Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pages 120–125
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Multilingual Joint Fine-tuning of Transformer models for identifying Trolling, Aggression and Cyberbullying at TRAC 2020

Sudhanshu Mishra[1], **Shivangi Prasad**[2], **Shubhanshu Mishra**[2]
[1]Indian Institute of Technology Kanpur, [2]University of Illinois at Urbana Champaign
sdhanshu@iitk.ac.in, sprasad6@illinois.edu, mishra@shubhanshu.com

Abstract

We present our team '3Idiots' (referred as 'sdhanshu' in the official rankings) approach for the Trolling, Aggression and Cyberbullying (TRAC) 2020 shared tasks. Our approach relies on fine-tuning various Transformer models on the different datasets. We also investigated the utility of task label marginalization, joint label classification, and joint training on multilingual datasets as possible improvements to our models. Our team came second in English sub-task A, a close fourth in the English sub-task B and third in the remaining 4 sub-tasks. We find the multilingual joint training approach to be the best trade-off between computational efficiency of model deployment and model's evaluation performance. We open source our approach at https://github.com/socialmediaie/TRAC2020.

Keywords: Aggression Identification, Misogynistic Aggression Identification , BERT, Transformers, Neural Networks.

1. Introduction

The internet has become more accessible in recent years, leading to an explosion in content being produced on social media platforms. This content constitutes public views, and opinions. Furthermore, social media has become an important tool for shaping the socio-economic policies around the world. This utilization of social media by public has also attracted many malicious actors to indulge in negative activities on these platforms. These negative activities involve, among others, misinformation, trolling, displays of aggression, as well as cyberbullying behaviour (Mishra et al., 2014). These activities have led to derailment and disruption of social conversation on these platforms. However, efforts to moderate these activities have revealed the limits of manual content moderation systems, owing to the the scale and velocity of content production. This has allowed more and more platforms to move to automated methods for content moderation. However, simple rule based methods do not work for subjective tasks like hate-speech, trolling, and aggression identification. These limitations have moved the automated content moderation community to investigate the usage of machine learning based intelligent systems which can identify the nuance in language to perform the above mentioned tasks more efficiently.

In this work, we utilize the recent advances in information extraction systems for social media data. In the past we have used information extraction for identifying sentiment in tweets (Mishra and Diesner, 2018) (Mishra et al., 2015), enthusiastic and passive tweets and users (Mishra et al., 2014) (Mishra and Diesner, 2019), and extracting named entities (Mishra, 2019) (Mishra and Diesner, 2016). We extend a methodology adopted in our previous work (Mishra and Mishra, 2019) on on Hate Speech and Offensive Content (HASOC) identification in Indo-European Languages (Mandl et al., 2019). In our work on HASOC, we investigated the usage of monolingual and multilingual transformer (Vaswani et al., 2017) models (specifically Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2019)) for hate speech identification. In this work, we extend our analysis to include a newer variant of transformer model called XLM-Roberta (Conneau et al., 2019). In this year's TRAC (Ritesh Kumar and Zampieri, 2020) shared tasks, our team '3Idiots' (our team is referred as 'sdhanshu' in the rankings(Ritesh Kumar and Zampieri, 2020)) experimented with fine-tuning different pre-trained transformer networks for classifying aggressive and misogynistic posts. We also investigated a few new techniques not used before, namely, joint multitask multilingual training for all tasks, as well as marginalized predictions based on joint multitask model probabilities. Our team came second in English sub-task A, a close fourth in the English sub-task B and third in the remaining 4 sub-tasks. We open source our approach at https://github.com/socialmediaie/TRAC2020.

2. Related Work

The shared tasks in this year's TRAC focused on Aggression and Misogynistic content classification(Ritesh Kumar and Zampieri, 2020), the related work in this field focuses on a more general topic that is hate speech and abusive content detection. The abusive content identification tasks are are challenging due to the lack of large amounts of labeled datasets. The currently available datasets lack variety and uniformity. They are usually skewed towards specific topics in hate speech like racism, sexism. A good description of the various challenges in abusive content detection can be found here. (Vidgen et al., 2019) The recent developments in the field of Natural Language Processing (NLP) have really spearheaded research in this domain. One of the most remarkable developments in NLP was the introduction of transformer models (Vaswani et al., 2017) using different attention mechanisms, which have become state of the art in many NLP tasks beating recurrent neural networks and gated networks. These transformer models can process longer contextual information than the standard RNNs. One of the main state of the art models in many NLP tasks are Bidirectional Encoder Representation from Transformers (BERT) models (Devlin et al., 2019). The open source transformers library by HuggingFace Inc. (Wolf et al., 2019) has made fine-tuning pre-trained trans-

