Can the Students surpass the Teacher ?

Gustave Besacier [1] **Guillaume Ferrer** [1] **Clément Renard** [1]

Abstract

In recent years, the proliferation of toxic speech on the Internet has triggered the need for new tools of systematic detection. Deep learning (DL) models are often the privileged choice to address the problem. However, they fails to provide good performance when strictly talking about hate speech and, as Large Language Model, their large memory usage create speed issues during inference, indeed all parameters are used when it might not be necessary. In this paper, we present an architecture aiming to improve state of the art hate speech detection model HateBERT by having faster inference and at least similar results. The architecture consists of a classifier sending sentences to one of eleven distilled hate speech detectors based on the hate speech target. While the model fails to obtain good performance overall due to the first classifier, good performance of students models indicate an interesting way to follow for further research.

Keywords: Hate speech, knowledge distillation, HateBERT, binary classification.

1. Introduction

Hate speech refers to offensive discourse targeting groups or individuals, based on their inherent characteristics such as race, religion or gender [1]. While it has always existed, the rise of social media and other forms of digital communication enabled more widespread and instantaneous expression of ideas, including hateful rhetoric.

Diverse DL models aim to foster a safer online environment by detecting hate speech in various type of modalities. These models have proven to effectively discriminate between harmful hate speech and legitimate critical discourse or satire. However it is at the cost of being extremely large in the number of parameters, impacting the speed of inference.[2].

In this context, we aim to build a model faster in inference, for detecting hate speech in English that discriminates sentences into two categories: `neutral` and `hate` speech, without loosing much performance compared to larger models.

The strategy consists of a first classifier head to identify the group targeted by a possible message of hate. The input is then softly assigned to one to three specialised hate speech classifiers, trained by distillation from HateBert a re-trained BERT model on data containing hateful speech (Caselli et al., 2021) [3]. They are fine-tuned to detect hate speech against a specific group. By training our smaller models on very specified targets, they also have the potential to learn more precise biased words embedding.

2. Related Work

Papakyriakopoulos et al. (2020) [4] proposed that biased word embedding can improve learning if the bias is towards what we want to discriminate. Indeed, further pre-training of transformer based pre-trained language models has shown to be an effective solution to adapt pre-trained language models to new language domains and further downstream tasks. An example is HateBERT for abusive language detection. Additionally Malik et al. (2023) [5] showed that distilled model such as small-BERT can have better computational efficiency while not sacrificing too much performance.

HateBERT learns new words embedding but without discriminating between targets of hate during training. We take it further by trying to create more precise biased embedding by training our distilled models not only on hate speech, but on hate speech directed towards a specific target. Because the relationship between words can vary considerably within each minority, we believe in the interest of learning particular relationships for different targets, while keeping basic relationship in sentences (BERT) and probably a general structure of hate speech (HateBERT).

3. Data

Our dataset is a combination of different existing datasets. Datasets have annotated distinction between hateful speech `hate` and other tones `neutral`, as well as the targets of the hateful rhetoric. We linked each couple sentence-target from the original datasets to one of 10 arbitrary chosen categories listed in Table 1. We added an `others` category for the sentences targeting groups we have not been taken into account. Single sentence can have multiples targets, hence produce multiple unique couple sentence-targets.

[1] Group 44.

EE-559: *deepLearning*, 2024.

3.1. Data acquisition

We used sentences generated by ToxiGen, a model for hate speech generation [6]. ToxiGen is based on small samples of hateful and neutral sentences targeting 1 out of 16 possible targets. Samples are used as inputs to the model to create one thousand prompts per category and constitute the dataset.

The second dataset comes from the paper HateXplain (2020) [7]. Sources are Tweets and Gabs. Samples were manually annotated the following way: whether the text is hate speech, offensive speech, or normal, and which group is targeted in this text. We chose to keep sentences annotated as hate speech following the majority of annotator. Sentences labeled as "offensive" are in our `normal` set. A sentence can have several targets; for sentences with a `normal` label, two annotators have to agree and for sentences with a `hate` label, only one is needed. The dataset was already tokenized so sentences were reconstructed by adding whitespace between each words.

The last dataset is from The Hate Speech Measurement Project from Sachdeva et al. [8]. The dataset is composed from user comments from YouTube, Reddit, and Twitter. The labels consist in a continuous hate score, where scores larger than 0.5 are considered as hate speech. There are also 42 different targets that we mapped to our 11 groups.

3.2. Data processing

The data is organized by file of the type `tone_target.csv`, for tones being either `hate` or `neutral`; for each of the groups presented in Table 1. To uniform it for our analyses, we regrouped the categories from each datasets to the categories of Table 1 as follows:

1. ToxiGen: [women, jewish, [asian, chinese], black, lgbtq, latin, muslim, native american, middle east, [physical disability, mental disability], [immigrant, mexican]]
2. HateXplain: [Women, Jewish, Asian, African, [], Hispanic, Islam, Indigenous, [], [], [None, Caucasian, Men, Christian, Heterosexual, Hindu, Buddhism]]
3. *Referring to index in column of* Measuring Hate Speech: [51, 35 ,22 ,23 ,[47, 48, 49, 50, 54, 55, 56], 24, 37, [26, 27], 25, [67, 68, 69, 70, 71, 72, 73, 74], [28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 52, 53, 57, 58, 59, 60, 61, 62, 63, 65, 66]]

	hate	neutral		hate	neutral
women	4250	11038	muslim	2109	4171
jews	1871	2372	indigeneous	305	1227
asian	649	2339	arab	1207	2994
black	3890	6579	disabilities	649	735
lgbtq	2478	5907	others	10370	40022
latino	966	2478	total	15731	42781

Table 1. Categories (order is top to bottom) with the number of sentences per tones. The data does not add up to the total as some sentences can have multiples targets.

