Teacher and Student Models of Offensive Language in Social Media

Tharindu Ranasinghe*
Aston University
Birmingham, UK
t.ranasinghe@aston.ac.uk

Marcos Zampieri*
George Mason University
Fairfax, VA, USA
mzampier@gmu.edu

Abstract

State-of-the-art approaches to identifying offensive language online make use of large pre-trained transformer models. However, the inference time, disk, and memory requirements of these transformer models present challenges for their wide usage in the real world. Even the distilled transformer models remain prohibitively large for many usage scenarios. To cope with these challenges, in this paper, we propose transferring knowledge from transformer models to much smaller neural models to make predictions at the token- and at the post-level. We show that this approach leads to lightweight offensive language identification models that perform on par with large transformers but with 100 times fewer parameters and much less memory usage.

1 Introduction

The presence of offensive content in social media has been linked to suicide attempts and ideation in teenagers, psychological stress, and other undesirable consequences to users (Bannink et al., 2014; Saha et al., 2019). This motivated researchers to develop models to identify offensive content automatically. There have been several recent studies published on identifying different types of offensive content such as profanity (Holgate et al., 2018; Sarkar et al., 2021), pejorative language (Dinu et al., 2021), abuse (Koufakou et al., 2020), hate speech (Davidson et al., 2017; Mathew et al., 2021), cyber-bullying (Paul and Saha, 2020), and toxicity (Ranasinghe and Zampieri, 2021). While most studies on this topic deal with the identification of offensive content at the post level, recent work (Mathew et al., 2021) has addressed the identification of offensive tokens in posts along with their targets (Zampieri et al., 2023), which can assist human moderators (e.g., news portals moderators) who often review lengthy comments (Pavlopoulos et al., 2021; Weerasooriya et al., 2023). Overall, both token-level and sentence-level tasks are important to real-world applications.

Recent international competitions on the topic, such as OffensEval (Zampieri et al., 2019b), HatEval (Basile et al., 2019) and HASOC (Modha et al., 2021; Satapara et al., 2023), have shown that large pre-trained transformer models such as BERT (Devlin et al., 2019), and XLNet (Yang et al., 2019) deliver state-of-the-art performance in offensive language identification at both sentence and token levels. However, the models based on transformers have an extremely large number of parameters and are well-known for demanding computing resources such as disk and RAM. This makes inference time slow, posing challenges for real-time inference and their deployment in the real world.

Making models smaller and more usable in practice is an active area in machine learning and NLP (Tay et al., 2022). One approach is Knowledge Distillation (KD) which aims to extract knowledge from a top-performing large model (the teacher) into a smaller, yet well-performing model (the student) (Gou et al., 2021). The student model is ideally a model which is less demanding in terms of memory print, computing power and with lower prediction latency. DistilBERT is such an example where KD has been used to create smaller language models (Sanh et al., 2019). The search for computational efficiency is in line with several initiatives such as Green AI (Schwartz et al., 2020) and the ACL's Efficient NLP policy.[1]

In this paper, we perform a detailed evaluation of performing KD in offensive language identification. First, we evaluate KD in sentence-level offensive language detection. Previous attempts for KD in text classification propose to use computationally augmented synthetic data (Tang et al., 2019) to address data scarcity (Rizos et al., 2019).

*The two authors contributed equally to this work.

[1] https://www.aclweb.org/adminwiki/images/7/7e/ACL_Efficient_NLP_Policy.pdf

Findings of the Association for Computational Linguistics: ACL 2023, pages 3910–3922

We address this limitation by using a very large offensive language identification dataset in the KD process. Secondly, we evaluate KD in token-level offensive language identification. We introduce a novel way to perform KD at the token level by using the Viterbi algorithm (Viterbi, 1967).

In both sentence-level and token-level, we show that smaller student models deliver results on par with KD despite requiring fewer computing resources, and interestingly smaller student models outperform machine learning models based on distilled transformers such as DistilBERT. To the best of our knowledge, this is the first comprehensive KD research applied to offensive language online opening exciting new avenues for research in this area. The findings of this research will be beneficial not only to offensive language research but also to a wide range of related social media tasks, such as sentiment analysis and fake news detection.

The contributions of this paper are as follows:

1. An empirical evaluation of KD in sentence-level offensive language identification using multiple teachers and large real-world datasets.

2. A novel method to perform KD in word-level offensive language identification using the Viterbi algorithm.

3. The release of a novel version of SOLID (Rosenthal et al., 2021) with both sentence-level labels and token-level teacher scores.

4. The release of the models made freely available to the research community, which are high-performing and competitive with large transformer models, yet lightweight machine learning models for multilingual offensive language identification[2].

