Offensive Language Identification with Multi-task Learning

Marcos Zampieri[1*], Tharindu Ranasinghe[2], Diptanu Sarkar[3] and Alex Ororbia[3]

[1]George Mason University, Fairfax, VA, USA.
[2]Aston University, Birmingham, UK.
[3]Rochester Institute of Technology, Rochester, NY, USA.

*Corresponding author(s). E-mail(s): mzampier@gmu.edu;

Abstract

The widespread presence of offensive content is a major issue in social media. This has motivated the development of computational models to identify such content in posts or conversations. Most of these models, however, treat offensive language identification as an isolated task. Very recently, a few datasets have been annotated with post-level offensiveness and related phenomena, such as offensive tokens, humor, engaging content, etc., creating the opportunity of modeling related tasks jointly which will help improve the explainability of offensive language detection systems and potentially aid human moderators. This study proposes a novel multi-task learning (MTL) architecture that can predict: (1) offensiveness at both post and token levels in English; and (2) offensiveness and related subjective tasks such as humor, engaging content, and gender bias identification in multilingual settings. Our results show that the proposed multi-task learning architecture outperforms current state-of-the-art methods trained to identify offense at the post level. We further demonstrate that MTL outperforms single-task learning (STL) across different tasks and language combinations.

Keywords: Offensive Language Identification, Deep Learning, Multi-task Learning, Transformers

1 Introduction

The pervasiveness of offensive content in social media has motivated the development of computational models that can identify the various forms of content such as aggression [1–3], cyber-bullying [4], sentiment [5, 6], emotion[7, 8] and hate speech [9]. Prior work has generally focused either on identifying conversations that are likely to derail [10, 11] or on identifying offensive content within posts, comments, or documents. This has also been the goal of recent popular competitions, e.g., SemEval 2019 and 2020 [12–16].

Substantial progress has been made on identifying offensive language in conversations and posts. Recently, with the goal of improving explainability, multiple post-level offensive language datasets have been annotated with respect to related phenomena such as humor [17], gender bias [18], and offensive token identification [19]. Prior work has addressed these tasks in isolation, building separate models for each [20]. However, this study hypothesizes that jointly modeling these tasks with post-level offensive language identification would help improving the explainability of offensive language identification models. For example, systems capable of detecting offensive token spans would allow content moderators to quickly identify objectionable parts of the posts, especially in long posts. Furthermore, this would allow moderators to more easily approve or reject the decisions of offensive language detection systems. Since these tasks are related, an ideal scenario for multi-task learning (MTL) emerges.

In MTL [21], a model is designed to learn multiple tasks simultaneously using the same set of data. Parameters allocated for two (or more) tasks are shared throughout the optimization process (training), yielding models that often outperform single-task learning (STL) models while reducing potential overfitting [22]. Finally, given that only a single model is produced by MTL compared to the multiple individual models (one for each task) produced by STL, MTL is generally more environmentally friendly, demanding less computing resources (e.g. disk, memory) and energy. This addresses recent efforts in Green AI [23] as well as a growing interest in the NLP community to steer towards both explainable AI and green AI [24], as evidenced by recent workshops such as SustaiNLP.[1]

In this paper, we propose an MTL approach that jointly tackles both token-level and post-level offensive language identification related tasks. As a first step, we jointly model token-level and post-level offensive language identification in a unified system. Then, we extend the MTL approach to model post-level related tasks. To the best of our knowledge, this is the first detailed evaluation of MTL in offensive language identification at both the post and token levels. With our MTL approach, we address four research questions based on performance, speed, efficiency and generalizability, which we describe in detail in Section 4.1. The main contributions of this work are:

1. We develop an MTL model that learns the following jointly: (a) token-level and post-level offensive language identification for English; (b) post-level

[1] https://sites.google.com/view/sustainlp2021/home

offensiveness language identification and related tasks in a multilingual setting.
2. We evaluate the resulting MTL model in terms of performance, efficiency, and generalization ability. We show that the proposed MTL model performs better than STL models at the post- and token-level and is noticeably faster than training multiple STL models. We also evaluate the performance of our model on datasets containing Arabic, Bengali, German, Hindi, and Meitei data.
3. We test the MTL model in zero-shot and few-shot learning scenarios and show that MTL performs better than STL models when there are fewer training data samples available. We demonstrate that MTL is better for zero-shot learning and that MTL generalises well across different languages and domains.
4. We make the code and the trained models freely available to the community. Our complete multi-task framework; MAD (Multi-task Aggression Detection Framework) will be released as an open-source Python package.

2 Related Work

MTL has been employed extensively in computer vision [25, 26] and in NLP tasks such as part-of-speech tagging and named entity recognition [27], text classification [28], natural language generation [29], and offensive language identification [30].

[31] trained an MTL model on different post-level offensive language detection tasks and found that MTL vastly improves the performance on each task, allowing the overall model to strongly generalize to unseen datasets. In [32] sentiment prediction was used as an auxiliary task to detect offensive and hate speech in an MTL setup using a CNN-Bi-LSTM model. Past studies have also demonstrated the value of using neural transformer multi-task learning models to achieve competitive results in offensive language identification shared tasks. [30] trained an MTL model on all three levels of the OLID dataset [33] while [34] trained an MTL model on two post-level tasks – identifying offense and identifying hate speech in Arabic texts using AraBERT [35]. [36] uses a BiLSTM based MTL model to identify toxic comments and spans. In recent work, [19] designed an MTL model based on transformers to detect both token-level and post-level offensive language. Apart from a few notable exceptions (e.g. MUDES [20]), due to the lack of suitable available datasets, there has not been much work in developing statistical learning models that can detect offensive tokens. Our work fills this gap.