4. Method

We built 11 `student models`. These models are specialized on a specific target. We use the model HateBERT as `teacher model`. Before using the model, we fine-tune HateBERT on all our data (see learning section 5).

The `student models` are trained using knowledge distillation [9][10] using the Kullback–Leibler (KL) divergence:

$$Div_{KL}(\mathbb{P}||\mathbb{Q}) = \sum_x \mathbb{P}(x)\log(\frac{\mathbb{P}(x)}{\mathbb{Q}(x)}) \tag{1}$$

It measures the divergence between the true probability distribution $\mathbb{P}$ from the predicted distribution $\mathbb{Q}$. We compute the distribution $\mathbb{P}_{\texttt{model}} = softmax(\frac{logits(y)}{T})$, where hyperparameter T (temperature) controls the smoothness of the soft targets. The distillation loss is given by:

$$\begin{aligned} loss_{\texttt{distil}} =& (1-\alpha)\cdot loss_{\texttt{student}} \\ &+ \alpha T^2 \cdot Div_{KL}(\mathbb{P}_{\texttt{teacher}}||\mathbb{Q}_{\texttt{student}}) \end{aligned} \tag{2}$$

where α balances the contributions of the soft targets from the `teacher model` and the hard targets (true labels) during the training. This loss is used to train the `student model`. The process is shown on fig.(1).

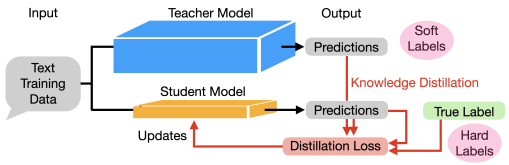

Figure 1. Training a model with knowledge distillation, it uses the distillation loss described by eq.(2).

Once all `student models` are trained, we trained a classifier that assigns the input text to the list of minorities in Table 1. In order to reduce the size and computational requirements, we start with a pre-trained `distilbert-base-uncased` student model derived from the BERT base model.

A seen on Fig. 2, the classifier output logits are fed into a softmax to select between the `student models` used to assess whether the input sentence is hateful or neutral. Students are selected up to the three most probable targets, if the cumulative probability does not go over 50%. Per example if the three largest probabilities are 0.3, 0.2, 0.1, only the first two students are selected. Normalized kept probabilities also give the weights for the final classification.

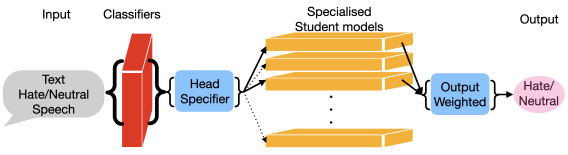

Figure 2. The architecture of our new model is shown above, as explained the classifiers assigns specialized heads for inference and the outputs are weighted according to the sentence target(s). Full model is called `octopus`

5. Details of learning

The sentiment analysis head is trained on the full training set. Inputs tokenized with `DistilBertTokenizerFast`, and pre-trained on `distilbert-base-uncased`. We trained the model using Cross entropy loss function taking into account the frequency of each class in the dataset, as it is unbalanced. We use AdamW optimizer with a learning rate scheduler starting at $5 \cdot 10^{-5}$ for 4 epochs.

Before training the `student models`, we fine-tuned HateBERT on our full training set, using AdamW optimizer with a linear learning rate scheduler starting at $5 \cdot 10^{-5}$. It can expect tokens of maximum length of 512, and is composed of 4 attention heads with transformers of 256 and 4 hidden layers.

`Student models` are also trained using AdamW optimizer with a custom cosine learning rate with hard restart and warm-up. It is similar to the one proposed by Loshchilov and Hutter [11] as it seems to perform better according to Gotmare and al. [12]; its motion is shown below:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{T_{\text{cur}}}{T_i}\pi\right)\right) \quad (3)$$

The learning rate starts at $5 \cdot 10^{-5}$ after a warmup of 50 steps, the initial length of the first cycle is 1 epoch and it doubles at each restart. The model trains for 7 epochs with $\alpha = 0.55$ and $T = 0.75$ in the distillation loss.

Every model are trained with balanced data with size depending on the minimum cardinality between the `neutral` and `hate` set. The ratio was a 4:1 train and test set using a random state of 38 and batch size of 16.

