We developed a prototype of the User Interface (UI) to demonstrate the capabilities of the system. The UI is based on Streamlit[7] which provides functionalities to easily develop dashboards for machine learning projects. The code base for the UI is available in GitHub [8]. This UI is hosted in a Linux server. [9] We also release a Docker container image of the UI in Docker Hub[10] for those who are interested in self hosting the UI. Docker enables developers to easily deploy and run any application as a lightweight, portable, self-sufficient container, which can run virtually anywhere. The released Docker container image follows Continuous Integration/Continuous Deployment (CI/CD) from the GitHub repository which allows sharing and deploying the code quickly and efficiently.

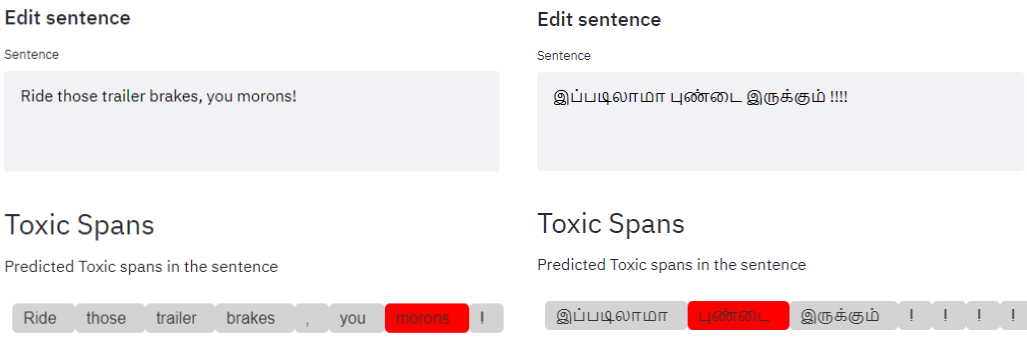
Once Docker is installed, one can easily run our UI with this command.

```
$ docker run tharindudr/mudes
```

This command will automatically install all the required packages, download and load the pre-trained models and open the system in the default browser. We provide the following functionalities from the user interface.

Switch through pretrained models - The users can switch through the pre-trained models using the radio buttons available in the left side of the UI under Available Models section. They can select

[4] https://pypi.org/project/mudes/
[5] https://github.com/tharindudr/MUDES
[6] https://huggingface.co/models
[7] www.streamlit.io
[8] https://github.com/tharindudr/MUDES-UI.
[9] http://rgcl.wlv.ac.uk/mudes/
[10] Docker Hub is a hosted repository service provided by Docker for finding and sharing container images.



(a) Example from Civil Comments Dataset

(b) Example from Tamil. (Pussy like this !!!!)

Figure 2: Examples in English and in a low-resource languages. The experiments were conducted with en-large and the multilingual-large models respectively.

an option from en-base, en-large, multilingual-base and multilingual-large. These models have been already downloaded from the HuggingFace model hub and they are loaded in to the random-access memory of the hosting computer.

Switch through available datasets - We have made the four datsets used in this paper available from the UI for the users to experiment with (Borkan et al., 2019; Zampieri et al., 2019a; Pitenis et al., 2020; Sigurbergsson and Derczynski, 2020). Once the user selects a particular option, the system will automatically load the test set of the selected dataset. Once it is loaded the user can iterate through the dataset using the scrollbar. For each text the UI will display the offensive spans in red.

Get offensive spans for a custom text - The users can also enter a custom text in the text box, hit ctrl+enter and see the offensive spans available in the input text. Once processed through the system, any offensive spans available in the text will be displayed in red. Figure 2 shows several screenshots from the UI. It illustrates an example on English for the texts taken from civil comments dataset (Borkan et al., 2019) conducted with en-large model. To show that MUDES framework works on low resource language too, Figure 2 also displays an example from Tamil.

6.3 System Efficiency

The time taken to predict the offensive spans for a text will be critical in an online system developed for real time use. Therefore, we evaluated the time MUDES takes to predict the offensive spans in 100 texts for all the released models in a CPU and GPU environment. The results show that large models take around 3 seconds for a sentence in a CPU and take around 1 second for a sentence in a GPU on average while the base models take approximately one third of that time in both environments. From these results it is clear that MUDES is capable of predicting toxic spans efficiently in any environment. The full set of results are reported in the Appendix. We used a batch size of one, in order to mimic the real world scenario. The full specifications of the CPU and GPU environments are listed in the Appendix.

