CEUR-WS.org/Vol-2517/T3-14.pdf

KMI-Panlingua at HASOC 2019: SVM vs BERT for Hate Speech and Offensive Content Detection*

Ritesh Kumar[1] and Atul Kr. Ojha[2,3]

[1] K.M. Institute of Hindi and Linguistics, Dr. Bhimrao Ambedkar University, India
ritesh78_llh@jnu.ac.in
[2] Panlingua Language Processing LLP, India
[3] Charles University, Prague
shashwatup9k@gmail.com

Abstract. This paper presents KMI-Panlingua's system description which was submitted at the FIRE Shared Task 2019 on Hate Speech and Offensive Content Identification in Indo-European Languages. Our team submitted systems for all the 3 sub-tasks in two languages - English and Hindi. We experimented with 2 kinds of systems - classic machine learning using SVM and BERT-based system. We discuss the systems and their results in this paper.

Keywords: Hate Speech· Offensive Language· Hindi· English· SVM· BERT.

1 Introduction

In the digital era, social media such as Facebook, Twitter, WhatsApp, etc, is one of the most important mediums to circulate information as well as communicate in the society. While it helps in quickly spreading the information in society, it has also become hotbeds for the spread of hate speech and offensive contents. The hate speech and offensive content could range from political and religious to caste and gender or any issue that could divide and polarise a community. So, it is required to build a robust automatic hate speech and offensive content detection system which may help to filter out these types of content such that they do not spread in the community.

There have been a lot of efforts towards building such systems (notably [6], [9], [2], [10]). There have also been some shared tasks that have been organised around the automatic detection of offensive language and aggression on social media [5], [11].

The FIRE 2019 shared task on Hate Speech and Offensive Content Detection in Indo-European Languages (HASOC 2019) is another such effort in this direction. In this paper, we discuss the development of automatic hate speech and

* Copyright©2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). FIRE 2019, 12-15 December 2019, Kolkata, India.

offensive content identification systems for all the 3 sub-tasks in two languages - English and Hindi - as part of this shared task. We experimented with 2 kinds of systems - classic machine learning using SVM and BERT-based system - to explore their relative applicability for the task.

The rest of the paper is divided into four section. Section 2 discusses of the dataset size and its types. Section 3 provides a detailed description of the conducted experiments. While section 4 reports the developed systems' results and their error analysis.Section 5 ends with the concluding remarks.

2 Dataset

In order to conduct the experiments, we used the data for English and Hindi languages which were shared in the FIRE Shared Task HASOC 2019[7]. A statistics of the data is given in Table 1. The data is labelled at 3 levels and they were presented as 3 sub-tasks as given below -

1. **Sub-task 1:** In this sub-task, the data is annotated as HOF and NOT. HOF stands for Hates speech and Offensive Language while NOT is not offensive. Thus it is a binary classification task.
2. **Sub-task 2:** If the content is marked HOF in the first sub-task then it is marked as Profanity (PRFN), Offensive (OFFN) or Hate Speech (HATE) in this stage. It is a 3-class classification problem.
3. **Sub-task 3:** The HOF contents are also marked for whether they are targeted towards an individual or a group (TIN) or not (UNT).

No additional dataset or resources (except the BERT pre-trained models) have been used for the task.

Table 1. The HASOC Dataset

	Train Sub-task 1			**Train Sub-task 2**				**Test Sub-task 3**			**Test Set**
	TOTAL	**HOF**	**NOT**	**TOTAL**	**HATE**	**PROFN**	**OFFN**	**TOTAL**	**TIN**	**UNT**	**TOTAL**
EN	**5,852**	2,261	3,591	**2,261**	1,143	667	451	**2,261**	2,041	220	**1,153**
HI	**4,665**	2,469	2,196	**2,469**	556	1,237	676	**2,469**	1,545	924	**1,318**

3 Experimental Setup

We experimented with broadly two kinds of systems - a SVM classifier and a BERT-based classifier for sub-task 1 and 3 of English dataset. We used the scikit-learn implementation of SVM ([8], [1]) and Fast-Bert library [4] (which itself is

[4] `https://github.com/kaushaltrivedi/fast-bert`

based on Hugging Face pytorch-transformers library [5]) for the experiments. The implementation details of the Fast-Bert are given in the two blogs by the author [6], [7].

Support Vector Machines [4] are one of the most successful classic machine learning models used for various kinds of text classification tasks. On the other hand, BERT (Bidirectional Encoder Representations from Transformers) [3] makes use of a masked language model (MLM), which enables it to fuse the left and right context, thereby, allowing to pre-train a deep bidirectional Transformer. These pre-trained models could be fine-tuned for specific tasks. BERT models are demonstrated to have given significant improvements in several NLP tasks including general language understanding, question answering, next sentence prediction / text generation as well as some text classification tasks. The main aim of our experiments was to explore the usefulness and efficacy of BERT vis-a-vis SVMs and see if BERT could be helpful in the specific task of offensive and hate speech detection.

3.1 Experiments with SVM

For SVM, we used 5-fold cross-validation for figuring out the optimum model. We experimented with the following sets of features -

1. Word n-grams (unigrams, bigrams and trigrams)
2. Character n-grams (trigrams to 5-grams)
3. A combination of different word n-grams and character n-grams features

A gird search was performed for C-values from 0.0001 upto 10 (with a 10x interval in between two C-values) for each of the feature combination and each of the sub-tasks. The classifiers that gave the best performance for each sub-task in each language are give in Table 2.

Table 2. Comparison of character and word n-gram features for best SVM classifier

	Sub-task 1		**Sub-task 2**		**Sub-task 3**	
	EN	**HI**	**EN**	**HI**	**EN**	**HI**
Character n-grams	3	3, 4, 5	3, 4, 5	3, 4	3, 4	3
Word n-grams	1, 2, 3	1	1, 2, 3	1	1, 2, 3	1

As we could see, a combination of word n-grams and character n-grams have given the best performance in all the cases. It is apparent from the table, as

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

[6] `https://medium.com/huggingface/introducing-fastbert-a-simple-deep-learning-library-for-bert-models-89ff763ad384`

[7] `https://medium.com/huggingface/multi-label-text-classification-using-bert-the-mighty-transformer-69714fa3fb3d`

expected, word n-grams are not very helpful in case of Hindi while for English using upto trigrams gives the best performance in all the three sub-tasks. It is expected since Hindi is expected to have much more morphological information than English and those are generalised by the use of character n-grams. Moreover, character 5-grams are expected to be almost equivalent to word unigrams (or at least stemmed word n-grams) and so, in English, adding more than character trigrams in the first task does not add any new information. For the other two sub-tasks, the performance of the SVM classifiers are not as good as that of the first sub-task and the classifiers fail to generalise well enough for both the training as well as test dataset. As such the best performance of one particular classifier might be incidental. Moreover, it must also be noted that the difference between classifiers trained using different feature sets was not huge and only a marginal improvement was noticed with different word-level and character-level feature combinations.

3.2 Experiments with BERT

We could experiment with BERT in only sub-task 1 and 3 for English using BERT (because of the lack of sufficient hardware resources required for fine-tuning the BERT models for other sub-tasks as well as for Hindi). We fine-tuned the pre-trained BERT-base-uncased model released by Google for this task. The fine-tuning was carried out on a standard Google Colab GPU system. For both the sub-tasks, the models were trained for 10 epochs and used the LAMB optimizer.

4 Results and Error Analysis

The overall results on the test set show that for sub-task 1 in English, BERT substantially outperforms SVM by a huge margin. However, for sub-task 3, it fails to perform at par with the SVM. In general, classifiers for sub-task 1 is able to perform much better than those for sub-task 2 (which was a 3-class classification problem) and sub-task 2 (which was also a binary classification problem, like sub-task 1) [7]. Our BERT-based classifier is placed at 12th position in English sub-task 1 (and the macro F1 score is 5 points below that of the top team), while the SVM classifier is placed at the 49th position (with an overall difference of over 17 points in the micro F1 in comparison to the BERT system). However, for sub-task 3 in English (the only other sub-task where we could experiment with BERT), the situation is opposite of this. Our SVM classifier is placed at the 19th position (macro F1 score being almost 10 points below the best team), BERT is placed at the 27th position (with a difference of almost 5 points in macro F1 in comparison to the SVM system). The performance of the two systems in these two sub-tasks are summarised in the Fig 1.

Besides these, in other cases where we experimented with only SVM-based classifier, while our overall rank was relatively good, the difference in macro F1 scores of our system and that of the top team hovered from 6 points (Hindi sub-task 1) to 10 points (English sub-task 2). These comparisons are summarised in the Fig 2.

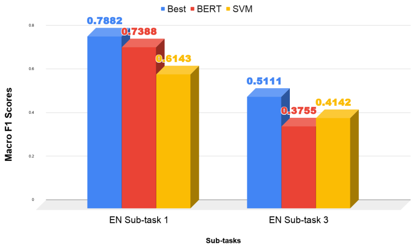

Fig. 1. Performance of BERT system vis-a-vis SVM and the best system

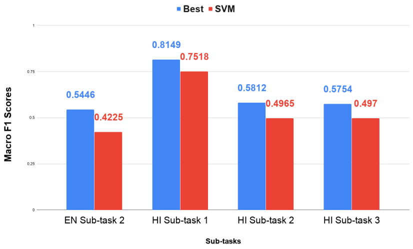

Fig. 2. Performance of SVM system vis-a-vis the best system

In addition to this big picture, if we take a closer look at the kind of errors our system has produced, it is quite apparent that the BERT system has better generalised and has led to an improvement in the precision score while maintaining a good recall, leading to an overall improvement in the performance (see Fig 3 and Fig 4 for a comparison of SVM and BERT systems for English sub-task 1).

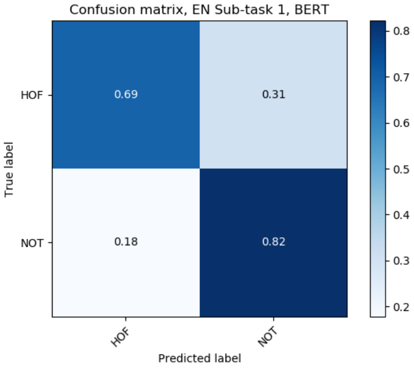

Fig. 3. Confusion Matrix for English sub-task 1, trained using BERT

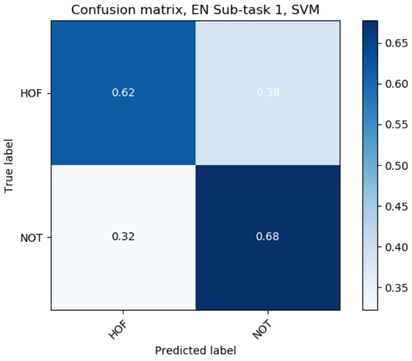

Fig. 4. Confusion Matrix for English sub-task 1, trained using SVM

In the sub-task 3 [8] (see Fig 5 and Fig 6), the picture is little more complicated. There is an improvement in the precision of both the classes using the BERT system. However, since the total number of training instances are low, the features for classification are less. Thus the precision is extremely low for both the classes, more so for the minority class. Moreover, since the dataset is highly imbalanced, there are not sufficient discriminating features. And thus the recall is also very low for both the classes. However, what is quite interesting to note is that for SVM, recall is extremely low for the minority class (UNT), while in BERT, it is exactly the opposite - the majority class gets a very low recall while the recall for the minority class is quite good. This shows a fundamentally different learning pattern for deep learning systems like BERT and thereby depicting a better tendency for generalisation.

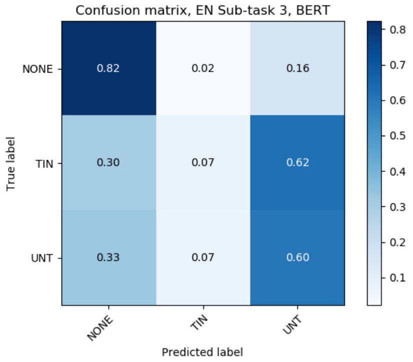

Fig. 5. Confusion Matrix for English sub-task 3, trained using BERT

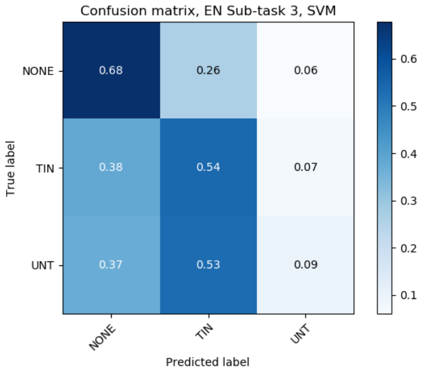

Fig. 6. Confusion Matrix for English sub-task 3, trained using SVM

Among the other tasks, the lack of sufficient training instances as well as the availability of explicit, lexical features seem to have played a major role in the low performance in those tasks. In sub-task 2 of English (Fig 7), PRFN is defined in a way that they have more explicit, lexical level features which could be generalised by a classifier like SVM, hence, a better performance than HATE and OFFN despite HATE being substantially higher in number. A similar trend is noticed in Hindi dataset where the performance is best in task 1 (Fig 8), which had a relatively large training sample. In sub-task 2 (Fig 9), PRFN is the best-

[8] For sub-task 2 and 3, the category the classifiers were not trained for predicting NONE, as per the instructions given by the task organisers. However, only one test file was given for testing and it was compulsory to give a prediction for each sample. So we gave a NONE to all those test samples that were classified as NOT in sub-task 1. So the errors related to NONE are not made by the classifiers for sub-task 2 and 3; rather these errors have percolated because of misclassifications in sub-task 1 and they should be interpreted as such in the confusion matrices for these two sub-tasks

performing class, while in sub-task 3 (Fig 10), the majority class performs the best. Among the two languages, the classifiers for Hindi seems to be performing relatively better because of the greater number of training samples as well as relatively more balanced dataset (especially in sub-task 3).

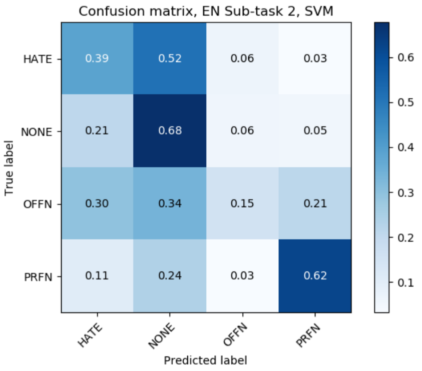

Fig. 7. Confusion Matrix for English sub-task 2, trained using SVM

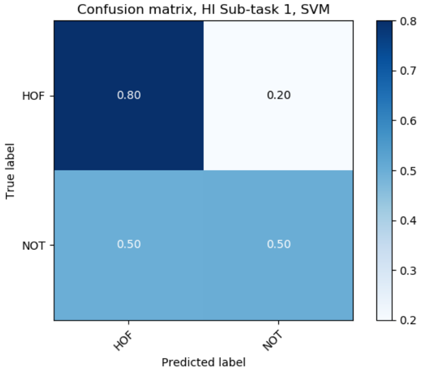

Fig. 8. Confusion Matrix for Hindi sub-task 1, trained using SVM

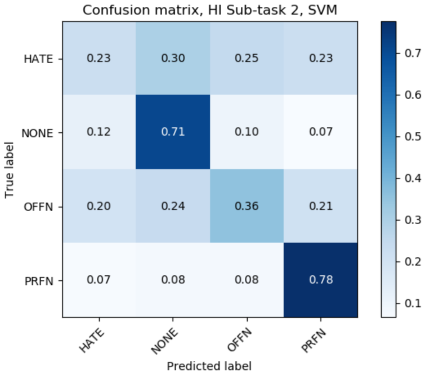

Fig. 9. Confusion Matrix for Hindi sub-task 2, trained using SVM

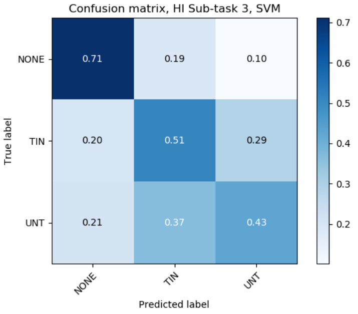

Fig. 10. Confusion Matrix for Hindi sub-task 3, trained using SVM

5 Conclusion

In this paper, we have given a description of the KMI-Panlingua system developed for HASOC at FIRE 2019. Our analysis of the results show that BERT is able to generalise better for the task than SVM and even in cases of unbalanced dataset, BERT is able to achieve high recall (but low precision) even for the minority class (with very little training samples). This depicts a fundamental difference in the way a linear classifier like SVM and BERT learns. In addition to this, the low performance in sub-task 2 and 3 could be largely attributed to the unbalanced dataset and the absence of sufficient training samples for different classes. A more balanced dataset with large learning samples for each class might produce better results in these instances.

References

1. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
2. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of ICWSM (2017)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Hearst, M.A.: Support vector machines. IEEE Intelligent Systems **13**(4), 18–28 (Jul 1998). https://doi.org/10.1109/5254.708428, http://dx.doi.org/10.1109/5254.708428
5. 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)
6. Malmasi, S., Zampieri, M.: Challenges in discriminating profanity from hate speech. Journal of Experimental & Theoretical Artificial Intelligence **30**, 1 – 16 (2018)
7. 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)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
9. Waseem, Z., Davidson, T., Warmsley, D., Weber, I.: Understanding abuse: A typology of abusive language detection subtasks. In: Proceedings of the First Workshop on Abusive Language Online. pp. 78–84. Association for Computational Linguistics (2017), http://aclweb.org/anthology/W17-3012
10. 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 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology (NAACL-HLT) (2019)
11. 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 (SemEval) (2019)