None of the studies discussed in this section provided an empirical evaluation of MTL in few-shot, zero-shot, and multilingual settings. Furthermore, these studies have not experimented with MTL in multilingual settings. To address this important gap, we evaluate MTL across different settings for token and post-level offensive language identification. Finally, we evaluate our MTL architecture

on several post-level tasks related to offensive language identification across a wide range of languages.

3 Multitask Architecture

Considering the success that transformers have demonstrated in various offensive language identification tasks, we chose to employ a transformer as the base model for our MTL approach. Our approach will learn several tasks jointly including post-level tasks and token-level tasks. The implemented architecture shares hidden layers between both post and token-level tasks. The shared portion includes a transformer model that learns shared representations (and extracts information) across tasks by minimizing a combined/compound loss function. The task-specific classifiers receive input from the last hidden layer of the transformer language model and predict the output for each of the tasks (details provided in the next two sections).

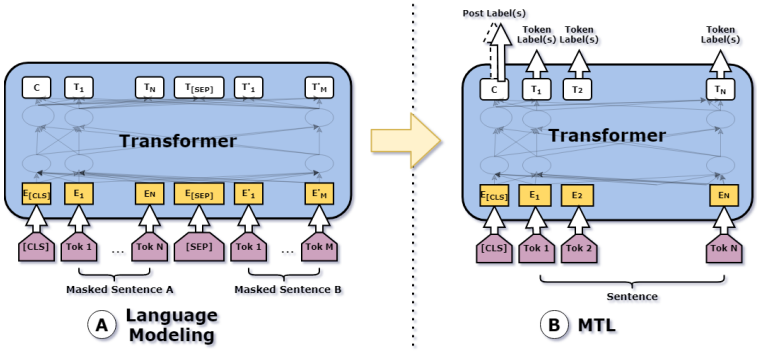


Fig. 1: The two components of the MAD framework. Section A depicts the language modeling component. Section B shows the multi-task aggression detection classifier – the post label predicts post-level aggression; a token label of 0 and 1 denotes non-toxic and toxic tokens, respectively [20]

Post-level Aggression Detection

By utilizing the hidden representation of the classification token (CLS) within the transformer model, we predict the target labels (offensive/hate speech/normal) by applying a linear transformation followed by the softmax activation function (σ):

$$\hat{\mathbf{y}}_{post} = \sigma(\mathbf{W}_{[CLS]} \cdot \mathbf{h}_{[CLS]} + \mathbf{b}_{[CLS]}) \tag{1}$$

where $\cdot$ denotes matrix multiplication, $\mathbf{W}_{[CLS]} \in \mathcal{R}^{D \times 3}$, $\mathbf{b}_{[CLS]} \in \mathcal{R}^{1 \times 2}$, and D is the dimension of input layer $\mathbf{h}$ (top-most layer of the transformer).

Token-level Aggression Detection

We predict the token labels (toxic/non-toxic) by also applying a linear transformation (also followed by the softmax) over every input token from the last hidden layer of the model:

$$\hat{\mathbf{y}}_{token} = \sigma(\mathbf{W}_{token} \cdot \mathbf{h}_t + \mathbf{b}_{token}) \tag{2}$$

where t marks which token the model is to label within a T-length window/token sequence, $\mathbf{W}_{token} \in \mathcal{R}^{D \times 2}$, and $\mathbf{b}_{token} \in \mathcal{R}^{1 \times 2}$.

In the MTL setting, we used different strategies to combine the losses from different type of related tasks; token-level tasks and post-level tasks. We will explain these in the following two sections.

4 Post-level and Token-level

Data

The HateXplain dataset [19] is, to our knowledge, the first benchmark dataset that contains both post and token-level annotations of hate speech and offensiveness. The dataset contains data collected from Twitter and Gab and is annotated using Amazon Mechanical Turk. Each instance in the dataset is annotated by three annotators that choose between three categories - *label* ("offensive", "hate speech", and "normal"), *rationales* (tokens based on which the labeling decision was made), and *target communities* (the group of people denounced in the post). We present examples from the dataset in Table 1.

Post:	[<user>, keep, running, to, russia, you, nazi, sympathizer]
Rationales:	[0, 0, 0, 0, 0, 0, 1, 0]
Label:	Offensive
Post:	[expectations, are, a, bitch]
Rationales:	[0, 0, 0, 0]
Label:	Normal
Post:	[yep, communist, nigger, fag]
Rationales:	[0, 1, 1, 1]
Label:	Hate speech

Table 1: Four instances sampled/extracted from the dataset along with their respective annotations [19]

The dataset contains 20,148 posts (9,055 from Twitter and 11,093 from Gab), out of which 5,935 instances are hateful, 5,480 are offensive, and 7,814 are normal. The dataset also contains 919 undecided posts where all three annotators annotated the label differently. We present the statistics in Table 2. For the task of interest, we used the labels and rationales from the HateXplain dataset. A majority vote strategy, where half or more of the annotators agree

	Class	Train	Dev	Test
Posts	*Offensive*	3,325	1,061	1,093
	Hate	3,547	1,185	1,202
	Normal	4,663	1,598	1,550
	Total	11,535	3,844	3,845
Tokens	*Toxic*	22,224	7,493	7,561
	Non toxic	248,054	82,622	82,432
	Total	270,278	90,115	89,993

Table 2: The distribution of hate speech, offensive, and normal posts and the number of toxic and non toxic tokens in train, development (dev), and test sets.

on an annotation, was used to determine the final annotation of the label and individual tokens in the rationales. We removed the 919 undecided annotations from the final database. The dataset was further split into 11,535 train, 3,844 development (dev), and 3,844 test subsets. The distribution of labels and tokens of the final processed dataset is shown in Table 2. We observe that the train, dev, and test sets follow a similar imbalanced class distribution.

