SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification

Sara Rosenthal[1], Pepa Atanasova[2], Georgi Karadzhov[3], Marcos Zampieri[4], Preslav Nakov[5]
[1]IBM Research, USA, [2]University of Copenhagen, Denmark,
[3]University of Cambridge, UK, [4]Rochester Institute of Technology, USA,
[5]Qatar Computing Research Institute, HBKU, Qatar
sjrosenthal@us.ibm.com

Abstract

The widespread use of offensive content in social media has led to an abundance of research in detecting language such as hate speech, cyberbullying, and cyber-aggression. Recent work presented the *OLID* dataset, which follows a taxonomy for offensive language identification that provides meaningful information for understanding the type and the target of offensive messages. However, it is limited in size and it might be biased towards offensive language as it was collected using keywords. In this work, we present *SOLID*, an expanded dataset, where the tweets were collected in a more principled manner. *SOLID* contains over nine million English tweets labeled in a semi-supervised fashion. We demonstrate that using *SOLID* along with *OLID* yields sizable performance gains on the *OLID* test set for two different models, especially for the lower levels of the taxonomy.

1 Introduction

Offensive language in social media has become a concern for governments, online communities, and social media platforms. Free speech is an important right, but moderation is needed in order to avoid unexpected serious repercussions. In fact, this is so serious that many countries have passed or are planning legislation that makes platforms responsible for their content, e.g., the *Online Harm Bill* (HM Government, 2019) in the UK and the *Digital Services Act* (European Commission, 2020) in the EU. Even in the United States, content moderation or the lack thereof can have significant impact on business (e.g., Parler was denied server space), government (U.S. Capitol Riots), and individuals (hate speech is linked to self-harm). Explainability is needed to indicate in detail why content has been deleted or flagged as inappropriate. Moreover, users can be educated by such feedback to avoid future biases.

There have been several areas of work in the detection of offensive language (Basile et al., 2019; Fortuna and Nunes, 2018; Ranasinghe and Zampieri, 2020), covering overlapping characteristics such as toxicity, hate speech, cyberbullying, and cyber-aggression. Further, using a hierarchical approach to analyze different aspects of the offensive content, such as the type and the target of the offense, helps provide explainability. The Offensive Language Identification Dataset, or *OLID*, (Zampieri et al., 2019a) is one such example, and it has been widely used in research. *OLID* contains 14,100 English tweets, which were manually annotated using a three-level taxonomy:

A: Offensive Language Detection
B: Categorization of Offensive Language
C: Offensive Language Target Identification

The taxonomy proposed in *OLID* makes it possible to represent different kinds of offensive content as a function of the *type* and the *target* of a post. For example, offensive messages targeting a group are likely hate speech, whereas offensive messages targeting an individual are likely cyberbullying. *OLID* has been used to annotate datasets in languages such as Arabic (Mubarak et al., 2021), and Greek (Pitenis et al., 2020), allowing for multilingual learning and analysis.

An inherent feature of the hierarchical annotation is that the lower levels of the taxonomy contain a subset of the instances in the higher levels, and thus there are fewer instances in the categories in each subsequent level. This makes it very difficult to train robust deep learning models on such datasets. Moreover, due to the natural infrequency of offensive language (e.g., less than 3% of the tweets are offensive when selected at random), obtaining offensive content is

WARNING: This paper contains tweet examples and words that are offensive in nature.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 915–928

a costly and time-consuming effort. In this paper, we address these limitations by proposing a new dataset: **S**emi-Supervised **O**ffensive **L**anguage **I**dentification **D**atatset (*SOLID*)[1]. Our contributions are as follows:

1. We are the first to apply a semi-supervised method for collecting new offensive data using *OLID* as a seed dataset, thus avoiding the need for time-consuming annotation.
2. We create and publicly release *SOLID*, a training dataset containing 9 million English tweets for offensive language identification, the largest dataset for this task. *SOLID* is the official dataset of the SemEval shared task OffensEval 2020 (Zampieri et al., 2020).
3. We demonstrate sizeable improvements over prior work on the mid and lower levels of the taxonomy, where gold training data is scarce when training on *SOLID* and testing on *OLID*.
4. We provide a new larger test set and a comprehensive analysis of *EASY* (i.e., simple explicit tweets such as using curse words) and *HARD* (i.e., more implicit tweets that use underhanded comments or racial slurs) examples of offensive tweets.