2 Related Work

Post-level Offensive Language Identification There is growing interest in the development of computational models to identify offensive content online. Early approaches relied on feature engineering combined with traditional machine learning classifiers, most notably, Naive Bayes and SVMs (Xu et al., 2012; Dadvar et al., 2013; Malmasi and Zampieri, 2018). More recently, neural networks have proved to outperform traditional machine learning methods on most available benchmark datasets (Aroyehun and Gelbukh, 2018; Modha et al., 2018). The impact of the neural networks in offensive language identification has been significant with the introduction of BERT (Devlin et al., 2019). Transformer models such as BERT (Devlin et al., 2019), and XLNet (Yang et al., 2019) have been applied to offensive language identification at the sentence level topping the leaderboards in recent shared tasks (Liu et al., 2019; Ranasinghe et al., 2019). The success of BERT models in this task has motivated the development of high-performing task-specific models such as fBERT (Sarkar et al., 2021) and HateBERT (Caselli et al., 2021).

Token-level Offensive Language Identification Similar to the sentence-level (Sarkar et al., 2021) transformer models have provided state-of-the-art results at the token-level as well. This is further confirmed by several transformer-based open-source frameworks, such as MUDES (Ranasinghe and Zampieri, 2021) that have been released to perform token-level offensive language identification. However, even though transformers provide state-of-the-art results, in terms of inference efficiency, their running time is still considerably higher than other neural network architectures. While token-level offensive language models can assist human moderators and improve the explainability of the sentence-level models, there are not many datasets annotated at the token-level. As far as we know, there are only a few English datasets with token-level, namely HateXplain (Mathew et al., 2021), TSD (Pavlopoulos et al., 2021), and TBO (Zampieri et al., 2023) - the first two are used in this research. The lack of post-level annotated data for this task is mainly due to the fact that token-level labels are more expensive to annotate. Therefore, we believe that our approach to leverage unlabeled data in token-level offensive language identification will be beneficial to the community.

Knowledge Distillation KD enables the transfer of knowledge from a large model to a smaller "student" network, which is improved in the process (Ba and Caruana, 2014). In NLP, KD has previously been used in neural machine translation (Yu et al.), language modelling (Kim and Rush, 2016), paraphrase detection (Tang et al., 2019) and translation quality estimation (Gajbhiye et al., 2021). These studies have explored the effect of having

[2] https://github.com/tharindudr/DistilOffense

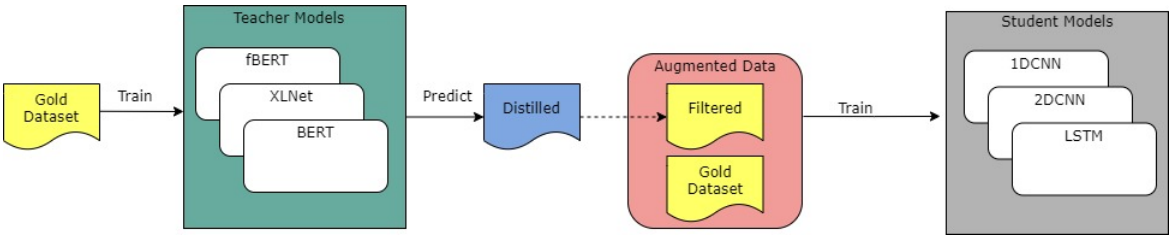


Figure 1: Knowledge Distillation strategy with data augmentation and filtering based on teacher uncertainty.

multiple teachers (Wu et al., 2021; Liu et al., 2020) and teacher uncertainty (Mukherjee and Awadallah, 2020). As discussed in the introduction, sentence-level offensive language identification is essentially a text classification problem. Previous research for KD in text classification has used synthetic data augmentation (Tang et al., 2019). However, producing high quality synthetic data for social media downstream tasks is a challenge (Rizos et al., 2019) as social media texts are non-standard, containing emojis, hashtags and specific words that do not appear in more standard general domain texts (e.g. news). In this research, we address this gap by proposing a KD process based on a non-synthetic data augmentation process using a SOLID (Rosenthal et al., 2021), large offensive language identification dataset.

3 Methods

Figure 1 summarizes our teacher and student approach. We follow two different approaches for sentence-level and token-level described in the next sub-sections.

3.1 Sentence-level Approach

The following components are included in the sentence-level KD process.

Gold Datasets We used two different English gold datasets to train the teacher models.

i **OLID** (Zampieri et al., 2019a) is the official dataset of the SemEval-2019 Task 6 (OffensEval) (Zampieri et al., 2019b). It contains data from Twitter annotated with a three-level hierarchical annotation in which level A classifies posts into offensive and not offensive. We used the level A in OLID as our offensive language identification task.

ii **HASOC** (Mandl et al., 2020) is the dataset used in the HASOC shared task 2020. It contains posts retrieved from Twitter and Facebook. The upper level of the annotation taxonomy used in HASOC is hate-offensive and non hate-offensive. We used these labels as our offensive language identification task for HASOC.

The process described next is repeated for both gold datasets.