To evaluate how well our architecture performs in zero-shot environments, we used five publicly available offensive language detection datasets released as part of OffensEval 2020, presented in Table 3. Since these datasets have been annotated at the instance level, we followed the evaluation process as explained in Section 4.2.

Lang.	Inst.	S	Reference
Arabic	8,000	T	[37]
Danish	2,961	F, R	[38]
English	14,100	T	[33]
Greek	8,743	T	[**?**]
Turkish	31,756	T	[39]

Table 3: Language (Lang.), instances (Inst.), sources (S), and the source reference for each dataset. "F" stands for Facebook, "R" for Reddit, "T" for Twitter.

4.1 Experimental Setup

MAD consists of two main parts, as depicted in Figure 1. The first part is the language modeling component which runs masked language modeling (MLM) on the given dataset. By default, the modeler masks 15% of the tokens randomly in the dataset and considers sequences with a maximum length of 512. The model weights are stored and loaded into the next part/stage of the MAD framework [40]. The second part/stage of the MAD framework is the multi-task architecture presented in Section 3. It starts by loading the model saved from the first stage to then perform MTL [40].

We train the system by minimising the cross-entropy loss for both (constituent) tasks as defined in Equation 5, where $\mathbf{y}_{post}$ and $\mathbf{y}_{token}$ represent

ground true label vectors (one-hot encodings of the label integers). These particular losses are:

$$\mathcal{L}_{post} = -\sum_{i=1}^{3} \Big(\mathbf{y}_{post} \odot \log(\hat{\mathbf{y}}_{post})\Big)[i] \tag{3}$$

$$\mathcal{L}_{token} = -\sum_{i=1}^{2} \Big(\mathbf{y}_{token} \odot \log(\hat{\mathbf{y}}_{token})\Big)[i] \tag{4}$$

where $\mathbf{v}[i]$ retrieves the ith item in a vector $\mathbf{v}$ and $\odot$ indicates element-wise multiplication. In combining the above two losses into one objective, we introduced α and β parameters to balance the importance of the tasks. To assign equal importance to each task in our experiments, we set $\alpha = \beta = 1$ in this study. The full loss is:

$$\mathcal{L}_{MAD} = \frac{\alpha\mathcal{L}_{post} + \beta\mathcal{L}_{token}}{\alpha + \beta}. \tag{5}$$

We set up two STL baselines – post-level and token-level aggression detection models (each based on neural transformers). The post-level STL model takes the complete sentence as an input and predicts the aggression label – "normal", "offensive", or "hate speech" – using a softmax classifier on top of the CLS token (activation vector). Note that the token-level STL model predicts whether each token (word) in the sentence is toxic or not using a softmax classifier as well. We performed experiments using BERT-base-cased [41] and RoBERTa-base [42] transformer model variants, available in the HuggingFace model repository. We also performed experiments using BERT-base-cased and RoBERTa-base models retrained on HatEval [12] and OLID [33] datasets using MLM; the shifted models are denoted by the H_2O suffix. Furthermore, we used the recently released fBERT model [43] which is a retrained BERT-base-cased model on over 1.4 million offensive instances from SOLID [44].

Metric (Avg.)	**STL Models**		**MAD Model**
	Post-level	*Token-level*	
RAM usage (GB)	2.21	3.39	3.21
GPU usage (GB)	6.18	6.55	8.33
Training time per epoch (Sec)	193.68	178.37	184.73
*Inference w/ GPU (Sec)	3.64	3.56	4.76
*Inference w/ CPU (Sec)	4.24	4.91	5.49

Table 4: Performance comparison of two STL models and the MAD framework model. *Inference was conducted over 100 instances.

For all of the experiments, we optimized parameters with the AdamW update rule using a learning rate of $1e-4$, a maximum sequence length of 128, and a batch size of 16 samples. Early stopping was also executed if the validation loss

did not improve over 10 iterations. The models were trained using a 16 GB Tesla P100 GPU over three epochs. All experiments were run using ten different random seeds, and the mean value plus the standard deviation score across these experiments were reported. We did not perform any data pre-processing steps and used the same datasets published.

Finally, to better cope with class imbalance, we have chosen the macro F_1 score as the evaluation measure for all tasks. For the post-level evaluation, we used a macro F_1 score that is computed as a mean of per-class F_1 scores, as shown below:

$$F_1 = \frac{F_1(Off) + F_1(Hate) + F_1(Normal)}{3}. \tag{6}$$

If the total number of instances is n, the final aggregated F_1 score A for the token-level task is:

$$A = \frac{1}{n}\sum_{i=1}^{n} F_1(Per\ Instance). \tag{7}$$

4.2 Results and Analysis

In this section, we answer each of our following research questions (RQs).

- **RQ1- Performance:** Can MTL outperform STL in (a) token-level and post-level offensive language identification, (b) post-level offensiveness language identification and related tasks?
- **RQ2 - Speed:** Is the proposed MTL approach, in which either two tasks are learned jointly, faster than separate STL models for post- and token-level offensive language identification?
- **RQ3 - Efficiency:** Can MTL learn from fewer training samples compared to a STL setup?
- **RQ4 - Generalizability:** How well does MTL perform in different domains and languages in zero-shot environments compared to STL models?

4.2.1 Supervised Learning

We start by first answering RQ1. We train our MAD framework on the HateXplain training sets and evaluate on the test sets. In Table 5, we compare the results of doing this to the STL setup. We achieve the best result for both the token-level and post-level with our MAD framework model. The fBERT model achieves the overall highest macro F_1 score for the token-level aggression detection using MAD. The RoBERTa-H_2O model achieves a macro F_1 score of 0.6949 at the post-level using the proposed multi-task learning framework. The re-trained language models with MTL achieve better results than the STL model across tasks. Based on these results, we can empirically conclude that MTL outperforms STL in both token-level and post-level offensive language identification by sharing information across the two tasks.

