3Idiots at HASOC 2019: Fine-tuning Transformer Neural Networks for Hate Speech Identification in Indo-European Languages

Shubhanshu Mishra[1][0000−0001−9931−1690] and Sudhanshu Mishra[2][0000−0003−0552−8659]

[1] iSchool, University of Illinois at Urbana-Champaign, Champaign IL 61820, USA
smishra8@illinois.edu
[2] IIT Kanpur UP 208016, India
sdhanshu@iitk.ac.in

Abstract. We describe our team **3Idiots**'s approach for participating in the 2019 shared task on hate speech and offensive content (HASOC) identification in Indo-European languages. Our approach relies on fine-tuning pre-trained monolingual and multilingual transformer (BERT) based neural network models. Furthermore, we also investigate an approach based on labels joined from all sub-tasks. This resulted in good performance on the test set. Among the eight shared tasks, our solution won the first place for English sub-tasks A and B, and Hindi sub-task B. Additionally, it was within the top 5 for 7 of the 8 tasks, being within 1% of the best solution for 5 out of the 8 sub-tasks. We open source our approach at https://github.com/socialmediaie/HASOC2019.

Keywords: Hate Speech Identification · Offensive Content Identification · Neural Networks · BERT · Transformers · Deep Learning.

1 Introduction

Information extraction from social media data is an important topic. In the past we have used it for identifying sentiment in tweets [5] [7], enthusiastic and passive tweets and users [3] [6], and extracting named entities [2] [4]. The hate speech and offensive content (HASOC) shared task of 2019 focused on Indo-European languages, gave us an opportunity to try out BERT [1] for this shared task. BERT based pre-trained transformer based neural network models are publicly available in multiple languages and the model supports fine-tuning for specific tasks. We also tried a joint-label based approach called shared-task D, which alleviates data sparsity issues for some shared tasks, while achieving competitive performance in the final leader board evaluation.

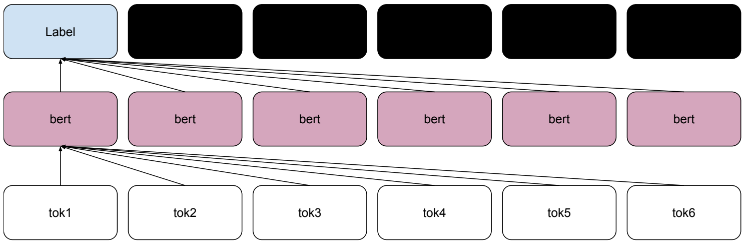

Fig. 1. Model illustration. *tok* items are tokens of the post, *bert* is the representation extracted using BERT, and *Label* is the classification label for the task.

task	**DE**			**EN**			**HI**		
	train	dev	test	train	dev	test	train	dev	test
A	3819	794	850	5852	505	1153	4665	136	1318
B	407	794	850	2261	302	1153	2469	136	1318
C				2261	299	1153	2469	72	1318

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

2 Data

The data supplied by the organizing team, consisted of posts taken from Twitter and Facebook respectively. The posts were in the following three languages : English (EN), German (DE) and Hindi (HI). The competition had three sub-tasks for English data, two sub-tasks for German data and three sub-tasks for Hindi data respectively. **Sub-Task A** consisted of labeling a post with **(HOF) Hate and Offensive**, if the post contained any hate speech, profane content, or offensive content, otherwise the label should be **(NOT) Non Hate-Offensive**. Next, **Sub-Task B** was more fine grained, and specified identification of **(HATE) Hate Speech**, **(OFFN) Offensive content**, and **(PRFN) Profane content**. Finally, **Sub-Task C** focused on identifying via label **(TIN) Targeted Insult**, if the hate speech, offensive, or profane content (collectively referred to as insult) was targeted towards an individual, group, or other. If content was non-targetted, the label should be **(UNT) Untargeted**. For details about the task, we refer the reader to the shared task publication [8]. The organizers released teaser data, which we utilized as a *dev* dataset for selecting hyperparameters of our models. The distribution of the number of samples for each sub-task in each language is tabulated in table 1. it can be observed that the data-set size for each task is quite small. Table 1 describes the distribution of data in each language under each sub-task.