Training Teachers For the teacher models, we use three pre-trained, fine-tuned transformer models: BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and fBERT (Sarkar et al., 2021). They have achieved state-of-the-art results on a variety of offensive language identification tasks. From an input sentence, transformers compute a feature vector $\boldsymbol{h} \in \mathbb{R}^d$, upon which we build a classifier for the task. For this task, we implemented a softmax layer, i.e., the predicted probabilities are $\boldsymbol{y}^{(B)} = \text{softmax}(W\boldsymbol{h})$, where $W \in \mathbb{R}^{k \times d}$ is the softmax weight matrix, and k is the number of labels. We employed a batch-size of 16, Adam optimiser with learning rate $2\text{e}{-5}$, and a linear learning rate warm-up over 10% of the training data. During the training process, the parameters of the transformer model, as well as the parameters of the subsequent layers, were updated. The models were trained using only training data. Furthermore, they were evaluated while training using an evaluation set that had one fifth of the rows in training data. We performed early stopping if the evaluation loss did not improve over three evaluation steps. All the models were trained for three epochs.

Teacher Knowledge Transfer In order to extract the knowledge from the teacher models, we used the tweets from another offensive language identification dataset: SOLID (Rosenthal et al., 2021). SOLID is a recently released large dataset created using OLID's general annotation model but using semi-supervised learning instead of manually annotated labels containing over 1.4 million offensive English tweets. Since HASOC follows a slightly different annotation taxonomy (hate-offensive vs non hate-offensive), instead of using the semi-

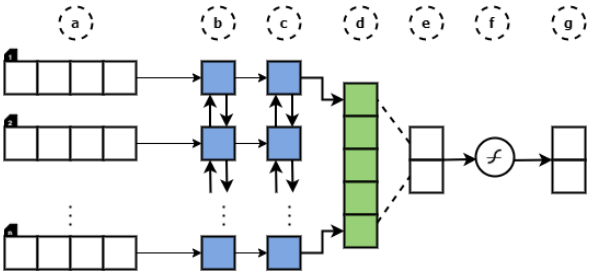
supervised labels provided in SOLID, we used our pre-trained teacher models from step 1 to predict the labels for the tweets in SOLID. Furthermore, this approach can provide a general KD solution to a wide range of tasks that follows different annotation taxonomies other than OLID. For each tweet in SOLID, the teacher models predict the labels and the confidence of the label, which is useful for the next step. We report the results on three teacher models. In order to compare the efficiency of using multiple teachers to a single-teacher model, we also considered a scenario where there is only one teacher model. We trained a BERT model on three different random seeds and performed the same steps mentioned above. We did not consider having more than three teachers as training teachers requires more computational power.

Teacher Model Configurations We used a NVIDIA RTX A6000 48GB GPU to train the teacher models. We divided the dataset into a training set and a validation set using 0.8:0.2 split. For the teacher models, we used the same set of configurations mentioned in Table 1 in all the experiments. We performed *early stopping* if the validation loss did not improve over 10 evaluation steps. All the experiments were conducted for three times and the mean value is taken as the final reported result.

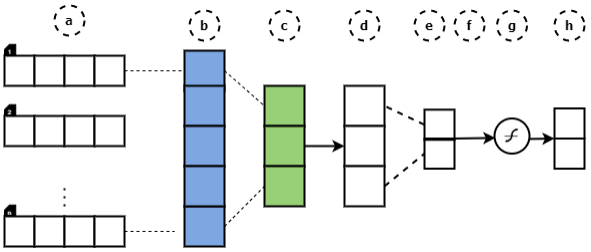
Parameter	Value
adam epsilon	1e-8
batch size	64
epochs	3
learning rate	1e-5
warmup ratio	0.1
warmup steps	0
max grad norm	1.0
max seq. length	256
gradient accumulation steps	1

Table 1: Teacher Parameter Specifications.

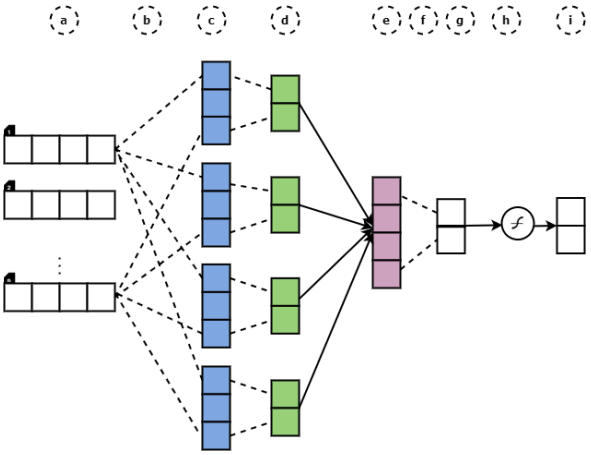
Filtering and Augmentation In this step, we augment the gold training set with the texts from SOLID and with the pseudo-labels provided by the teachers for tweets in SOLID, to aid in effective knowledge distillation. However, the benefits of data augmentation can be hampered by noise in teacher predictions. Therefore, we perform a filtering process to filter out noisy examples in SOLID based on teacher uncertainty. We calculated the standard deviation of the confidence of the teacher models for the positive class, which corresponds to the uncertainty of the teacher models. We used different threshold values for teacher uncertainty to understand the behaviour of the student models with teacher confidence. We repeated this process for one teacher model scenario, too, where we had three teacher models from BERT in different random seeds.