The remainder of this paper is organized as follows: Section 2 presents related studies in aggression identification, bullying detection, and other related tasks. Section 3 describes the *OLID* dataset and annotation taxonomy. Section 4 introduces the computational models used in this study. Section 5 presents the *SOLID* dataset. Section 6 discusses the experimental results and Section 6.3 offers additional discussion and analysis. Finally, Section 7 concludes and discusses possible directions for future work.

2 Related Work

There have been several recent studies on offensive language detection and related tasks such as hate speech, cyberbulling, aggression, and toxic comment detection.

Hate speech identification is by far the most studied abusive language detection task (Ousidhoum et al., 2019; Chung et al., 2019; Mathew et al., 2021). One of the most widely used datasets is the one by Davidson et al. (2017), which contains over 24,000 English tweets labeled as non-offensive, hate speech, and profanity. A recent shared task on the topic is HatEval (Basile et al., 2019).

In cyberbullying detection, Xu et al. (2012) used sentiment analysis and topic models to identify relevant topics. Dadvar et al. (2013) and Safi Samghabadi et al. (2020) studied the use of the conversational context for detecting cyberbullying. In particular, Dadvar et al. (2013) used user-related features such as the frequency of profanity in previous messages. More recent work has addressed the issues of scalable and timely detection of cyberbullying in online social networks. To this end, Rafiq et al. (2018) employed a dynamic priority scheduler, and Yao et al. (2019) proposed a sequential hypothesis testing. Safi Samghabadi et al. (2020) constructed a dataset of cyberbullying episodes from the semi-anonymous social network ask.fm.

There were two editions of the TRAC shared task on Aggression Identification (Kumar et al., 2018, 2020) which provided participants with datasets containing annotated Facebook posts and comments in English and Hindi for training and validation. Facebook and Twitter datasets were used for testing. The goal was to discriminate between three classes: non-aggressive, covertly aggressive, and overly aggressive. Two other shared tasks addressed toxic language. The Toxic Comment Classification Challenge[2] at Kaggle provided participants with comments from Wikipedia annotated using six labels: toxic, severe toxic, obscene, threat, insult, and identity hate. The recent SemEval-2021 Toxic Spans Detection shared task addressed the identification of the token spans that made a post toxic (Pavlopoulos et al., 2021).

There were several shared tasks that have focused specifically on offensive language identification. For example, GermEval 2018 (Wiegand et al., 2018) which focused on offensive language identification in German tweets, HASOC 2019 (Mandl et al., 2019), and TRAC 2018 (Fortuna et al., 2018).

In this paper, we extend the prior work of the *OLID* dataset (Zampieri et al., 2019a). *OLID* is annotated using a hierarchical annotation schema as in (Basile et al., 2019; Mandl et al., 2019). In contrast to prior approaches, it takes both the target and the type of offensive content into account. This allows multiple types of offensive content (e.g., hate speech and cyberbullying) to be repre-

[1] Available at: https://sites.google.com/site/offensevalsharedtask/solid

[2] http://kaggle.com/c/jigsaw-toxic-comment-classification-challenge

sented in *OLID*'s taxonomy. We create a large-scale semi-supervised dataset using the same annotation taxonomy as in *OLID*.

3 The OLID Dataset

The *OLID* (Zampieri et al., 2019a) dataset tackles the challenge of detecting offensive language using a labeling schema that classifies each example using the following three-level hierarchy:

Level A: *Offensive Language Detection* Is the text offensive?
OFF Inappropriate language, insults, or threats.
NOT Neither offensive, nor profane.

Level B: *Categorization of Offensive Language* Is the offensive text targeted?
TIN Targeted insult or threat towards a group or individual.
UNT Untargeted profanity or swearing.

Level C: *Offensive Language Target Identification* What is the target of the offense?
IND The target is an individual explicitly or implicitly mentioned in the conversation;
GRP Hate speech, targeting a group of people based on ethnicity, gender, sexual orientation, religion, or other common characteristic.
OTH Targets that do not fall into the previous categories (e.g., organizations, events, and issues.)