former models easy. In a 2019 task on Hate Speech and Offensive Content (HASOC) identification in Indo-European Languages (Mandl et al., 2019), we had the opportunity to try out different BERT models (Mishra and Mishra, 2019). Our models performed really well in the HASOC shared task, achieving first position on 3 of the 8 sub-tasks and being within top 1% for 5 of the 8 sub-tasks. This motivated us to try similar methods in this year's TRAC (Ritesh Kumar and Zampieri, 2020) shared tasks using other transformer models using our framework from HASOC based on the HuggingFace transformers library[1].

3. Data

The data-set provided by the organizers consisted of posts taken from Twitter and YouTube. They provided us with training and dev datasets for training and evaluation of our models for three languages, namely **English (ENG)**, **Hindi (HIN)** and **Bengali (IBEN)**. For both the sub-tasks, the same training and dev data-sets were used with different fine-tuning techniques. The **Aggression Identification** sub-task (task - A) consisted of classifying the text data into **'Overtly Aggressive' (OAG)**, **'Covertly Aggressive' (CAG)** and **'Non-Aggressive' (NAG)** categories. The **Misogynistic Aggression Identification** sub-task (task - B) consisted of classifying the text data into **'Gendered' (GEN)** and **'Non-gendered' (NGEN)** categories. For further details about the shared tasks, we refer to the TRAC website and the shared task paper (Ritesh Kumar and Zampieri, 2020). The data distribution for each language and each sub-task is mentioned in Table 1.

Lang	task A			task B		
	train	dev	test	train	dev	test
ENG	4263	1066	1200	4263	1066	1200
HIN	3984	997	1200	3984	997	1200
IBEN	3826	957	1188	3826	957	1188

Table 1: Distribution of number of tweets in different datasets and splits.

4. Methodology

Our methods used for the TRAC (Ritesh Kumar and Zampieri, 2020) shared tasks are inspired from our previous work (Mishra and Mishra, 2019) at HASOC 2019 (Mandl et al., 2019). For the different shared tasks we fine-tuned different pre-trained transformer neural network models using the HuggingFace transformers library.

4.1. Transformer Model

For all of the sub-tasks we used different pre-trained transformer neural network models. The transformer architecture was proposed in (Vaswani et al., 2017). It's effectiveness has been proved in numerous NLP tasks like machine translation, sequence classification and natural language generation. A transformer consists of a set of stacked encoders and decoders with different attention mechanisms. Like any encoder-decoder model, it takes an input sequence produces a latent representation which is passed on to the decoder which gives an output sequence. A major change in the transformer architectures was that the decoder is supplied with all of the hidden states of the encoder. This helps the model to gain contextual information for even large sequences. To process the texts we utilized the model specific tokenizers provided in the HuggingFace transformers library to convert the texts into a sequence of tokens which are then utilised to generate the features for the model. We utilised similar training procedures like the one used in our HASOC 2019 submission code[2]. We investigated with two variants of transformer models, namely BERT (both monolingual and multilingual) (Devlin et al., 2019) and XLM-Robert (Conneau et al., 2019). While, for BERT we tested its in English, and multilingual versions, whereas, for XLM-Roberta we tried only the multilingual model. There are many other variants of transformers but we could not try them out because of GPU memory constraints, as these models require GPUs with very large amounts of RAM.

4.2. Fine-Tuning Techniques

For the TRAC shared tasks we investigated the following fine-tuning techniques on the different transformer models.

- **Simple fine-tuning:** In this approach we simply fine tune an existing transformer model for the specific language on the new classification data.
- **Joint label training (C):** In our approach during the HASOC (Mishra and Mishra, 2019) shared tasks we had to tackle the problem of data sparsity as the different tasks did not have enough data samples, which makes the training of deep learning models very difficult. To tackle this issue, we had combined the labels of the different shared tasks, which enabled us to train a single model for both the tasks. We tried the same approach for TRAC (Ritesh Kumar and Zampieri, 2020) ,although, here both tasks had the same dataset, so this did not result in an increase in the size of the dataset but it did enable us to train a single model capable of handling both the tasks. We combined the labels of the 2 sub-tasks and trained a single model for the classification. The predicted outputs were **NAG-GEN**, **NAG-NGEN**, **CAG-GEN**, **CAG-NGEN**, **OAG-GEN** and **OAG-NGEN** respectively, taking the argmax of the outputs produces the corresponding label for each task. To get the output of the respective tasks is trivial, we just have to separate the labels by the '-' symbol, where the first word corresponds to sub-task A and second word corresponds to sub-task B. The models using this technique are labeled with **(C)** in the results table below.
- **Marginalization of labels (M):** While using the previous method, in HASOC (Mishra and Mishra, 2019) we just took the respective probability of the combined label and made our decision on the basis of that probability. A limitation of this approach is that it does not guarentee consistency