(i) The BiLSTM student model for offensive language identification. The labels are **(a)** input embeddings, **(b,c)** two BiLSTM layers, **(d, e)** fully-connected layers; **(f)** softmax activation, and **(g)** final probabilities.



(ii) The 1DCNN student model for offensive language identification. The labels are **(a)** input embeddings, **(b)** 1DCNN, **(c)** max pooling, **(d, e)** fully-connected layer; **(f)** with dropout, **(g)** softmax activation, and **(h)** final probabilities.



(iii) The 2DCNN student model for offensive language identification. The labels are **(a)** input embeddings, **(b)** spatial dropout, **(c, d)** four parallel 2DCNN layers with connected pooling layers, **(e)** concatenation layer, **(f)** dropout, **(g)** fully-connected layer; **(h)** softmax activation, and **(i)** final probabilities.

Figure 2: Graphic representations of the student models based on BiLSTM, 1DCNN, and 2DCNN.

Model	Parameter	Value
BiLSTM	First dense layer units	256
	LSTM units	64
	vocab size	3,000
1DCNN	Conv1D filters	128
	Conv1D kernel size	5
	Dropout	0.2
	First dense layer units	256
	MaxPooling1D pool size	5
	vocab size	3,000
2DCNN	Conv2D filters	32
	Conv2D kernel sizes	[(1, 2, 3, 5), 300]
	Dropout	0.2
	Spatial Dropout	0.4
	MaxPool2D kernel size	[(256, 257, 258, 260), 1]
	Vocab size	3,000

Table 2: BiLSTM, 1DCNN, and 2DCNN student parameter specifications.

Training Students We trained three simple and lightweight student networks; **BiLSTM** (Figure 2i), **1DCNN** Figure(2ii) and **2DCNN** (Figure 2iii) on the augmented dataset. We employed a batch-size of 16, Adam optimiser with learning rate $1e{-}4$, and we used Google word2vec embeddings. The student models were trained for ten epochs and performed early stopping if the loss did not improve on a validation set that had one-fifth of the rows in training data.

Student Model Configurations We used a NVIDIA RTX A6000 48GB GPU to train the student models. We divided the dataset into a training set and a validation set using 0.8:0.2 split. We performed *early stopping* if the validation loss did not improve over 10 evaluation steps. For all the experiments we used a batch size of 64, max seq. length of 256, learning rate of 1e-4 and 20 epochs in the training process. Individual configuration for the student models; BiLSTM, 1DCNN and 2DCNN are available on Table 2. All the experiments were conducted for three times and the mean value is taken as the final reported result.

Baseline We also trained a lightweight transformer model; DistilBERT (Sanh et al., 2019) on the augmented dataset, which we used the baseline to compare the student models. The configuration of the DistilBERT is similar to the teacher models. Following Gajbhiye et al. (2021), we do not train the large teacher models on the augmented dataset as it can consume a lot of resources.

3.2 Token-level Approach

Gold Datasets We used two English gold datasets to train the teacher models in at the token-level.

i **TSD** was released within the scope of SemEval-2021 Task 5: Toxic Spans Detection for English (Pavlopoulos et al., 2021). The dataset contains 10,000 posts (comments) from the publicly available Civil Comments dataset (Borkan et al., 2019). If a post is toxic, it has been annotated for its toxic spans.

ii **HateX** (Mathew et al., 2021) was also used as a gold dataset at the token level. The dataset contains 11535 training and 3844 testing instances from GAB and Twitter. We only used the word-level annotations, where if a instance is labelled as offensive or hatespeech, each token is labelled whether it contributes to the sentence-level label or not.

The process described next is repeated for both gold datasets.

Training Teachers For the teacher models in token level, we use three pre-trained, fine-tuned transformer models: BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and RoBERTa (Sarkar et al., 2021). As mentioned before, these models have achieved state-of-the-art performance in token-level offensive language detection. We used the default token classification architecture of the transformer models. All the configurations were similar to the sentence-level teacher models.