			split	macro		weighted	
task	lang	model	run_id	dev	train	dev	train
A	DE	bert-base-german-cased		0.783	0.999	0.926	1.000
		bert-base-german-cased (D)	3	**0.811**	0.994	**0.935**	0.998
		bert-base-multilingual-cased	1	0.769	0.840	0.923	0.943
		bert-base-multilingual-cased (D)	2	0.722	0.939	0.909	0.977
	EN	bert-base-cased	1	0.567	0.958	0.558	0.961
		bert-base-cased (D)		0.577	0.877	0.574	0.885
		bert-base-uncased	3	**0.610**	0.964	**0.606**	0.966
		bert-base-uncased (D)	2	0.603	0.902	0.596	0.908
	HI	bert-base-multilingual-cased	2	0.558	0.973	0.552	0.973
		bert-base-multilingual-uncased	1	0.742	0.961	0.742	0.961
		bert-base-multilingual-uncased (D)	3	**0.823**	0.941	**0.822**	0.941
B	DE	bert-base-german-cased	3	**0.486**	0.791	0.455	0.813
		bert-base-german-cased (D)		0.446	0.884	**0.896**	0.984
		bert-base-multilingual-cased	1	0.140	0.247	0.112	0.367
		bert-base-multilingual-cased (D)	2	0.282	0.409	0.865	0.918
	EN	bert-base-cased	1	0.303	0.852	0.338	0.873
		bert-base-cased (D)		0.325	0.806	0.387	0.849
		bert-base-uncased	3	0.314	0.846	0.349	0.867
		bert-base-uncased (D)	2	**0.332**	0.839	**0.401**	0.875
	HI	bert-base-multilingual-cased	2	0.222	0.726	0.264	0.772
		bert-base-multilingual-uncased	1	0.322	0.701	0.466	0.749
		bert-base-multilingual-uncased (D)	3	**0.459**	0.736	**0.757**	0.826
C	EN	bert-base-cased	1	0.542	0.957	0.858	0.985
		bert-base-cased (D)		0.401	0.691	0.537	0.856
		bert-base-uncased	3	**0.627**	0.942	**0.880**	0.980
		bert-base-uncased (D)	2	0.393	0.651	0.548	0.874
	HI	bert-base-multilingual-cased	2	0.550	0.590	0.800	0.635
		bert-base-multilingual-uncased	1	**0.550**	0.866	**0.800**	0.877
		bert-base-multilingual-uncased (D)	3	0.537	0.622	0.769	0.724

Table 2. F1 scores for train and dev splits of the data for the final epoch of model training. Models with suffix **(D)** are trained using Task D.

3 Models

Each sub-task can be modelled as a text classification problem. Our submission models are derived from fine-tuning the pre-trained language model to the shared task data. We used BERT [1] as our pre-trained language model because of its recent success as well as public availability in multiple languages. We utilize the BERT implementation present in pytorch-transfomers library[3]. In order to predict on HI and DE language datasets, we used *bert-multilingual* as well as *bert-german* pre-trained models. Our fine-tuned model is illustrated in figure 1.

[3] https://github.com/huggingface/pytorch-transformers

1. **English Language Task (EN)** - For the English language task we experimented with the *bert-base-cased* and *bert-base-uncased* models. We experimented on all three sub-tasks using the above models.
2. **German Language Task (DE)** - For the German language task we experimented with the *bert-base-german-cased* and bert-base-multilingual-cased models. We experimented on sub-tasks A, and B using the above models.
3. **Hindi Language Task (HI)** - For the Hindi language task we experimented with the *bert-base-multilingual-cased* and *bert-base-multilingual-uncased* models. We experimented on all three sub-tasks using the above models.

4 Training

Our models were trained using the Adam optimizer (with $\epsilon = 1e - 8$) for five epochs, with a training/eval batch size of 32. Finally, each sequence is truncated to max allowed sequence length of 28 characters. We use a learning rate of $5e-5$, weight decay of 0.0, and we also use a max gradient norm of 1.0.

4.1 Training via joint labels - Sub-Task D

In order to alleviate the data sparsity issue we utilize an approach which we call **sub-task D**. Herein, the labels of each sub-task are combined to form a unified multi-label task. All possible class combinations across all sub-tasks are utilized to create new classes. The motivation behind this approach is to share information between tasks via their label combinations, training a single model for this task, followed by post-processing to identify labels for sub-tasks *A, B, and C*. The final set of classes are **NOT-NONE-NONE**, **HOF-HATE-TIN**, **HOF-HATE-UNT**, **HOF-OFFN-TIN**, **HOF-OFFN-UNT**, **HOF-PRFN-TIN**, **HOF-PRFN-UNT**. Furthermore, the approach also addresses data sparsity issue as we use the full training data to solve all tasks.

5 Results

5.1 Internal evaluation of model training

Since, we did not have test labels, we evaluated our model on both the training as well as dev set (as described above). Similar to the shared task evaluation protocol, our evaluation also utilized macro-f1 and weighted f1 scores. Our evaluation is presented in table 2. We selected the best models from each evaluation as our submission for the respective sub-task.