Models	STL		MAD	
	Post-level	*Token-level*	*Post-level*	*Token-level*
BERT	0.670 ± 0.005	0.800 ± 0.008	0.685 ± 0.003	0.810 ± 0.003
BERT-H_20	0.672 ± 0.003	0.803 ± 0.005	0.693 ± 0.004	0.816 ± 0.004
fBERT	0.673 ± 0.003	0.803 ± 0.005	0.693 ± 0.003	**0.818** ± 0.003
RoBERTa	0.674 ± 0.006	0.801 ± 0.004	0.690 ± 0.004	0.812 ± 0.004
RoBERTa-H_20	0.673 ± 0.004	0.803 ± 0.003	**0.694** ± 0.004	0.814 ± 0.004

Table 5: The test set mean macro F_1 scores of 10 runs as well as the standard deviation for different transformer models. Best results for each task are show/presented in bold.

To answer RQ2, we compared the performance of the STL baseline models to our MAD models with respect to computing resources. The results are shown in Table 4. Desirably, we observe that the MAD framework model outperformed the STL models combined for every metric shown/measured. MTL uses less RAM than the token-level model and the training time per epoch is less compared to the post-level model. This demonstrates that MTL is faster and more resource efficient than separate STL models for post and token-level offensive language identification, an insight that should prove to be beneficial for real world applications.

4.2.2 Few-shot Learning

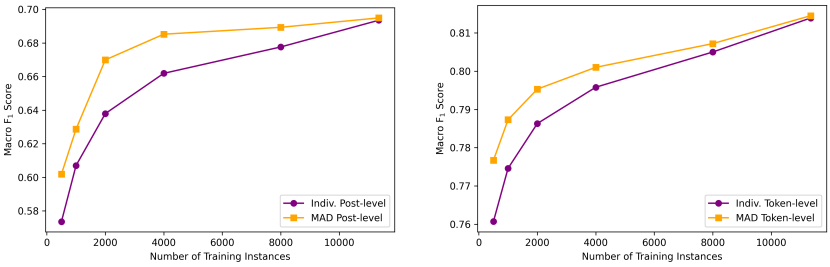
One advantage of multi-task learning is that less data is required to generalize, owing to the fact that information is shared across related tasks; hence it reduces the strict need for a large, labeled dataset [21]. Motivated by this potential benefit, we answer our third RQ: *Can MTL learn from fewer training instances?* by comparing the MAD model framework performance with STL baseline models when the number of training instances is limited. We conduct experiments (see Figure 2 for resulting plot) for the RoBERTa-H_20 model, which performed the best in the previous experiment.

Figure 2 depicts that MTL consistently outperforms STL when varying the number of training instances for both post and token-level aggression detection tasks. This result demonstrates the generalization ability of MTL even when the number of labeled instances available is low. We conclude that the MTL setup performs much better when the number of samples is limited, which is particularly the case for low resource language problems.

4.2.3 Zero-shot Learning

We answer our fourth RQ by evaluating the MTL approach in the zero-shot setting, comparing it with heuristics based on STL. We use the datasets described in Section 4. Since these datasets only contain annotations at the post-level, we carried out the evaluation only at the post-level.

We consider three heuristics to train in the context of zero-shot learning: (1) We train a post-level offensive language identification (transformer)



(a) Post-level Results (b) Token-level Results

Fig. 2: The test F_1 with an increasing number of training samples with RoBERTa-$H_2$0 in STL & MAD setups.

model – a softmax layer is added on top of the CLS token. We train on HateXplain post-level annotations, saving weights. Then we perform zero-shot learning on OffensEval 2020 languages using the saved weights (model is named $\text{Post}_{zeroshot}$). Since the data is labeled as Offensive/Not Offensive, we concatenate predicted offensive and hate speech labels to create a single, offensive label. (2) We train a span-level offensive language identification model based on transformers using MUDES [20]. We train the model on HateXplain token-level annotations, saving the weights. Then we run the model on OffensEval 2020 languages and if the model predicts at least one offensive token, our system labels that post as offensive. This is consistent with OLID annotation guidelines [33]. We call this model $\text{Token}_{zeroshot}$. (3) We train our MTL architecture on HateXplain post-level and token-level annotations, saving the weights. Then we perform zero-shot learning at the post-level of the OffensEval 2020 languages using the saved weights. We call this model $\text{MTL}_{zeroshot}$. Since the other datasets are labeled as Offensive/Not Offensive, we concatenate predicted offensive and hate speech labels to craft one label.

Dataset	**Language(s)**	**Inst.**	**S**	**Task 1**	**Task 2**	**Task 3**
ComMA	Bengali, Hindi, English, Meitei	12,000	Y	**Aggression**	Gender Bias	Communal Bias
GermEval	German	3,000	F	**Toxicity**	Engaging	Fact-Claiming
Hahackathon	English	9,000	K,T	Humor	**Offense**	-
OSACT4	Arabic	10,000	F	**Offense**	**Hate Speech**	-

Table 6: Data Properties: Number of instances (Int.), data sources (S), and label types in all datasets. "F" stands for Facebook, "T" for Twitter, "Y" for Youtube and "K" for Kaggle. Offensive language tasks are shown in bold.

For OLID [33], we used the best model that we obtained on the HateXplain dataset – the RoBERTa-$H_2$0 model. Following the strong cross-lingual offensive language identification results obtained using XLM-R [**?**], we used the xlm-RoBERTa-base [45] model for the non-English datasets. Since this is a purely

zero-shot setup, we do not compare our results to systems that were specifically trained on these datasets.