7 Conclusion

This paper introduced *MUDES:* **Mu**ltilingual **De**tection of Offensive **S**pans. We evaluated MUDES on the recently released SemEval-2021 Toxic Spans Detection dataset. Our results show that MUDES outperforms the strong baselines of the competition. Furthermore, we show that once MUDES is trained on English data using state of the art cross-lingual transformer models, it is capable of detecting offensive spans in other languages. With MUDES, we release a Python library, four pre-trained models and an user interface. We show that MUDES is efficient to use in real time scenarios even in a non GPU environment. In future work, we would like to further evaluate MUDES on other datasets. Finally, we would like to implement a flexible multitask architecture capable of detecting offense at both span and post level.

Acknowledgments

We would like to thank the SemEval-2021 Toxic Spans Detection shared task organisers for making this interesting dataset available. We further thank the anonymous reviewers for their insightful feedback.

References

Hwijeen Ahn, Jimin Sun, Chan Young Park, and Jungyun Seo. 2020. NLPDove at SemEval-2020 task 12: Improving offensive language detection with cross-lingual transfer. In *Proceedings of SemEval*.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of SemEval*.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Proceedings of WWW*.

Çağrı Çöltekin. 2020. A Corpus of Turkish Offensive Language on Social Media. In *Proceedings of LREC*.

Patricia Chiril, Farah Benamara Zitoune, Véronique Moriceau, Marlène Coulomb-Gully, and Abhishek Kumar. 2019. Multilingual and multitarget hate speech detection in tweets. In *Proceedings of TALN*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of EMNLP-IJCNLP*.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving Dyberbullying Detection with User Context. In *Proceedings of ECIR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Paula Fortuna, Joao Rocha da Silva, Leo Wanner, Sérgio Nunes, et al. 2019. A Hierarchically-labeled Portuguese Hate Speech Dataset. In *Proceedings of ALW*.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2019. Emoji powered capsule network to detect type and target of offensive posts in social media. In *Proceedings of RANLP*.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2020. InfoMiner at WNUT-2020 task 2: Transformer-based covid-19 informative tweet extraction. In *Proceedings of W-NUT*.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of TRAC*.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating Aggression Identification in Social Media. In *Proceedings of TRAC*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of ICLR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Prasenjit Majumder, Thomas Mandl, et al. 2018. Filtering Aggression from the Multilingual Social Media Feed. In *Proceedings TRAC*.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of RANLP*.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1 – 16.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

Marc Pàmies, Emily Öhman, Kaisla Kajava, and Jörg Tiedemann. 2020. LT@Helsinki at SemEval-2020 task 12: Multilingual or language-specific BERT? In *Proceedings of SemEval*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS*.

John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection. In *Proceedings of SemEval*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL*.

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive Language Identification in Greek. In *Proceedings of LREC*.

Fabio Poletto, Marco Stranisci, Manuela Sanguinetti, Viviana Patti, and Cristina Bosco. 2017. Hate Speech Annotation: Analysis of an Italian Twitter Corpus. In *Proceedings of CLiC-it*.

Tharindu Ranasinghe, Sarthak Gupte, Marcos Zampieri, and Ifeoma Nwogu. 2020. WLV-RIT at HASOC-Dravidian-CodeMix-FIRE2020: Offensive Language Identification in Code-switched YouTube Comments. In *Proceedings of FIRE*.

Tharindu Ranasinghe and Hansi Hettiarachchi. 2020. BRUMS at SemEval-2020 task 12: Transformer based multilingual offensive language identification in social media. In *Proceedings of SemEval*.

Tharindu Ranasinghe, Diptanu Sarkar, Marcos Zampieri, and Alex Ororbia. 2021. WLV-RIT at SemEval-2021 Task 5: A Neural Transformer Framework for Detecting Toxic Spans. In *Proceedings of SemEval*.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual Offensive Language Identification with Cross-lingual Embeddings. In *Proceedings of EMNLP*.

Tharindu Ranasinghe and Marcos Zampieri. 2021. Multilingual Offensive Language Identification for Low-resource Languages. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*.

Tharindu Ranasinghe, Marcos Zampieri, and Hansi Hettiarachchi. 2019. BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Offensive Language Identification. In *Proceedings of FIRE*.

Michael Ridenhour, Arunkumar Bagavathi, Elaheh Raisi, and Siddharth Krishnan. 2020. Detecting Online Hate Speech: Approaches Using Weak Supervision and Network Embedding Models. *arXiv preprint arXiv:2007.12724*.