The taxonomy was successfully adopted for several languages (Mubarak et al., 2021; Pitenis et al., 2020; Sigurbergsson and Derczynski, 2020; Çöltekin, 2020), and it was used in a series of shared tasks (Zampieri et al., 2019b; Mandl et al., 2019). Tweets from the *OLID* dataset labeled with the taxonomy are shown in Table 1. The *OLID* dataset consists of 13,241 training and 860 test tweets.

Table 2 presents detailed statistics about the distribution of the labels. There is a substantial class imbalance on each level of annotation, especially at Level B. Furthermore, there is a sizable difference in the total number of annotations between the levels due to the schema (e.g., Level C is 1/3 smaller than Level A), and the data sizes for B and C are rather small. These drawbacks indicate the need to create a larger dataset.

4 Models

In this section, we describe the models used for semi-supervised annotation and for evaluating the contribution of *SOLID* for offensive language identification. We use a suite of heterogeneous machine learning models: PMI (Turney and Littman, 2003), FastText (Joulin et al., 2017), LSTM (Hochreiter and Schmidhuber, 1997), and BERT (Devlin et al., 2019). They have diverse inductive biases, which is an essential prerequisite for our semi-supervised setup (see Section 4.5). We assume that an ensemble of models with different inductive biases decreases each individual model's bias.

Tweet	A	B	C
@USER Anyone care what that dirtbag says?	OFF	TIN	IND
Poor sad liberals. No hope for them.	OFF	TIN	GRP
LMAO....YOU SUCK NFL	OFF	TIN	OTH
@USER What insanely ridiculous bullshit.	OFF	UNT	-
@USER you are also the king of taste	NOT	-	-

Table 1: Examples from the *OLID* dataset.

4.1 PMI

We use a PMI-based model that computes the n-gram-based similarity of a tweet to the tweets of a particular class c in the training dataset. The model is considered naïve as it accounts only for the n-gram frequencies in the discrete token space and only in the context of n neighboring tokens. We compute the PMI score (Turney and Littman, 2003) of each n-gram in the training set w.r.t. each class:

$$PMI(w_i, c_j) = log_2\left(\frac{p(w_i, c_j)}{p(w_i) * p(c_j)}\right) \quad (1)$$

where $p(w_i, c_j)$ is the frequency of n-gram w_i in instances of class c_j, $p(w_i)$ is the frequency of n-gram w_i in instances from the entire training dataset, and $p(c_j)$ is the frequency of class c_j. Additionally, we find that semantically oriented PMI scores (Turney and Littman, 2003) improve the performance of this naïve method:

$$PMI - SO(w_i, c_j) = log_2(\frac{p(w_i, c_j) * p(C \setminus \{c_j\})}{p(w_i, C \setminus \{c_j\}) * p(c_j)}) \quad (2)$$

where $C \setminus \{c_j\}$ is the set of all classes except c_j. At training time, we collect the frequencies of the n-grams on the training set. At inference time, we use the frequencies to calculate PMI and PMI-SO scores for each unigram and bigram in each instance and then average PMI and PMI-SO to a single score for each instance and class. Finally, we select the class with the highest score. If the instance contains no words with associated scores, we choose NOT for Level A, UNT for Level B –

Level	Label	*OLID*		*SOLID*	
		Train	Test	Train	Test
A	OFF	4,640	240	1,448,861	3,002
	NOT	9,460	620	7,640,279	2,991
B	TIN	4,089	213	149,550	1,546
	UNT	551	27	39,424	1,451
C	IND	2,507	100	120,330	1,055
	GRP	1,152	78	22,176	349
	OTH	430	35	7,043	140

Table 2: Train and Test data distribution for the *OLID* and the *SOLID* datasets.

the classes most likely to contain neutral orientation, and the majority class IND for Level C. We remove words appearing less than five times in the training set and add a smoothing factor of 0.01 to all frequencies.