	Model	Macro F1
AR	$\text{MTL}_{zeroshot}$	0.458 ± 0.006
	$\text{Post}_{zeroshot}$	0.446 ± 0.005
	$\text{Token}_{zeroshot}$	0.438 ± 0.007
DA	$\text{MTL}_{zeroshot}$	0.568 ± 0.008
	$\text{Post}_{zeroshot}$	0.555 ± 0.007
	$\text{Token}_{zeroshot}$	0.542 ± 0.007
EN	$\text{MTL}_{zeroshot}$	0.5230 ± 0.007
	$\text{Post}_{zeroshot}$	0.519 ± 0.005
	$\text{Token}_{zeroshot}$	0.500 ± 0.005
GR	$\text{MTL}_{zeroshot}$	0.489 ± 0.009
	$\text{Post}_{zeroshot}$	0.477 ± 0.004
	$\text{Token}_{zeroshot}$	0.466 ± 0.005
TR	$\text{MTL}_{zeroshot}$	0.478 ± 0.006
	$\text{Post}_{zeroshot}$	0.462 ± 0.006
	$\text{Token}_{zeroshot}$	0.450 ± 0.005

Table 7: Results ordered by Macro F1 for Arabic (AR), Danish (DA), English (EN), Greek (GR), and Turkish (TR) datasets for zero shot experiments.

As observed in Table 7, $\text{MTL}_{zeroshot}$ outperforms the other zero-shot approaches that are based on STL for all the languages. This affirmatively answers our fourth and final research question: *MTL outperform STL models in zero-shot scenarios, demonstrating strong generalization ability.*

5 Related Tasks

Data

We used four publicly available datasets: ComMA [46], GermEval [18], Hahackathon [17], and OSACT4 [47], summarized in Table 6.

5.1 Experimental Setup

For these datasets, we also use/employ the MAD framework as showed in Figure 1. However, since these related tasks only contained post-level labels, we did not use the token heads in the MTL architecture. Instead, we used/adapted multiple post-level heads. We train our MTL model by minimising the cross-entropy loss for all of the (inherent) tasks. All of the losses (for all tasks) are then combined into one objective and we assign equal importance to each task in our experiments. The full loss is:

$$\mathcal{L}_{MAD} = \frac{\sum_{j=1}^{n} \mathcal{L}_j}{n} \tag{8}$$

where n is the number of tasks and $\mathcal{L}_j$ is the loss function associated with task j.

Similar to the previous experiments, we compared the MTL architecture with STL post-level baselines where the STL model takes in the complete sentence as an input and predicts the post label using a softmax classifier on top of the CLS token (activation vector). We performed experiments using multilingual transformer models, such as mBERT and XLM-R, as well as monolingual transformer models, specifically ones that were trained specifically to support each language. For ComMA, we used IndicBERT [48] which supports Bengali, Hindi, and English. For GermEval we used gBERT [49] and gELECTRA [49] while for OSACT4, we used AraBERT and AraElectra.

Similar to the previous set of experiments, for all tasks in this section, we optimized parameters (with AdamW) using a learning rate of $1e-4$, a maximum sequence length of 128, and a batch size of 16 samples. Early stopping was also executed if the validation loss did not increase over a 10 iteration period. The models were trained using a 16 GB Tesla P100 GPU over three epochs. The output results of neural transformer models can heavily depend on the initial weights and, more importantly, on the experimental and simulation setup [50]. The standard procedure to address this variation is to run the transformer model in different random seeds and report the mean and standard deviation of multiple runs [50–52]. Recent literature suggests that running experiments ten times provides more reliable results [53], therefore, all experiments were ran for ten different random seeds with reported mean values over 10 trials with standard deviation. For the evaluation of each task, we used the same evaluation metrics used by the authors of the original datasets.

Dataset	**Metric (Avg.)**	**STL**			**MTL**
		Task 1	*Task 2*	*Task 3*	
ComMa	RAM (GB)	1.86	1.88	1.84	2.45
	GPU (GB)	3.56	3.56	3.56	4.12
	Training time (Sec)	140.67	150.98	148.93	135.76
	*Inference w/ GPU (Sec)	2.86	2.88	2.92	3.76
	*Inference w/ CPU (Sec)	4.12	4.20	4.15	5.62
GermEval	RAM (GB)	2.45	2.48	2.48	3.12
	GPU (GB)	6.32	6.35	6.36	8.91
	Training time (Sec)	198.42	199.42	197.54	190.23
	*Inference w/ GPU (Sec)	3.91	3.94	3.97	4.45
	*Inference w/ CPU (Sec)	4.41	4.43	4.45	5.68
Hahackathon	RAM (GB)	1.12	1.13	-	1.56
	GPU (GB)	3.12	3.14	-	1.23
	Training time (Sec)	92.54	98.45	-	90.03
	*Inference w/ GPU (Sec)	1.92	1.95	-	2.56
	*Inference w/ CPU (Sec)	2.65	2.69	-	3.45
OSACT4	RAM (GB)	2.18	2.23	-	3.18
	GPU (GB)	6.15	6.19	-	8.44
	Training time (Sec)	193.65	194.45	-	184.67
	*Inference w/ GPU (Sec)	3.61	3.69	-	4.70
	*Inference w/ CPU (Sec)	4.21	4.29	-	5.31

Table 8: Performance of 3 STL models versus MTL model for ComMa with mBERT. *Inference was conducted over 100 instances.

5.2 Results and Analysis

To evaluate our proposed multi-task learning model, we experimented with it on the four datasets. For the evaluation of each task, we used the same evaluation metrics used by the authors of each of the original datasets.