[1] https://github.com/huggingface/transformers

[2] https://github.com/socialmediaie/HASOC2019

lang	task	model	run_id	Macro-F1 dev	Macro-F1 train	Weighted-F1 dev	Weighted-F1 train	rank
ENG	**A**	**bert-base-multilingual-uncased (ALL)**	**9**	0.611	0.903	0.798	0.957	1
		bert-base-uncased (C)	**4 (C)**	0.596	0.902	0.795	0.956	2
		bert-base-uncased (M)	**4 (M)**	0.595	0.900	0.795	0.956	3
		bert-base-cased (C)	**3 (C)**	0.571	0.912	0.786	0.961	4
		bert-base-uncased	**2**	0.577	0.948	0.784	0.979	5
		bert-base-cased (M)	**3 (M)**	0.568	0.908	0.782	0.960	6
		bert-base-multilingual-uncased (ALL) (M)	**9 (M)**	0.555	0.865	0.780	0.939	7
		bert-base-multilingual-uncased (ALL) (C)	**9 (C)**	0.550	0.871	0.778	0.941	8
		bert-base-cased	**1**	0.563	0.966	0.774	0.987	9
		xlm-roberta-base	**5**	0.531	0.676	0.772	0.862	10
		xlm-roberta-base (C)	**6 (C)**	0.515	0.640	0.762	0.835	11
		xlm-roberta-base (ALL)	**9**	0.512	0.610	0.762	0.823	12
		xlm-roberta-base (M)	**6 (M)**	0.518	0.634	0.761	0.830	13
HIN	**A**	**bert-base-multilingual-uncased**	**5**	0.637	0.846	0.708	0.881	1
		bert-base-multilingual-uncased (ALL) (C)	**9 (C)**	0.628	0.903	0.696	0.924	2
		bert-base-multilingual-uncased (ALL) (M)	**9 (M)**	0.626	0.899	0.695	0.921	3
		bert-base-multilingual-uncased (ALL)	**9**	0.626	0.939	0.694	0.952	4
		bert-base-multilingual-uncased (C)	**3 (C)**	0.616	0.849	0.688	0.884	5
		bert-base-multilingual-uncased (M)	**3 (M)**	0.611	0.848	0.684	0.884	6
		xlm-roberta-base (ALL)	**9**	0.598	0.698	0.672	0.753	7
		xlm-roberta-base	**2**	0.394	0.388	0.527	0.509	8
		xlm-roberta-base (C)	**4 (C)**	0.245	0.240	0.426	0.406	9
		xlm-roberta-base (M)	**4 (M)**	0.245	0.240	0.426	0.406	9
IBEN	**A**	**bert-base-multilingual-uncased (ALL)**	**9**	0.698	0.933	0.737	0.945	1
		xlm-roberta-base (M)	**4 (M)**	0.694	0.758	0.732	0.796	2
		xlm-roberta-base (C)	**4 (C)**	0.692	0.757	0.731	0.796	3
		bert-base-multilingual-uncased (M)	**3 (M)**	0.686	0.856	0.729	0.879	4
		bert-base-multilingual-uncased (C)	**3 (C)**	0.684	0.860	0.728	0.883	5
		bert-base-multilingual-uncased	**5**	0.680	0.903	0.726	0.918	6
		bert-base-multilingual-uncased (ALL) (C)	**9 (C)**	0.686	0.893	0.726	0.912	7
		bert-base-multilingual-uncased (ALL) (M)	**9 (M)**	0.683	0.893	0.723	0.913	8
		xlm-roberta-base (ALL)	**9**	0.663	0.728	0.710	0.767	9
		xlm-roberta-base	**2**	0.584	0.631	0.646	0.691	10

Table 2: Results of sub-task A for each model and each language.