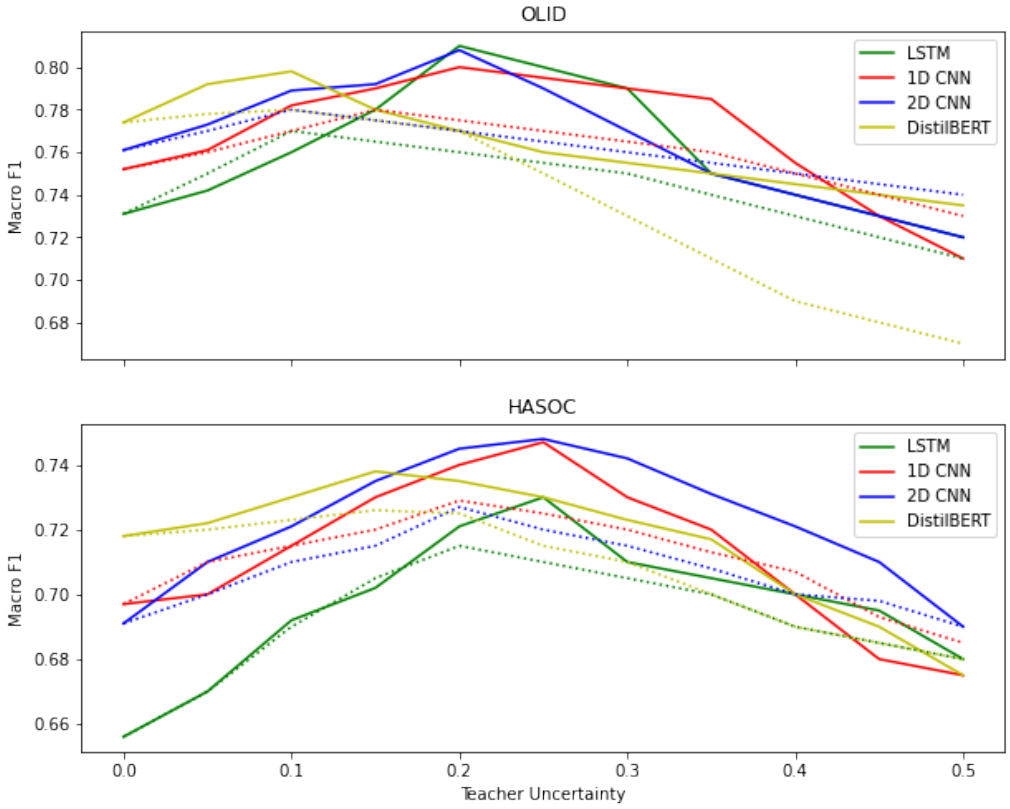


Figure 3: The Macro F1 scores of the student models with different teacher uncertainty levels on both datasets

Teacher Knowledge Transfer In order to extract the knowledge from the teacher models, we used the tweets SOLID (Rosenthal et al., 2021). While SOLID contains sentence-level semi-supervised labels on OLID level A, it does not have token-level labels. Therefore, we used our pre-trained teacher models from the previous step to predict the labels for the tweets in SOLID. For each tweet's token in SOLID, the teacher models predict the labels and the confidence of the label. As we have three teachers, we used the mean confidence value to represent the confidence of a single token.

Augmentation The data augmentation step in the token-level is different from the sentence-level because each unique combination of the output sequence is treated as a different category, then the standard distillation objective is no longer appropriate as the number of unique combinations for a length L sequence with offensive and non offensive labels scale at 2^L. Therefore, we followed a different approach with k-best Viterbi decoding, which can find the top-K sequences that are most probable. Our motivation is that the k-best Viterbi can be repurposed to pick out the K-most probable label sequences predicted by the teacher model. Our approach extracts information from the teacher models by drawing a set of most probable sequences, together with the respective confidence to those sequences. Then these sequences are augmented to the training sets of the students. We selected top-1 and (top-1+top-2) sequences to augment the dataset. We repeated this process for one teacher model scenario where we had three teacher models from BERT in different random seeds.

Training Students We trained two simple and lightweight student networks at token level; **BiLSTM** model with (i) an input embedding layer, (ii) a bidirectional LSTM layer with 64 units, followed by (iii) a linear chain conditional random field (CRF) (Lafferty et al., 2001), **1DCNN** model with (i) an input embedding layer, (ii) a bidirectional CNN with a kernel size of five, followed by (iii) a fully connected layer. We employed a batch-size of 16, Adam optimiser with learning rate $1e{-}4$, and we used Google word2vec embeddings. The student models were trained for 25 epochs and performed early stopping if the loss did not improve a validation set that had one-fifth of the rows in training data.

Baseline Similar to the sentence-level baseline, we trained DistilBERT (Sanh et al., 2019) as a token classification task on the augmented dataset.

4 Results and Discussion

4.1 Sentence-level Offensive Language Detection

With the sentence-level experiments, we answer the following research questions:

- **RQ1** How do different student models behave with different confidence levels of the teacher models?
- **RQ2** Can student models perform competitively with teacher models at sentence-level?

Figure 3 shows the Macro F1 scores of different teacher confidence values that were used to filter the SOLID before augmenting with the gold training set.