5.2 Evaluation on test data

To identify our model performance on the test data, we utilized the leader board rankings released by the organizers based on on all the shared task submissions (see table 3). Among the eight shared tasks, our solutions won the first place for

task	**lang**	**run_id**	model	macro f1	weighted f1	rank
A	**DE**	**-**	best	0.616	0.791	1
		1	bert-base-multilingual-cased	0.577	0.789	4
		2	bert-base-multilingual-cased (D)	-	-	-1
		3	bert-base-german-cased (D)	0.522	0.778	12
	EN	**-**	best	0.788	0.840	1
		1	bert-base-cased	0.739	0.790	14
		2	bert-base-uncased (D)	0.747	0.801	7
		3	bert-base-uncased	0.740	0.790	11
	HI	**-**	best	0.815	0.820	1
		1	bert-base-multilingual-uncased	0.802	0.802	10
		2	bert-base-multilingual-cased	0.800	0.801	11
		3	bert-base-multilingual-uncased (D)	0.811	0.814	3
B	**DE**	**-**	best	0.347	0.775	1
		1	bert-base-multilingual-cased	0.249	0.756	12
		2	bert-base-multilingual-cased (D)	0.276	0.778	4
		3	bert-base-german-cased	0.274	0.773	6
	EN	**-**	best (ours)	0.545	0.728	1
		1	bert-base-cased	0.517	0.701	3
		2	bert-base-uncased (D)	0.537	0.698	2
		3	bert-base-uncased	0.545	0.728	1
	HI	**-**	best (ours)	0.581	0.715	1
		1	bert-base-multilingual-uncased	0.553	0.688	7
		2	bert-base-multilingual-cased	0.553	0.675	6
		3	bert-base-multilingual-uncased (D)	0.581	0.715	1
C	**EN**	**-**	best (ours)	0.511	0.756	1
		1	bert-base-cased	0.500	0.753	2
		2	bert-base-uncased (D)	0.476	0.764	6
		3	bert-base-uncased	0.511	0.756	1
	HI	**-**	best	0.575	0.736	1
		1	bert-base-multilingual-uncased	0.565	0.727	2
		2	bert-base-multilingual-cased	0.549	0.748	5
		3	bert-base-multilingual-uncased (D)	0.550	0.758	4

Table 3. Evaluation on test data. Models with suffix **(D)** are models trained using task D. Best models which are ours are identified with suffix **(ours)**. Rank of **-1** means no evaluation on the test data is available for this setting.

English sub-tasks A and B, and Hindi sub-task B. Furthermore, it was within the top 5 for 7 of the 8 tasks, being within 1% of the best solution for 5 out of the 8 sub-tasks. For the English sub-task B, our submissions took all the top three ranks. Our submissions also came close second for sub-task C for both English and Hindi.

6 Conclusion

We have presented our team **3Idiots**'s approach based on fine-tuning monolingual and multi-lingual transformer networks to classify social media posts in three different languages, for hate-speech, and offensive content. We open source our approach at: https://github.com/socialmediaie/HASOC2019

References

1. 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 (Jun 2019). https://doi.org/10.18653/v1/N19-1423
2. Mishra, S.: 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. pp. 283–284. ACM Press, New York, New York, USA (2019). https://doi.org/10.1145/3342220.3344929, http://dl.acm.org/citation.cfm?doid=3342220.3344929
3. Mishra, S., Agarwal, S., Guo, J., Phelps, K., Picco, J., Diesner, J.: Enthusiasm and support: alternative sentiment classification for social movements on social media. In: Proceedings of the 2014 ACM conference on Web science - WebSci '14. pp. 261–262. ACM Press, Bloomington, Indiana, USA (jun 2014). https://doi.org/10.1145/2615569.2615667, http://dl.acm.org/citation.cfm?doid=2615569.2615667
4. Mishra, S., Diesner, J.: Semi-supervised Named Entity Recognition in noisy-text. In: Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT). pp. 203–212. The COLING 2016 Organizing Committee, Osaka, Japan (2016), https://aclweb.org/anthology/papers/W/W16/W16-3927/
5. Mishra, S., Diesner, J.: 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. pp. 2–10. ACM Press, New York, New York, USA (2018). https://doi.org/10.1145/3209542.3209562, http://dl.acm.org/citation.cfm?doid=3209542.3209562
6. Mishra, S., Diesner, J.: 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. pp. 19–24. ACM Press, New York, New York, USA (2019). https://doi.org/10.1145/3345645.3351104, http://dl.acm.org/citation.cfm?doid=3345645.3351104
7. Mishra, S., Diesner, J., Byrne, J., Surbeck, E.: 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. pp. 323–325. ACM Press, New York, New York, USA (2015). https://doi.org/10.1145/2700171.2791022, http://doi.acm.org/10.1145/2700171.2791022http://dl.acm.org/citation.cfm?doid=2700171.2791022
8. Modha, S., Mandl, T., Majumder, P., Patel, D.: 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 2019)