in relative ranks of labels for that subtasks when combined with labels from other subtasks, i.e. p(**NAG-GEN**) $>$ p(**CAG-GEN**) does not guarentee that p(**NAG-NGEN**) $>$ p(**CAG-NGEN**). Hence, we introduce a marignalized post processing of label to get the total probablity assigned to labels of a given subtasks by marignalizing probabilities across all other subtask labels. This can be done very easily by just summing the combined labels of a particular task label, **Eg.** the probabilities of **CAG-GEN** and **CAG-NGEN** can be added to get the probability of the label **CAG** for **sub-task A**. This provides a stronger signal for each task label. Then, finally taking the argmax of the marginalized labels of the respective tasks, determines the output label for that task. The models using this technique are labeled with **(M)** in the results table below. We only use this approach for post-processing the label probabilities of the joint model. In future we plan to investigate using this marginalized approach during the training phase.

- **Joint training of different languages (ALL):** This was a technique that we previously did not experiment with in HASOC (Mishra and Mishra, 2019). Currently we do not have models dedicated to many languages, e.g., there are specific pre-trained BERT (Devlin et al., 2019) models for the English language but no such model for Hindi exists. For those languages, our only choice is to utilize a multilingual or cross-lingual model. Furthermore, as the data consisted of social-media posts, which predominantly consists of sentences containing a mix of different languages, we expected the cross-lingual models to perform better than the others. An obvious advantage of using a multi-lingual model is that it can process data from multiple languages, therefore we can train a single model for all of the different languages for each subtask. To do so we combined the datasets of the three languages into a single dataset, keeping track of which text came from which language. This can easily be done by flagging the respective id with the respective

lang	task	model	run_id	Macro-F1 dev	Macro-F1 train	Weighted-F1 dev	Weighted-F1 train	rank
ENG	**B**	**bert-base-uncased (M)**	**4 (M)**	0.757	0.920	0.943	0.978	1
		xlm-roberta-base (ALL)	**9**	0.765	0.878	0.941	0.968	2
		bert-base-multilingual-uncased (ALL) (M)	**9 (M)**	0.760	0.939	0.940	0.983	3
		bert-base-uncased (C)	**4 (C)**	0.734	0.914	0.939	0.977	4
		bert-base-cased (C)	**3 (C)**	0.729	0.931	0.939	0.982	5
		bert-base-multilingual-uncased (ALL) (C)	**9 (C)**	0.752	0.936	0.938	0.983	6
		bert-base-cased (M)	**3 (M)**	0.727	0.935	0.938	0.983	7
		bert-base-uncased	**2**	0.737	0.991	0.938	0.998	8
		bert-base-multilingual-uncased (ALL)	**9**	0.751	0.987	0.937	0.996	9
		xlm-roberta-base	**5**	0.734	0.915	0.936	0.978	10
		xlm-roberta-base (M)	**6 (M)**	0.728	0.807	0.934	0.948	11
		xlm-roberta-base (C)	**6 (C)**	0.711	0.813	0.933	0.952	12
		bert-base-cased	**1**	0.700	0.982	0.929	0.995	13
HIN	**B**	**bert-base-multilingual-uncased**	**6**	0.780	0.974	0.891	0.986	1
		bert-base-multilingual-uncased (ALL)	**9**	0.778	0.990	0.888	0.994	2
		bert-base-multilingual-uncased (ALL) (C)	**9 (C)**	0.783	0.932	0.888	0.962	3
		bert-base-multilingual-uncased (ALL) (M)	**9 (M)**	0.778	0.931	0.886	0.962	4
		bert-base-multilingual-uncased (M)	**3 (M)**	0.760	0.844	0.882	0.916	5
		bert-base-multilingual-uncased (C)	**3 (C)**	0.750	0.847	0.874	0.917	6
		xlm-roberta-base (ALL)	**9**	0.745	0.831	0.870	0.909	7
		xlm-roberta-base	**2**	0.459	0.455	0.778	0.759	8
		xlm-roberta-base (C)	**4 (C)**	0.459	0.455	0.778	0.759	8
		xlm-roberta-base (M)	**4 (M)**	0.459	0.455	0.778	0.759	8
IBEN	**B**	**bert-base-multilingual-uncased (ALL)**	**9**	0.849	0.987	0.905	0.992	1
		bert-base-multilingual-uncased (ALL) (M)	**9 (M)**	0.849	0.943	0.904	0.965	2
		bert-base-multilingual-uncased (ALL) (C)	**9 (C)**	0.846	0.943	0.902	0.966	3
		bert-base-multilingual-uncased	**6**	0.830	0.975	0.894	0.985	4
		bert-base-multilingual-uncased (M)	**3 (M)**	0.827	0.924	0.892	0.954	5
		bert-base-multilingual-uncased (C)	**3 (C)**	0.824	0.923	0.890	0.953	6
		xlm-roberta-base (ALL)	**9**	0.792	0.845	0.873	0.908	7
		xlm-roberta-base (M)	**4 (M)**	0.783	0.835	0.869	0.903	8
		xlm-roberta-base (C)	**4 (C)**	0.783	0.833	0.868	0.902	9
		xlm-roberta-base	**2**	0.714	0.743	0.830	0.855	10