Dataset	Models	STL			MTL		
		Task 1	*Task 2*	*Task 3*	*Task 1*	*Task 2*	*Task 3*
ComMA	IndicBERT	0.501± 0.005	0.742± 0.004	0.745± 0.006	0.526± 0.004	0.768± 0.008	0.769 ± 0.005
	mBERT	0.513± 0.005	0.762± 0.006	0.766± 0.008	**0.538**± 0.007	**0.789**± 0.008	**0.788** ± 0.005
	XLM-R	0.481± 0.005	0.721± 0.006	0.720± 0.006	0.495 ± 0.007	0.749 ± 0.005	0.748 ± 0.004
GermEval	mBERT	0.651± 0.005	0.732± 0.005	0.745± 0.007	0.681± 0.006	0.758± 0.007	0.766 ± 0.005
	gBERT	0.693± 0.005	0.784± 0.006	0.785± 0.006	0.701± 0.005	0.802± 0.004	0.806± 0.004
	gELECTRA	0.713± 0.004	0.784± 0.007	0.796± 0.006	**0.732**± 0.005	**0.801**± 0.006	**0.819**± 0.005
HaHackathon	BERT	0.951± 0.006	0.452± 0.006	-	0.964± 0.005	0.436± 0.004	-
	XLNet	0.953± 0.007	0.452± 0.006	-	0.971± 0.004	0.429± 0.005	-
	fBERT	0.941± 0.007	0.445± 0.007	-	**0.981** ± 0.006	**0.405**± 0.005	-
OSACT4	mBERT	0.854± 0.006	0.796± 0.007	-	0.877± 0.005	0.805± 0.004	-
	AraBERT	0.901± 0.006	0.812± 0.006	-	**0.911**± 0.005	**0.845**± 0.004	-
	AraElectra	0.891± 0.006	0.813± 0.007	-	0.909± 0.005	0.828± 0.005	-

Table 9: Micro F_1 scores of different models for the four datasets, except for Hahackathon Task 2, where the results reported are with respect to RMSE. The best results is shown in bold.

Again, we start by answering **RQ1**, we compared the results of the MTL to STL models across all datasets presented in Section 4. We train each model on the training set of each database and evaluate it on the relevant test set. In Table 9 we present the mean results of ten runs along with standard deviation.

We observe that MTL consistently outperforms STL in all tasks across all of the datasets. For ComMA, mBERT [41] with MTL performed best across all the tasks. For GermEval 2021, the gElectra [49] model with MTL outperformed all of the other models. For Hahackathon, fBERT [43] with MTL performed best and, finally, for OSACT4 2020, AraBERT with MTL produced the best results across all tasks. Note that, for all of the transformer models we experimented with, MTL variants achieve better performance than STL ones.

To answer **RQ2**, we compared the performance of the STL baseline models with our MTL models with respect to computing resources (using the best transformer model of each dataset). The results of this comparison are shown in Table 8 – observe that the model used within the proposed MTL framework outperforms the STL models combined for every metric across all datasets.

6 Conclusion

In this paper, we introduce MAD, a multi-task architecture based on neural transformers, and evaluated it across different key training setups. To the best of our knowledge, this is the first empirical evaluation of MTL in both post-level and token-level offensive language identification.

This work demonstrates that the proposed MTL model outperforms STL models on both the tasks of token-level and post-level offensive language identification (**RQ1**). We furthermore demonstrated that our MTL model uses less resources (in terms of RAM usage, GPU usage, and training and inference

time) than the two STL models combined, showing that MTL could prove valuable for practical applications (**RQ2**). Furthermore, we experimented with MTL in a few-shot setup and demonstrated that it could desirably outperform STL models when the amount of training data available is small, confirming that MTL could prove useful for even low resource language problems (**RQ3**). When considering the zero-shot setup, we showed that MTL outperforms STL-based heuristics across five different datasets. This showcased that MTL models generalize better across datasets than STL models (**RQ4**).

Finally, as only a single machine learning model is produced by MTL compared to the multiple statistical learning models produced in a standard STL approach, we show that our proposed approach not only achieves higher performance but is also faster than STL. MTL is, therefore, environmentally friendlier as it demands less computing resources and energy compared to STL. This efficient use of resources addresses recent efforts in Green AI [23] and the recent ACL Policy Document on Efficient NLP.[2]

With respect to future work, we would like to expand MTL-based offensive language identification to operate with more languages and domains by annotating additional datasets. We believe that our MTL systems improve interpretability as well as generalizability over the current state-of-the-art post-level offensive language identification models, offering a powerful neural transformer-based framework for the development of future, promising offensive content and language identification systems and applications. Finally, we would like to use MTL to explore the interplay between offensive content and sarcasm as in the recent HaHackathon competition at SemEval-2021 [54]. BERT-based models have been successfully applied to sarcasm detection [55, 56] suggesting that the approach presented in this paper would potentially achieve good performance on identified sarcasm in an MTL setting.

Statements and Declarations

- Funding - None
- Conflict of interest/Competing interests : The authors declare that they have no confict of interest.
- Availability of data and materials : Data is available at https://github.com/imdiptanu/MAD.
- Code availability : Code is available at https://github.com/imdiptanu/MAD
- Authors' contributions :
  - Marcos Zampieri - Problem formulation, Conducting experiments, Writing, Supervising
  - Tharindu Ranasinghe - Coding, Conducting experiments, Writing
  - Diptanu Sarkar - Coding, Conducting experiments, Writing
  - Alex Ororbia - Problem formulation, Writing, Supervising
- Ethical Approval : Not Applicable

[2] https://www.aclweb.org/portal/content/efficient-nlp-policy-document

- Acknowledgments : We would like to thank the creators of the datasets for making them available.