6. Results

Training of the classifier head obtained fairly low performance. Accuracy reaches 0.486 and F1 is 0.145. Student models reached better performances, as shown in Tab. 2

model	parameters	size[MB]	inference[s]	accuracy	f1
`student`	15'373'570	58.6	1.374	0.863	0.868
`teacher`	109'483'778	417.7	11.117	0.787	0.786
`classifier`	66'961'931	255.4	5.463	0.486	0.145
`octopus`	236'071'231	900	[6.84, 9.58]	0.692	0.655

Table 2. Characteristics of the model and performance. Results under `student model` are the ones trained on the `muslim` dataset as it was the one where we searched for the best hyperparameters.

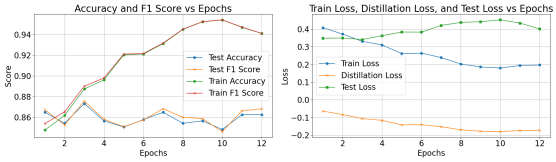

Figure 3. `student` overfits after 3 epochs as it can be seen both on the test loss (*right*) with starts increasing and the test f1 score and accuracy which peeks before decreasing *(left)*.

It was hard to train multiple specific `students` with similar rules as each needed specific hyperparameters for optimal performances. Depending on the size of the dataset, models may overfit at different stages, as illustrated in fig.3. The `Octopus` framework demonstrates faster performance compared to pure HateBert (9.58s at max vs 11.11s) when considering the classifier and three students. However, lack of specificity in the classifier resulted in a decrease in performance, as the individual models are not used efficiently.

We postulate that the alpha factor should be gradually reduced over time. As the `student` starts to learn the basics in hate classification, it should then focus on specialization and try to surpass the `teacher` for specific targets, all while avoiding overfitting. We experimented this hypothesis with a learning rate of $2 \cdot 10^{-5}$ on fig.4, the test performance are smoothed over time and seems gradually increase. We also search for the best hyperparamaters by grid search both in α and T. It was hard to make any conclusion as models would overfit more of less quickly over the epochs. Nevertheless, in fig4, colder temperature seemed to be favorable.

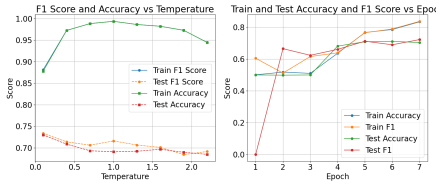

Figure 4. `muslim model` while varying hyperparameters α *(left)* and T *(right)*. The α paramaters was decreased over epochs to allow more freedom in the `student` learning process. We investigated the influence of temperature on learning over 7 epochs, lr=$5 \cdot 10^{-5}$, $\alpha = 0.5$.

7. Conclusion

Final predictions were deceiving because of the poor performances of the first head. Further idea could be to implement multi-label classification to deal with the sentences targeting multiple groups, as we have in our dataset. However distilled models showed promising performance, this motivates to continue in a direction with lighter distilled and fine-tuned model. It also proved the effectiveness of distillation for saving time in inferences.

8. Ethics

The objective of this work is to provide a model of hate speech detection so anyone can use the Internet without the threat of ill-intended users. Used dataset contains toxic sentences that could potentially harm people. The use of this data is only for research purposes and should not be used with a other intentions.

Model of this type and their results should be carefully trained and interpreted to protect freedom of speech and not restrict it. We value the importance of not to censoring ideas based on their semantic, as long as they are respectful.

References

[1] U. Nations, "Understanding hate speech, what is hate speech?." Available at https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech (18/05/2024).

[2] M. S. Jahan and M. Oussalah, "A systematic review of hate speech automatic detection using natural language processing," *Neurocomputing*, vol. 546, p. 126232, 2023.

[3] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, "HateBERT: Retraining BERT for abusive language detection in English," in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, (Online), pp. 17–25, Association for Computational Linguistics, Aug. 2021.

[4] O. Papakyriakopoulos, S. Hegelich, J. C. M. Serrano, and F. Marco, "Bias in word embeddings," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, (New York, NY, USA), p. 446–457, Association for Computing Machinery, 2020.

[5] J. S. Malik, H. Qiao, G. Pang, and A. van den Hengel, "Deep learning for hate speech detection: A comparative study," 2023.

[6] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar, "Toxigen: A large-scale machine-generated dataset for implicit and adversarial hate speech detection," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[7] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee, "Hatexplain: A benchmark dataset for explainable hate speech detection," *CoRR*, vol. abs/2012.10289, 2020.

[8] P. Sachdeva, R. Barreto, G. Bacon, A. Sahn, C. von Vacano, and C. Kennedy, "The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism," in *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022* (G. Abercrombie, V. Basile, S. Tonelli, V. Rieser, and A. Uma, eds.), (Marseille, France), pp. 83–94, European Language Resources Association, June 2022.

[9] S. M. Kaleem, T. Rouf, G. Habib, T. jan Saleem, and B. Lall, "A comprehensive review of knowledge distillation in computer vision," 2024.

[10] S. Teki, "Knowledge distillation: Principles, algorithms, applications," tech. rep., ML Model Development, 2023.

[11] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with restarts," *CoRR*, vol. abs/1608.03983, 2016.

[12] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," *CoRR*, vol. abs/1810.13243, 2018.