As can be seen in the figure, F1 score of the student improves with KD process. However, after a certain teacher uncertainty value, student model performance seems to plateau and does not further improve despite having more training examples. This is expected as teacher uncertainty causes noisy training instances. This is true for both datasets. 2DCNN model provided the best results of the three student models we considered. Dotted lines in Figure 3 show the results of the KD with one teacher for each student model. It is clear that our KD approach with three teachers provided better results than one teacher. This is also true for both datasets we experimented with. With this analysis, we can answer our **RQ1**, student models improve only for a certain teacher uncertainty level, and performing further KD with less confident teacher predictions would not improve the student models. Furthermore, having three teachers provides better results than having one teacher in the KD process. This corroborates the findings of previous research involving multiple teachers (Gajbhiye et al., 2021; Sun et al., 2019).

Table 3 shows the Macro F1 scores for the teacher models, student models and student models after performing KD in both gold datasets. Additionally, we also report the results for DistilBERT with KD and without KD. The F1 scores for the student models do not reach the performance of Teacher models such as BERT. Smaller models may lack representation power for modelling tasks such as offensive language identification. However, *KD allows student models to outperform DistilBERT with much less parameters and disk/ RAM space requirements* (see Table 5). This is true for both gold datasets. Interestingly, simple student models with KD outperform DistilBERT with KD. We believe that DistilBERT architecture is very similar to the teacher models, and it will not gain more knowledge from the teacher models. With these findings, we answer **RQ2**, student models perform competitively with the teacher models after KD. Furthermore, student models with KD outperform lightweight transformer models such as DistilBERT.

Type	Model	OLID	HASOC
Teachers	BERT	**0.8174**	0.7585
	XLNet	0.8125	**0.7592**
	fBERT	0.8101	0.7511
Students	BiLSTM (3T)	0.7998	0.7302
	1DCNN (3T)	0.8045	0.7472
	2DCNN (3T)	**0.8082**	**0.7487**
	BiLSTM (1T)	0.7712	0.7154
	1DCNN (1T)	0.7841	0.7298
	2DCNN (1T)	0.7801	0.7276
	BiLSTM	0.7342	0.6562
	1DCNN	0.7556	0.6971
	2DCNN	0.7678	0.6914
Baseline	DistilBERT (3T)	0.7981	0.7389
	DistilBERT (1T)	0.7802	0.7267
	DistilBERT	0.7781	0.7189

Table 3: Macro F1 scores for models on OLID and HASOC test sets. 3T = KD with three teachers 1T = KD with one teacher. The best result for each dataset for teachers and students are highlighted in bold.

4.2 Token-level Offensive Language Detection

With the token-level experiments, we answer the following research questions:

- **RQ3** Can student models perform competitively with teacher models at token-level?
- **RQ4** How do student models behave with different values for k in k-best Viterbi?

We present the token-level KD results in Table 4 after performing KD in both gold datasets for the teacher models, student models and student models after performing KD. For comparison purposes, we also report the results for DistilBERT with KD and without KD. It is clear that student models

Type	Model	k	HateX	TSD
Teachers	BERT		**0.6875**	**0.6538**
	XLNet		0.6779	0.6432
	RoBERTa		0.6754	0.6351
Students	BiLSTM (3T)	1	**0.6652**	**0.6352**
	1DCNN (3T)	1	0.6551	0.6289
	BiLSTM (3T)	2	0.4567	0.4265
	1DCNN (3T)	2	0.4683	0.4489
	BiLSTM (1T)	1	0.6532	0.6232
	1DCNN (1T)	1	0.6411	0.6101
	BiLSTM (1T)	2	0.4367	0.4487
	1DCNN (1T)	2	0.4488	0.4551
	BiLSTM		0.5981	0.5398
	1DCNN		0.5881	0.5341
Baseline	DistilBERT (3T)	1	0.6562	0.6325
	DistilBERT (1T)	1	0.6557	0.6228
	DistilBERT (3T)	2	0.4533	0.4216
	DistilBERT (1T)	2	0.4672	0.4331
	DistilBERT		0.6441	0.6132

Table 4: Macro F1 scores for models on HateX and TSD test sets. 3T = KD with three teachers 1T = KD with one teacher. The best result for each dataset for teachers and students are highlighted in bold.

improve a lot after KD. For the TSD dataset BiLSTM model was improved by 10% Macro F1 score. Furthermore, most of the student models (apart from k=2 models) have outperformed the DistilBERT after KD. With these findings, we answer **RQ3**, student models perform competitively with the teacher models after KD, and they outperform efficient transformer models such as DistilBERT. However, unlike sentence-level models, there is no clear evidence that multiple teachers can provide better results for the students. We believe that this is mainly because we did not take teacher uncertainty into account in the Viterbi algorithm.

Also, from the results, it is clear that adding the two best instances from the Viterbi algorithm reduced the results of the student models drastically. This can be due to the fact that Viterbi algorithm adds more noisy instances by having k=2. While augmenting with two best instances can produce more instances overall, it is clear that it will not improve the results of the student models. With these findings, we answer **RQ4**, the results of the student models reduce when we consider higher values for k in the Viterbi algorithm.

Name	#params	Inference		
		Speed (secs.)	RAM (MiB)	Disk (M)
BERT	561M	1.09	9,263.5	2140
fBERT	135M	0.82	1,979.2	517
XLNet	561M	1.15	9,167.7	2140
BiLSTM	6.2M	0.39	155.6	132
1DCNN	6.0M	0.35	151.8	125
2DCNN	5.9M	0.36	142.5	112
DistilBERT	66M	0.65	802.5	286

Table 5: Parameters, speed, and RAM, and disk usage for one sentence prediction on a CPU (Intel Xeon Silver 4114 CPU @ 2.20GHz) in sentence-level offensive language identification.

5 Computational Efficiency

Table 5 and Table 6 show the number of parameters, memory, disk space requirements, and inference speed for the teacher models, and student models compared to the DistilBERT baseline in both sentence-level and token-level.

Name	#params	Inference		
		Speed (secs.)	RAM (MiB)	Disk (M)
BERT	561M	1.09	9,263.5	2140
RoBERTa	432M	0.93	4,979.2	1420
XLNet	561M	1.12	9,487.6	2080
BiLSTM	4.8M	0.36	145.6	112
1DCNN	4.5M	0.33	121.8	105
DistilBERT	66M	0.68	702.5	286

Table 6: Parameters, speed, and RAM, and disk usage for one sentence prediction on a CPU (Intel Xeon Silver 4114 CPU @ 2.20GHz) in token-level offensive language identification.

We observe that the student models provide efficient and greener models (Schwartz et al., 2020) compared to the popular transformers models in terms of parameters, processing speed, and inference time. The KD strategy presented in this paper allows the student models to perform competitively, as presented in Section 4.1.

6 Conclusion and Future Work

In this paper, we showed that KD through a teacher-student approach that directly distills offensive labels can be effective in building lightweight offensive language identification models in both sentence and token-level. KD allows the lightweight student models to outperform distilled yet large pre-trained state-of-the-art architecture such as DistilBERT. Furthermore, they perform competitively with large state-of-the-art architectures such as BERT and XLNet. These students models are 15 times smaller in disk space with 100 times fewer parameters, and 3 times faster in inference speed demonstrating that KD can contribute to greener computing. The search for computational efficiency is in line with several initiatives such as Green AI (Schwartz et al., 2020) and the aforementioned ACL's Efficient NLP policy. As the KD process we used does not involve task specific configurations, the findings of this research can be easily expanded to different tasks involving social media texts such as sentiment analysis and fake news identification.

For the KD in token-level offensive language identification, we introduced a novel method involving the Viterbi algorithm to reduce the complexity of the task. We showed that the student models can be improved with taking the top 1 label sequence from the Viterbi algorithm. However, adding more instances with top 2 etc can also reduce student performance. Our approach will benefit multiple applications that have token-level labels. Furthermore, our approach can be used to generate more token-level data addressing the scarcity of available token-level datasets.

In the future, we would like to experiment with cross-lingual KD architectures. Transformer-based cross-lingual approaches have been previously successfully applied for this task (Ranasinghe and Zampieri, 2020; Zia et al., 2022). Therefore, we would like to explore large transformer models such as XLM-R that can be used as teachers to distil knowledge from resource rich languages to resource poor languages thus benefit multiple languages around the world for which offensive language datasets and resources are not yet available.

Limitations

We only experimented with one and three teacher models. Training more teacher models and using them to predict on large datasets such as SOLID (Rosenthal et al., 2021) would require more computing resources. Furthermore, we did not train the teacher models on the augmented dataset for the same reason following recent research in KD (Gajbhiye et al., 2021; Sun et al., 2019).

We only conducted the experiments in English. The non-availability of large-scale offensive language identification datasets such as SOLID (Rosenthal et al., 2021) in languages other than English can be a challenge when expanding this KD research beyond English.

Ethics Statement

For this research, we used multiple datasets referenced in this paper which were previously collected and annotated. No new data collection has been carried out as part of this work. We have not collected or processed writers'/users' information nor have we carried out any form of user profiling protecting users' privacy and anonymity.

Acknowledgments

We would like to thank the creators of the datasets used in this paper for making the data available for this research. We further thank the anonymous ACL reviewers who have provided us with constructive feedback to improve the quality of this paper.

The computational experiments in this paper were conducted on Aston EPS Machine Learning Server, funded by the EPSRC Core Equipment Fund, Grant EP/V036106/1.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of TRAC*.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Proceedings of NIPS*.