References

[1] Kumar, R., Ojha, A.K., Malmasi, S., Zampieri, M.: Benchmarking aggression identification in social media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 1–11. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018). https://aclanthology.org/W18-4401

[2] Kumar, R., Ojha, A.K., Malmasi, S., Zampieri, M.: Evaluating aggression identification in social media. In: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pp. 1–5. European Language Resources Association (ELRA), Marseille, France (2020). https://aclanthology.org/2020.trac-1.1

[3] Casavantes, M., Aragón, M.E., González, L.C., Montes-y-Gómez, M.: Leveraging posts' and authors' metadata to spot several forms of abusive comments in twitter. Journal of Intelligent Information Systems (2023). https://doi.org/10.1007/s10844-023-00779-z

[4] Rosa, H., Pereira, N., Ribeiro, R., Ferreira, P.C., Carvalho, J.P., Oliveira, S., Coheur, L., Paulino, P., Simão, A.V., Trancoso, I.: Automatic cyberbullying detection: A systematic review. Computers in Human Behavior **93**, 333–345 (2019)

[5] Kanfoud, M.R., Bouramoul, A.: Senticode: A new paradigm for one-time training and global prediction in multilingual sentiment analysis. Journal of Intelligent Information Systems **59**(2), 501–522 (2022). https://doi.org/10.1007/s10844-022-00714-8

[6] Vohra, A., Garg, R.: Deep learning based sentiment analysis of public perception of working from home through tweets. Journal of Intelligent Information Systems **60**(1), 255–274 (2023). https://doi.org/10.1007/s10844-022-00736-2

[7] Skenduli, M.P., Biba, M., Loglisci, C., Ceci, M., Malerba, D.: Mining emotion-aware sequential rules at user-level from micro-blogs. Journal of Intelligent Information Systems **57**(2), 369–394 (2021). https://doi.org/10.1007/s10844-021-00647-8

[8] Abdi, S., Bagherzadeh, J., Gholami, G., Tajbakhsh, M.S.: Using an auxiliary dataset to improve emotion estimation in users' opinions. Journal of Intelligent Information Systems **56**(3), 581–603 (2021). https:

//doi.org/10.1007/s10844-021-00643-y

[9] Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. Proceedings of the International AAAI Conference on Web and Social Media **11**(1), 512–515 (2017). https://doi.org/10.1609/icwsm.v11i1.14955

[10] Zhang, J., Chang, J., Danescu-Niculescu-Mizil, C., Dixon, L., Hua, Y., Taraborelli, D., Thain, N.: Conversations gone awry: Detecting early signs of conversational failure. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1350–1361. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1125. https://aclanthology.org/P18-1125

[11] Chang, J.P., Cheng, J., Danescu-Niculescu-Mizil, C.: Don't let me be misunderstood:comparing intentions and perceptions in online discussions. In: Proceedings of The Web Conference 2020. WWW '20, pp. 2066–2077. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3366423.3380273. https://doi.org/10.1145/3366423.3380273

[12] Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F.M., Rosso, P., Sanguinetti, M.: SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 54–63. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019). https://doi.org/10.18653/v1/S19-2007. https://aclanthology.org/S19-2007

[13] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 75–86. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019). https://doi.org/10.18653/v1/S19-2010. https://aclanthology.org/S19-2010

[14] Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., Çöltekin, Ç.: SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In: Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 1425–1447. International Committee for Computational Linguistics, Barcelona (online) (2020). https://doi.org/10.18653/v1/2020.semeval-1.188. https://aclanthology.org/2020.semeval-1.188

[15] Modha, S., Mandl, T., Shahi, G.K., Madhu, H., Satapara, S., Ranasinghe, T., Zampieri, M.: Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and

conversational hate speech. In: Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation. FIRE '21, pp. 1–3. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3503162.3503176. https://doi.org/10.1145/3503162.3503176

[16] Satapara, S., Majumder, P., Mandl, T., Modha, S., Madhu, H., Ranasinghe, T., Zampieri, M., North, K., Premasiri, D.: Overview of the hasoc subtrack at fire 2022: Hate speech and offensive content identification in english and indo-aryan languages. In: Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation. FIRE '22, pp. 4–7. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3574318.3574326. https://doi.org/10.1145/3574318.3574326

[17] Meaney, J.A., Wilson, S., Chiruzzo, L., Lopez, A., Magdy, W.: SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 105–119. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.semeval-1.9. https://aclanthology.org/2021.semeval-1.9

[18] Risch, J., Stoll, A., Wilms, L., Wiegand, M.: Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In: Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments, pp. 1–12. Association for Computational Linguistics, Duesseldorf, Germany (2021). https://aclanthology.org/2021.germeval-1.1

[19] Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P., Mukherjee, A.: Hatexplain: A benchmark dataset for explainable hate speech detection. Proceedings of the AAAI Conference on Artificial Intelligence **35**(17), 14867–14875 (2021). https://doi.org/10.1609/aaai.v35i17.17745

[20] Ranasinghe, T., Zampieri, M.: MUDES: Multilingual detection of offensive spans. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, pp. 144–152. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-demos.17. https://aclanthology.org/2021.naacl-demos.17

[21] Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (1997). https://doi.org/10.1023/A:1007379606734

[22] Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering **34**(12), 5586–5609 (2022). https://doi.org/10.1109/TKDE.2021.3070203

[23] Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green ai. Commun.