Table 3: Results of sub-task B for each model and each language.

language name. This increases the size of the dataset which is beneficial for training deep learning models. We then fine-tuned the pre-trained multilingual model for our dataset. After training, we can separate the dataset based on their language id. Thus resulting in a single model that is able to classify data from all of the three languages. This can be especially useful in deploying situations as this results in models which are resource friendly. The models using this technique are labeled with **(ALL)** in the results table below.

- **Combining the above three techniques:** Finally, we also experimented with combining all of the above three techniques. This results in a single model that can be used for all of the six sub-tasks. Thus, this technique is very efficient in terms of resources used and flexibility. The models using this technique are labeled either **(ALL) (M)** or **(ALL) (C)** in the results table below, based on the presence and absence of the marignalization approach, respectively.

4.3. Training

For training our models we used the standard hyperparameters as mentioned in the transformers models. We used the Adam optimizer (with $\epsilon = 1e-8$) for five epochs, with a training/eval batch size of 32. Maximum allowable length for each sequence is 128. We use a learning rate of $5e-5$ with a weight decay of 0.0 and a max gradient norm of 1.0. All models were trained using Google Colab's [3] GPU runtimes.

4.4. Results and Experiments

For each language and each sub-task we experimented with different pre-trained transformer language models present in the transformers library using the various fine-tuning techniques mentioned in the previous section. The different models with their respective dev and training weighted-F1 and macro-F1 scores for sub-task A and sub-Task B are given in **Table 2** and **Table 3** respectively. The table fol-

[3] https://colab.research.google.com/

lang	task	model	weighted-F1		rank		Overall
			dev	test	dev	test	Rank
ENG	**A**	**bert-base-multilingual-uncased (ALL)**	0.798	0.728	1	3	-
		bert-base-uncased (C)	0.795	0.759	2	2	-
		bert-base-uncased (M)	0.795	0.759	3	1	**2**
		Overall Best Model****	-	0.802	-	-	**1***
HIN	**A**	**bert-base-multilingual-uncased**	0.708	0.778	1	3	-
		bert-base-multilingual-uncased (ALL) (C)	0.696	0.779	2	1	**3**
		bert-base-multilingual-uncased (ALL) (M)	0.695	0.778	3	2	-
		Overall Best Model****	-	0.812	-	-	**1***
IBEN	**A**	**bert-base-multilingual-uncased (ALL)**	0.737	0.780	1	1	**3**
		xlm-roberta-base (M)	0.732	0.772	2	2	-
		xlm-roberta-base (C)	0.731	0.772	3	3	-
		Overall Best Model****	-	0.821	-	-	**1***
ENG	**B**	**bert-base-uncased (M)**	0.978	0.857	1	1	**4**
		xlm-roberta-base (ALL)	0.968	0.844	2	2	-
		bert-base-multilingual-uncased (ALL) (M)	0.983	0.843	3	3	-
		Overall Best Model****	-	0.871	-	-	**1***
HIN	**B**	**bert-base-multilingual-uncased**	0.986	0.837	1	3	-
		bert-base-multilingual-uncased (ALL)	0.994	0.849	2	1	**3**
		bert-base-multilingual-uncased (ALL) (C)	0.962	0.843	3	2	-
		Overall Best Model****	-	0.878	-	-	**1***
IBEN	**B**	**bert-base-multilingual-uncased (ALL)**	0.992	0.927	1	1	**3**
		bert-base-multilingual-uncased (ALL) (M)	0.965	0.926	2	2	-
		bert-base-multilingual-uncased (ALL) (C)	0.902	0.925	3	3	-
		Overall Best Model****	-	0.938	-	-	**1***

Table 4: Test results of the submitted models

lows the following convention to describe the fine-tuning technique used in each experiment. We submitted the top three models based on the weighted-F1 scores on the dev dataset.