Hugo Rosa, N Pereira, Ricardo Ribeiro, Paula Costa Ferreira, Joao Paulo Carvalho, S Oliveira, Luísa Coheur, Paula Paulino, AM Veiga Simão, and Isabel Trancoso. 2019. Automatic cyberbullying detection: A systematic review. *Computers in Human Behavior*, 93:333–345.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Weakly Supervised Dataset for Offensive Language Identification. In *arXiv preprint arXiv:2004.14454*.

Joan Serrà and Alexandros Karatzoglou. 2017. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In *Proceedings of RecSys*.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive Language and Hate Speech Detection for Danish. In *Proceedings of LREC*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Proceedings of CCL*.

Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. 2016. A Dictionary-based Approach to Racism Detection in Dutch Social Media. In *Proceedings of TA-COS*.

Shuohuan Wang, Jiaxiang Liu, Xuan Ouyang, and Yu Sun. 2020. Galileo at SemEval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models. In *Proceedings of SemEval*.

Gregor Wiedemann, Seid Muhie Yimam, and Chris Biemann. 2020. UHH-LT at SemEval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection. In *Proceedings of SemEval*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of NeurIPS*.

Mengfan Yao, Charalampos Chelmis, and Daphney-Stavroula Zois. 2019. Cyberbullying Ends Here: Towards Robust Detection of Cyberbullying in Social Media. In *Proceedings of WWW*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of SemEval*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Appendix

i **Training Configurations** We used an Nvidia Tesla K80 GPU to train the models. We divided the dataset into a training set and a validation set using 0.8:0.2 split. We fine tuned the learning rate and number of epochs of the model manually to obtain the best results for the validation set. We obtained $1e^{-}5$ as the best value for learning rate and 3 as the best value for number of epochs for all the languages. We performed *early stopping* if the validation loss did not improve over 10 evaluation steps. Training large models took around 30 minutes while training base models took around 10 minutes. In addition to the learning rate and number of epochs we used the parameter values mentioned in Table 4. We kept these values as constants.

Parameter	Value
adam epsilon	1e-8
warmup ratio	0.1
warmup steps	0
max grad norm	1.0
max seq. length	140
gradient accumulation steps	1

Table 4: Parameter Specifications.

ii **Hardware Specifications**

In Table 5 and in Table 6 we mention the specifications of the GPU and CPU we used for the experiments of the paper. For the training of the MUDES models, we mainly used the GPU. For the efficiency experiments mentioned in Section 6.3 we used both GPU and CPU environments.

Parameter	Value
GPU	Nvidia K80
GPU Memory	12GB
GPU Memory Clock	0.82GHz
Performance	4.1 TFLOPS
No. CPU Cores	2
RAM	12GB

Table 5: GPU Specifications.

iii **Run time**

As expected base models perform efficiently than the large models in both environments. Large models take around 3 seconds for a sentence in a CPU and take around 1 second for a sentence in a GPU while the base models take approximately one third of that time in both environments. From these results it is clear that MUDES is capable of predicting toxic spans efficiently in any environment.

Parameter	Value
CPU Model Name	Intel(R) Xeon(R)
CPU Freq.	2.30GHz
No. CPU Cores	2
CPU Family	Haswell
RAM	12GB

Table 6: CPU Specifications.

Model	GPU Time	CPU Time
en-base	35.51	100.81
en-large	100.36	315.72
multilingual-base	36.23	115.98
multilingual-large	120.54	335.65

Table 7: Time taken to do predictions on 100 sentences in seconds.

Ethics Statement

MUDES is essentially a web-based visualization tool with predictive models trained on multiple publicly available datasets. The authors of this paper used datasets referenced in this paper which were previously collected and annotated. No new data collection has been carried out as part of this work. We have not collected or processed writers'/users' information nor have we carried out any form of user profiling protecting users' privacy and identity.

We understand that every dataset is subject to intrinsic bias and that computational models will inevitably learn biased information from any dataset. We believe that MUDES will help coping with biases in datasets and models as it features: (1) a freely available Python library that other researchers can use to train new models on other datasets; (2) a web-based visualizing tool that can help efforts in reducing biases in offensive language identification as they can be used to process and visualize potentially offensive spans new data. Finally, unlike models trained at the post level, the projected annotation of spans allows users to understand which part of the instance is considered offensive by the models.