ACM **63**(12), 54–63 (2020). https://doi.org/10.1145/3381831

[24] Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A survey of the state of explainable AI for natural language processing. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pp. 447–459. Association for Computational Linguistics, Suzhou, China (2020). https://aclanthology.org/2020.aacl-main.46

[25] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)

[26] Zhao, X., Li, H., Shen, X., Liang, X., Wu, Y.: A modulation module for multi-task learning with applications in image retrieval. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018, pp. 415–432. Springer, Cham (2018)

[27] Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08, pp. 160–167. Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1390156.1390177. https://doi.org/10.1145/1390156.1390177

[28] Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1–10. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1001. https://aclanthology.org/P17-1001

[29] Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4487–4496. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1441. https://aclanthology.org/P19-1441

[30] Dai, W., Yu, T., Liu, Z., Fung, P.: Kungfupanda at SemEval-2020 task 12: BERT-based multi-TaskLearning for offensive language detection. In: Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 2060–2066. International Committee for Computational Linguistics, Barcelona (online) (2020). https://doi.org/10.18653/v1/2020.semeval-1.272. https://aclanthology.org/2020.semeval-1.272

[31] Talat, Z., Thorne, J., Bingel, J.: In: Golbeck, J. (ed.) Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection, pp. 29–55 (2018). https://doi.org/10.1007/978-3-319-78583-7_3.

https://doi.org/10.1007/978-3-319-78583-7_3

[32] Abu Farha, I., Magdy, W.: Multitask learning for Arabic offensive language and hate-speech detection. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 86–90. European Language Resource Association, Marseille, France (2020). https://aclanthology.org/2020.osact-1.14

[33] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Predicting the type and target of offensive posts in social media. 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), pp. 1415–1420. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1144. https://aclanthology.org/N19-1144

[34] Djandji, M., Baly, F., Antoun, W., Hajj, H.: Multi-task learning using AraBert for offensive language detection. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 97–101. European Language Resource Association, Marseille, France (2020). https://aclanthology.org/2020.osact-1.16

[35] Antoun, W., Baly, F., Hajj, H.: AraBERT: Transformer-based model for Arabic language understanding. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 9–15. European Language Resource Association, Marseille, France (2020). https://aclanthology.org/2020.osact-1.2

[36] Nelatoori, K.B., Kommanti, H.B.: Multi-task learning for toxic comment classification and rationale extraction. Journal of Intelligent Information Systems (2022). https://doi.org/10.1007/s10844-022-00726-4

[37] Mubarak, H., Rashed, A., Darwish, K., Samih, Y., Abdelali, A.: Arabic offensive language on Twitter: Analysis and experiments. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop, pp. 126–135. Association for Computational Linguistics, Kyiv, Ukraine (Virtual) (2021). https://aclanthology.org/2021.wanlp-1.13

[38] Sigurbergsson, G.I., Derczynski, L.: Offensive language and hate speech detection for Danish. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 3498–3508. European Language Resources Association, Marseille, France (2020). https://aclanthology.org/2020.lrec-1.430

[39] Çöltekin, Ç.: A corpus of Turkish offensive language on social media. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 6174–6184. European Language Resources Association, Marseille, France (2020). https://aclanthology.org/2020.lrec-1.758

[40] Sarkar, D.: An empirical study of offensive language in online interactions. Master's thesis, Rochester Institute of Technology

[41] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: 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), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423

[42] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)

[43] Sarkar, D., Zampieri, M., Ranasinghe, T., Ororbia, A.: fBERT: A neural transformer for identifying offensive content. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 1792–1798. Association for Computational Linguistics, Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.154. https://aclanthology.org/2021.findings-emnlp.154

[44] Rosenthal, S., Atanasova, P., Karadzhov, G., Zampieri, M., Nakov, P.: SOLID: A large-scale semi-supervised dataset for offensive language identification. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 915–928. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.findings-acl.80. https://aclanthology.org/2021.findings-acl.80

[45] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.747. https://aclanthology.org/2020.acl-main.747

[46] Kumar, R., Ratan, S., Singh, S., Nandi, E., Devi, L.N., Bhagat, A., Dawer, Y., Lahiri, B., Bansal, A.: ComMA@ICON: Multilingual gender biased and communal language identification task at ICON-2021. In: Proceedings of the 18th International Conference on Natural Language Processing: Shared Task on Multilingual Gender Biased and Communal Language

Identification, pp. 1–12. NLP Association of India (NLPAI), NIT Silchar (2021). https://aclanthology.org/2021.icon-multigen.1

[47] Mubarak, H., Darwish, K., Magdy, W., Elsayed, T., Al-Khalifa, H.: Overview of OSACT4 Arabic offensive language detection shared task. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 48–52. European Language Resource Association, Marseille, France (2020). https://aclanthology.org/2020.osact-1.7

[48] Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M.M., Kumar, P.: IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4948–4961. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.445. https://aclanthology.org/2020.findings-emnlp.445

[49] Chan, B., Schweter, S., Möller, T.: German's next language model. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6788–6796. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.598. https://aclanthology.org/2020.coling-main.598

[50] Ein-Dor, L., Halfon, A., Gera, A., Shnarch, E., Dankin, L., Choshen, L., Danilevsky, M., Aharonov, R., Katz, Y., Slonim, N.: Active Learning for BERT: An Empirical Study. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7949–7962. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.638. https://aclanthology.org/2020.emnlp-main.638

[51] Risch, J., Krestel, R.: Bagging BERT models for robust aggression identification. In: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pp. 55–61. European Language Resources Association (ELRA), Marseille, France (2020). https://aclanthology.org/2020.trac-1.9

[52] Mosbach, M., Andriushchenko, M., Klakow, D.: On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=nzpLWnVAyah

[53] Sellam, T., Yadlowsky, S., Tenney, I., Wei, J., Saphra, N., D'Amour, A., Linzen, T., Bastings, J., Turc, I.R., Eisenstein, J., Das, D., Pavlick, E.: The multiBERTs: BERT reproductions for robustness analysis. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=K0E_F0gFDgA

[54] Meaney, J.A., Wilson, S., Chiruzzo, L., Lopez, A., Magdy, W.: SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 105–119. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.semeval-1.9. https://aclanthology.org/2021.semeval-1.9

[55] Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., Poria, S.: Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4619–4629. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1455. https://aclanthology.org/P19-1455

[56] Pandey, R., Singh, J.P.: Bert-lstm model for sarcasm detection in code-mixed social media post. Journal of Intelligent Information Systems **60**(1), 235–254 (2023). https://doi.org/10.1007/s10844-022-00755-z