Rienke Bannink, Suzanne Broeren, Petra M van de Looij-Jansen, Frouwkje G de Waart, and Hein Raat. 2014. Cyber and Traditional Bullying Victimization as a Risk Factor for Mental Health Problems and Suicidal Ideation in Adolescents. *PloS one*, 9(4).

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of SemEval*.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Proceesings of WWW*.

Tommaso Caselli, Valerio Basile, Jelena Mitrovic, and Michael Granitzer. 2021. Hatebert: Retraining bert for abusive language detection in english. In *Proceedings of WOAH*.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving Dyberbullying Detection with User Context. In *Proceedings of ECIR*.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Liviu P Dinu, Ioan-Bogdan Iordache, Ana Sabina Uban, and Marcos Zampieri. 2021. A computational exploration of pejorative language in social media. In *Findings of the ACL (EMNLP)*.

Amit Gajbhiye, Marina Fomicheva, Fernando Alva-Manchego, Frédéric Blain, Abiola Obamuyide, Nikolaos Aletras, and Lucia Specia. 2021. Knowledge distillation for quality estimation. In *Findings of the ACL (ACL-IJCNLP)*.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6).

Eric Holgate, Isabel Cachola, Daniel Preoţiuc-Pietro, and Junyi Jessy Li. 2018. Why swear? analyzing and inferring the intentions of vulgar expressions. In *Proceedings of EMNLP*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of EMNLP*.

Anna Koufakou, Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2020. HurtBERT: Incorporating lexical features with BERT for the detection of abusive language. In *Proceedings of ALW*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of SemEval*.

Yuang Liu, Wei Zhang, and Jun Wang. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In *Proceedings of FIRE*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. In *Proceedings of AAAI*.

Sandip Modha, Prasenjit Majumder, and Thomas Mandl. 2018. Filtering Aggression from the Multilingual Social Media Feed. In *Proceedings of TRAC*.

Sandip Modha, Thomas Mandl, Gautam Kishore Shahi, Hiren Madhu, Shrey Satapara, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech. In *Proceedings of FIRE*.

Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. In *Proceedings of NeurIPS*.

Sayanta Paul and Sriparna Saha. 2020. Cyberbert: Bert for cyberbullying identification. *Multimedia Systems*, 28:1897–1904.

John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection. In *Proceedings of SemEval*.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual Offensive Language Identification with Cross-lingual Embeddings. In *Proceedings of EMNLP*.

Tharindu Ranasinghe and Marcos Zampieri. 2021. MUDES: Multilingual Detection of Offensive Spans. In *Proceedings of NAACL*.

Tharindu Ranasinghe, Marcos Zampieri, and Hansi Hettiarachchi. 2019. BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Offensive Language Identification. In *Proceedings of FIRE*.

Georgios Rizos, Konstantin Hemker, and Björn Schuller. 2019. Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of CIKM*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2021. SOLID: A Large-Scale Weakly Supervised Dataset for Offensive Language Identification. In *Findings of the ACL (ACL-IJCNLP)*.

Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of WebSci*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of EMC2*.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander Ororbia. 2021. fBERT: A neural transformer for identifying offensive content. In *Findings of the ACL (EMNLP)*.

Shrey Satapara, Prasenjit Majumder, Thomas Mandl, Sandip Modha, Hiren Madhu, Tharindu Ranasinghe, Marcos Zampieri, Kai North, and Damith Premasiri. 2023. Overview of the hasoc subtrack at fire 2022: Hate speech and offensive content identification in english and indo-aryan languages. In *Proceedings of FIRE*.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Communications of the ACM*, 63(12):54–63.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of EMNLP*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

Andrew J Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Tharindu Cyril Weerasooriya, Sujan Dutta, Tharindu Ranasinghe, Marcos Zampieri, Christopher M Homan, and Ashiqur R KhudaBukhsh. 2023. Vicarious offense and noise audit of offensive speech classifiers. *arXiv preprint arXiv:2301.12534*.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. One teacher is enough? pre-trained language model distillation from multiple teachers. In *Findings of the ACL (ACL-IJCNLP)*.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of NAACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of NeurIPS*.

Seunghak Yu, Nilesh Kulkarni, Haejun Lee, and Jihie Kim. On-device neural language model based word prediction. In *Proceedings of COLING*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of SemEval*.

Marcos Zampieri, Skye Morgan, Kai North, Tharindu Ranasinghe, Austin Simmons, Paridhi Khandelwal, Sara Rosenthal, and Preslav Nakov. 2023. Target-based offensive language identification. In *Proceedings of ACL*.

Haris Bin Zia, Ignacio Castro, Arkaitz Zubiaga, and Gareth Tyson. 2022. Improving zero-shot cross-lingual hate speech detection with pseudo-label fine-tuning of transformer language models. In *Proceedings of ICWSM*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
6 and 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1 and 6

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
3

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We used existing datasets that already anonymized

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3 and 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.