- **No label:** This represents the simple fine-tuning approach.
- **(C):** Joint label training
- **(M):** Marginalization of labels
- **(ALL):** Joint training of different languages.
- **(ALL) (C):** Joint training of different languages with joint label training.
- **(ALL) (M):** Joint training of different languages with joint label training and marginalization of labels.

4.5. External Evaluation

We were only provided with the weighted-F1 scores of the three submitted models in each task. Hence, only those results are mentioned in Table 4. Based on the final leaderboard, our models were ranked second in 1/6 task, third in 4/6 tasks, and 4/6 in 1/6 tasks.

5. Discussion

On the basis of the various experiments conducted using the many transformer models, we see that most of them give a similar performance, being within $2-3\%$ of the best model. Exception being the **xlm-roberta-base** (Liu et al., 2019) model which showed appreciable variations. It performed extremely poorly in the Hindi sub-tasks, but with the joint training with different languages its performance increased significantly. Using the joint label training technique it performed really well in the Bengali sub-tasks whilst also being the bottom performer with the other techniques. One important thing to notice is that the joint training with different language fine-tuning technique **(ALL)** works really well. It was a consistent top performing model in our experiments, being the best for Bengali. In most cases, we can see that the **(ALL)** models were better than the base model without any marginalization or joint-training. The marginalization scheme does not change the results much from the joint label training approach. A major benefit of using joint training with different languages, is that is significantly reduces the computational cost of the usage of our models, as we have to only train a single model for multiple tasks and languages, so even if there is a slight performance drop in the **(ALL) (C) or (M)** model compared to the single model, usage of the **(ALL) (C) or (M)** model should still be preferred for its computational efficiently. Our team came second in English sub-task A, a close fourth in the English sub-task B and third in the remaining 4 sub-tasks.

6. Conclusion

From the experiments conducted for this year's TRAC (Ritesh Kumar and Zampieri, 2020) shared tasks, we see that the **(ALL)** models provide us with an extremely pow-

erful approach which gives us a single model capable of classifying texts across all the six shared sub-tasks. We have presented our team 3Idiots's (our team is referred as 'sdhanshu' in the rankings(Ritesh Kumar and Zampieri, 2020)) approach based on fine-tuning monolingual and multi-lingual transformer networks to classify social media posts in three different languages for Trolling, Aggression and Cyber-bullying content. We open source our approach at: https://github.com/socialmediaie/TRAC2020

7. Bibliographical References

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mandl, T., Modha, S., Patel, D., Dave, M., Mandlia, C., and Patel, A. (2019). Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*, December.

Mishra, S. and Diesner, J. (2016). Semi-supervised Named Entity Recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212, Osaka, Japan. The COLING 2016 Organizing Committee.

Mishra, S. and Diesner, J. (2018). Detecting the Correlation between Sentiment and User-level as well as Text-Level Meta-data from Benchmark Corpora. In *Proceedings of the 29th on Hypertext and Social Media - HT '18*, pages 2–10, New York, New York, USA. ACM Press.

Mishra, S. and Diesner, J. (2019). Capturing Signals of Enthusiasm and Support Towards Social Issues from Twitter. In *Proceedings of the 5th International Workshop on Social Media World Sensors - SIdEWayS'19*, pages 19–24, New York, New York, USA. ACM Press.

Mishra, S. and Mishra, S. (2019). 3Idiots at HASOC 2019: Fine-tuning Transformer Neural Networks for Hate Speech Identification in Indo-European Languages. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*.

Mishra, S., Agarwal, S., Guo, J., Phelps, K., Picco, J., and Diesner, J. (2014). Enthusiasm and support: alternative sentiment classification for social movements on social media. In *Proceedings of the 2014 ACM conference on Web science - WebSci '14*, pages 261–262, Bloomington, Indiana, USA, jun. ACM Press.

Mishra, S., Diesner, J., Byrne, J., and Surbeck, E. (2015). Sentiment Analysis with Incremental Human-in-the-Loop Learning and Lexical Resource Customization. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media - HT '15*, pages 323–325, New York, New York, USA. ACM Press.

Mishra, S. (2019). Multi-dataset-multi-task Neural Sequence Tagging for Information Extraction from Tweets. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media - HT '19*, pages 283–284, New York, New York, USA. ACM Press.

Ritesh Kumar, Atul Kr. Ojha, S. M. and Zampieri, M. (2020). Evaluating aggression identification in social media. In Ritesh Kumar, et al., editors, *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*, Paris, France, may. European Language Resources Association (ELRA).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., and Margetts, H. (2019). Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy, August. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing.