Vito at HASOC 2019: Detecting Hate Speech and Offensive Content through Ensembles

Victor Nina-Alcocer

Department of Computer Systems and Computation Universitat Politècnica de València Camí de Vera s/n 46022 València, Spain vicnial@inf.upv.es

Abstract. This paper describes our participation in the shared task "Hate Speech and Offensive Content Identification in Indo-European Languages" (HASOC) at the Forum for Information Retrieval Evaluation (FIRE) 2019. This work studies the detection of hate or offensive content on English posts published on Facebook or Twitter. For a finegrained study of the task, we analyzed two different approaches: the first one regards the design of two architectures using convolutional and recurrent neural networks. Meanwhile, the second approach examines a range of paradigms based on classical machine algorithms, neural networks, and transformers.

Keywords: Facebook \cdot Twitter \cdot Hate Speech \cdot Offensive Content \cdot Convolutional Neural Networks \cdot Recurrent Neural Networks \cdot Transformers.

1 Introduction

HASOC[12] at FIRE[2] aims to identify hate and offensive content in social media, taking into account tweets and Facebook posts for Indo-European languages. To accomplish this goal, they propose three sub-tasks that tackle this issue in three languages: English, German and, Hindi.

- Sub-task A: Hate or Offensive Identification.
- Sub-task B: Category Classification.
- Sub-task C: Target Identification.

The aim of sub-task A is to classify short-posts into two classes: "Hate and Offensive" (HOF) and "Non-Hate and Non-Offensive" (NOT). For the HOF posts, sub-task B implements further fine-grained classifications to categorize them into three groups: Hate speech (HATE), Offensive (OFFN) and Profane (PRFN). Furthermore, sub-task C focuses on whether a HOF post is targeting a particular individual/group (Targeted Insult, TIN) or just generally profane (Untargeted, UNT). We can find further information about these tasks on [12].

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.

Victor Nina-Alcocer et al.

As we did in previous researches [13–15], we put our effort to deal with these kinds of challenges using Machine Learning (ML) and Natural Language Processing (NLP). In this paper, we will use the same fields but digging a little deeper in Deep Learning (DL) [10], Neural Networks (NN) [11], and architectures based on transformers [19].

We organize this paper into four sections. The first section provides an introduction to this paper, followed by descriptions of the proposed approaches and their respective systems. Meanwhile, the third section presents many experiments, and those results achieved. The final section concludes our work.

2 Systems Description

This section presents the main approaches that have been taken into account in this research.

The first approach considers two architectures. The first one is DL-NN, which utilizes Convolutional Neural Networks (CNNs) [8] and Recurrent Neural Networks (RNNs). It has three embedding layers as inputs. The first layer (I1) takes into account the embedding of a pre-processed post. The following two layers: consider the embedding of its Part of Speech (POS) tagging [3, 16] (I2) as well as the existence of positive or negative words, according to a pre-defined lexicon [7] (I3). After that, the first embedding input feeds a CNN layer, which is followed by an LSTM layer (O1). The second and third input layers feed a CNN (O2) and a dense layer (O3) respectively. The outputs of these three layers (O1, O2, O3) are concatenated and feed to a dense layer, which compounded by two nodes. Softmax is used to get the predictions.

Our second architecture is NN-AA. It implements Long short-term memory (LSTM) [21] layers, which is a typical type of RNN. It takes the embedding representation of its respective pre-processed post, together with another feature input extracted from the tweet's POS tagging. It is essentially an embedding layer followed by an LSTM layer. The model also incorporates an attention layer [5] to pay focus on critical words in the sentence. We concatenate the output of this attention layer to the POS tagging vector representation. The vector is calculated as the sum of the frequency for each word's POS tag divided by the total number words of the sentence. We feed the concatenated vector into a dense layer that implements softmax function and generates predictions.

We considered another approach, which is called MA-MO. This approach examines a variety of models, that go from traditional Machine Learning, i.e. Support Vector Machine [1], Logistic Regression [20] and Naive Bayes [18] models to some Deep Learning models i.e., a simple CNN, a simple Dense layer (SDL), or a Multi-Layer Perceptron (MLP) [9]. We also considered the fine-tunning of some architectures based on slightly modified transformers. We combined the results of our systems to feed an ensemble classifier. Detecting Hate Speech and Offensive Content through Ensembles

3 Experiments and Results

In this section, we comment on all the experiments and the results achieved. HASOC provides a training set of approximately 6000 posts written in English. Table 1 shows that sub-task A has 5852 posts, and they are labeled as HOF (Hate and offensive) and NOT (Non-Hate-Offensive). Sub-task B and sub-task C each have 2261 posts. HATE (Hate speech), OFFN (Offensive), and PRFN (Profane) labels are for sub-task B. TIN (Target insult) and, UNT (Untargeted) are for sub-task C.

Table 1. Summary of the English training set provided by the organizers.

	SubtaskA		${f SubtaskB}$			SubtaskC	
	HOF	NOT	HATE	OFFN	PRFN	TIN	UNT
Num. posts	2261	3591	1143	451	667	2041	220
Total	5852		2261			2261	

Table 1 also shows that the dataset for sub-task A is slightly balanced. However, the others are not, especially for PRFN, OFFN, TIN and UNT classes in their respective datasets. We tackled this issue by performing data augmentation to the posts in the unbalanced categories and generates an acceptable training set. Furthermore, to this set, the organizers provided an unlabeled test set of almost 1000 posts to test our systems.

To get a development set, we split the training set into two parts. So, we used 10% of the training set for tunning our parameters. The official metrics to evaluate the proposed systems for these three sub-tasks were F1 macro, and F1 weighted[12].

We applied the following text pre-processing to the posts that are input to our systems: we remove URLs, numbers, users, times, dates, emails, and percents. Besides, we normalize hashtags (i.e., Converting "#TrumpIsATraitor" to "trump is a traitor"), emojis with its CLDR version, elongated words (i.e., Transforming "haaaaahaa" to "haha"), repetitions, emphasis, and censored words. Both proposed approaches use the text pre-processing mentioned above. Additionally, the first approach and part of the second approach (traditional ML and DL models) uses Glove embeddings[17]. Meanwhile, we feed the architectures based on transformers with pre-processed data only.

Regarding our first approach, DL-NN allows us to try a variety of settings. To get some patterns in n-gram words, we use CNN layers, with 256 filters and various sizes of kernels (1, 2, 3, and 4) to find the optimum. As we describe in section two, we use an LSTM for capturing long-term dependencies after the CNN layers. GRU [6] and BiLSTM [4] layers are our first attempt, but LSTM gives much better results. In short, our architecture is defined as we mentioned in section two, it used a kernel size of 2 for both CNNs (O1, O2) and 128 nodes for the dense layer (O3). In our second stage, it utilizes 128 nodes for LSTM. Next, we concatenate the outputs to feed a dense fully-connected layer of 64

Victor Nina-Alcocer et al.

nodes. Finally, a layer with two nodes and softmax is used to get predictions. The dimensions of word embeddings are 200. We used Adam optimizer with a learning rate of 0.01, and 50 epochs were run.

In contrast to DL-NN, we implement the attention mechanism in NN-AA architecture, which particularly focuses on important parts (words) of the sentence (post). For the attention layer, we experimented with GRU, BiLSTM, and LSTM structures, and LSTM outperforms the others. Another feature that boosts a few of our results was the addition of a vector POS tagging representation. To get this last feature, we took the main combinations of tags, i.e verbs or adjectives followed by a noun. These combinations are some of the most repeated patterns in the training dataset. We concatenated the vector pattern representation with the output of the attention layer. A dropout rate of 0.2 was used, but the results did not get better. Then we just used softmax to make the predictions.

We carried out many experiments with various setups for the second approach (MA-MO). The same features as with the other two architectures, together with Glove word embeddings, are tested on conventional ML models. Most models have experimented with default parameter settings. For the models that use CNN (with 128 filters and a kernel size of 2), SDL (with 128 nodes), or MLP (256 nodes), we just fed them with embeddings mentioned above (with the dimension of 200). Meanwhile, the fine-tuning models used raw tokenized posts as inputs. There are two nodes in the output layer for all models mentioned, and the softmax function for generating predictions.

	SubtaskA		Subta	askB	SubtaskC		
	F1 macro	F1 weig.	F1 macro	F1 weig.	F1 macro	F1 weig.	
			0.28458				
NN-AA	0.70324	0.71982	0.35568	0.77016	0.48912	0.73466	
MA-MO							
ENSEM	0.78093	0.84193	0.55161	0.81019	0.51410	0.83994	

Table 2. Results for English development dataset for SubtaskA, SubtaskB and SubtaskC.

As we can see, Table 2 shows a summary of the results that each one of the proposed approaches achieved. We can observe that DL-NN has the worst performance. However, when we incorporate the Attention layer (NN-AA) and the vector POS representation, the performance increases further. Furthermore, the performances significantly improved with the ensemble models (MA-MO). We combined the first two architectures with MA-MO to get the ENSEM model. And clearly, the last one outperforms the others.

3.1 Official Results

As we commented on in section 3, HASOC provides a test set of approximately 1000 unlabeled posts to evaluate our systems. We show the results of such an

evaluation in Table 3. For each one of the sub-tasks, three runs were submitted. The NN-AA, MA-MO, and ENSEM were submitted as run1, run3, and run2 respectively. It is emphasized that we used the same systems to face the three sub-tasks. The proposed system exceeded our expectations, but the ensemble classifier boosted us in the ranking.

SubtaskB SubtaskA SubtaskC F1 macro F1 weig. F1 macro F1 weig. F1 macro F1 weig. NN-AA.run1 0.65357 0.70063 0.25579 0.588950.41692 0.66778MA-MO.run3 0.74706 0.80711 **0.50638** 0.75140 **0.49399** 0.77353 ENSEM.run2 0.75676 0.81820 0.50511 0.75957 0.48799 0.78403 Ranking 2nd place./45 3rd place./79 2nd place./50

Table 3. Offical ranking for English SubtaskA, SubtaskB and SubtaskC.

4 Conclusions

In this paper, we proposed two approaches that allow us to face this shared task. We observe that, for the first approach, the combination of CNNs and RNNs for handling n-grams and long-term dependencies cannot generate satisfactory results. Such an issue gets solved when we incorporate attention layers and a vector POS representation. We believe this improvement is the result of considerations imposed on important words and the most repeated POS patterns inside sentences. For the second approach, the proposed systems perform robust enough. We consider that fact is due to each of the systems manages to capture specific patterns that other systems ignore, whereas the ensemble could join all these patterns. It is important to mention that for sub-task B and C, data augmentation improved the performances.

For future works, we have observed that more in-depth studies on DL and taking into account the actual state-of-the art can help us improve our results.

Acknowledgments

I would like to share my gratitude with Gong, Zheng. His suggestions were important to write this paper.

References

1. Chang, C.C., C.J.: Libsvm: library Lin, А for support vec- tor machines. ACM Trans. Intell. Syst. Technol. 2(3),27:1https://doi.org/10.1145/1961189.1961199, 27:272011). (May http://doi.acm.org/10.1145/1961189.1961199

Victor Nina-Alcocer et al.

- 2. Forum for Information Retrieval Evaluation: FIRE 2019 (2019), http://fire.irsi.res.in/fire/2019/home
- 3. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for twitter: Annotation, features, and experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2. pp. 42–47. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011), http://dl.acm.org/citation.cfm?id=2002736.2002747
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, http://dx.doi.org/10.1162/neco.1997.9.8.1735
- Hu, D.: An introductory survey on attention mechanisms in nlp problems. In: Bi, Y., Bhatia, R., Kapoor, S. (eds.) Intelligent Systems and Applications. pp. 432–448. Springer International Publishing, Cham (2020)
- Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37. pp. 2342–2350. ICML'15, JMLR.org (2015), http://dl.acm.org/citation.cfm?id=3045118.3045367
- Khoo, C.S., Johnkhan, S.B.: Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. Journal of Information Science 44(4), 491–511 (2018). https://doi.org/10.1177/0165551517703514, https://doi.org/10.1177/0165551517703514
- Kim, Y.: Convolutional neural networks for sentence classification. CoRR abs/1408.5882 (2014), http://arxiv.org/abs/1408.5882
- Kůrková, V.: Kolmogorov's theorem and multilayer neural networks. Neural Netw. 5(3), 501–506 (Mar 1992). https://doi.org/10.1016/0893-6080(92)90012-8, http://dx.doi.org/10.1016/0893-6080(92)90012-8
- LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (5 2015). https://doi.org/10.1038/nature14539
- Mikolov, T., Kombrink, S., Deoras, A., Burget, L., Černocký, J.: Rnnlm - recurrent neural network language modeling toolkit. In: Proceedings of ASRU 2011. pp. 1–4. IEEE Signal Processing Society (2011), https://www.fit.vut.cz/research/publication/10087
- 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)
- 13. Nina-Alcocer, V.: AMI at ibereval2018 automatic misogyny identification in spanish and english tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SE-PLN 2018), Sevilla, Spain, September 18th, 2018. pp. 274–279 (2018), http://ceurws.org/Vol-2150/AMI_paper8.pdf
- Nina-Alcocer, V.: Haterecognizer at semeval-2019 task 5: Using features and neural networks to face hate recognition. In: Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019. pp. 409–415 (2019), https://www.aclweb.org/anthology/S19-2072/

Detecting Hate Speech and Offensive Content through Ensembles

- Nina-Alcocer, V., González, J., Hurtado, L., Pla, F.: Aggressiveness detection through deep learning approaches. In: Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019. pp. 544–549 (2019), http://ceur-ws.org/Vol-2421/MEX-A3T_paper_9.pdf
- Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012). ELRA, Istanbul, Turkey (May 2012)
- Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. pp. 1532–1543 (2014), https://nlp.stanford.edu/projects/glove/
- Rish, I.: An empirical study of the naive bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. vol. 3, pp. 41–46. IBM New York (2001)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
- Wright, R.E.: Logistic regression. In: Reading and understanding multivariate statistics., pp. 217–244. American Psychological Association, Washington, DC, US (1995)
- Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of CNN and RNN for natural language processing. CoRR abs/1702.01923 (2017), http://arxiv.org/abs/1702.01923