4.2 FastText

A suitable extension to the word-based model is to use subword representations to overcome the naturally noisy structure of tweets. FastText (FT) (Joulin et al., 2017) is a strong subword model which has shown strong performance on various tasks without the need for extensive hyper-parameter tuning. It uses a shallow neural model for text classification similar to the continuous bag-of-words model (Mikolov et al., 2013). Instead of predicting the word based on its neighbors, it predicts the target label based on the sample's words. FT provides a valuable, diverse modeling representation to the ensemble due to its differences with the simple PMI model and the heavy-lifting LSTM and BERT models. We train FT with bigrams and a learning rate of 0.01 for Levels A and B and with trigrams and a learning rate of 0.09 for Level C. All tasks use a window size five and a hierarchical softmax loss.

4.3 LSTM

In contrast to the prior models, the LSTM model (Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017) can account for long-distance relations between words. First is an embedding layer initialized with a concatenation of the GloVe 300-dimensional (Pennington et al., 2014) and FastText's Common Crawl 300-dimensional embeddings (Grave et al., 2018). It is followed by a dropout and a bi-directional LSTM layer with an attention mechanism on top of it. We concatenate the attention mechanism's output with averaged and maximum global poolings on the outputs of the LSTM model. The final prediction is produced by a sigmoid layer for Levels A and B, where we have a binary classification, and a softmax layer for Level C, where we have three classes. We train the LSTM model using early stopping with patience for no improvements over the validation loss of up to five epochs. Level A uses a hidden size of 128, a dropout rate of 0.3, a batch size of 256, and a learning rate of 0.0002. Levels B and C use a hidden size of 50, a dropout rate of 0.1, a batch size of 32, and a learning rate of 0.0001. The Adam optimizer is used for training.

4.4 BERT

The Transformer architecture (Vaswani et al., 2017) has achieved (nearly) state-of-the-art performance for several NLP tasks. It displays both high representational power and robustness across tasks. We exploit the benefits of transfer learning in a low-resource setup by using the pre-trained BERT model (Devlin et al., 2019) and fine-tune it to our task. We use the base uncased model implementation from HuggingFace (Wolf et al., 2020), which has 12 layers, a hidden size of 768, and 12 attention heads, amounting to 110 million parameters. We fine-tune the BERT model for each task, starting from the pre-trained base model. We fine-tune BERT for 2, 3, and 3 epochs for Level A, B, and C, respectively. We use learning rates of 0.00002 for Levels A and B, and 0.00004 for Level C. We apply per-class weights to cope with the data imbalance in Level C as follows: IND=1, GRP=2, OTH=10. We use the Adam optimizer and a linear warm-up schedule with a 0.05 warm-up ratio.

4.5 Democratic Co-training

Democratic co-training (Zhou and Goldman, 2004) is a semi-supervised technique used to create large datasets with noisy labels when provided with a set of diverse models trained in a supervised way. This approach has been successfully applied in tasks like time series prediction with missing data (Mohamed et al., 2007), early prognosis of academic performance (Kostopoulos et al., 2019), and in the health domain (Longstaff et al., 2010). Using models with diverse inductive biases to label the target tweet can help ameliorate the individual model biases and produce predictions with a lower degree of noise.

In our work, we employ democratic co-training to create semi-supervised labels for all three levels of *SOLID* using *OLID* as our seed dataset. Dis-

tant supervision is conducted by the ensemble of models with different inductive biases as follows:

1. Train N *diverse* supervised models $\{M_j(X)\}$, where $j \in [1, N]$ on a labeled dataset $X = \{(x_i, y_i)\}$, where $i \in [1, |X|]$
2. For each example x'_i in the unannotated dataset $X' = \{(x'_i)\}$, $|i \in [1, |X'|])$ and each model M_j, predict the confidence p'^j_i for the positive class.

5 The *SOLID* Dataset

In this section, we describe the process of collecting a large dataset of over 12 million tweets All of the data was labeled using the democratic co-training approach described in the previous section. The statistics for the dataset are shown in Table 2.

5.1 A Large-Scale Dataset of Tweets

We collected our data from Twitter using the Twitter streaming API[3] via Twython[4] in 2019. We search the API using the 20 most common English stopwords (e.g. *the*, *of*, *and*, *to*) to ensure truly random tweets and avoid rate limits. Using stopwords ensures that we are more likely to obtain English tweets as well as a diverse set of random tweets. We kept the stream running the entire time and continuously choose a stopword at random based on its frequency in Project Gutenberg, a sizeable monolingual corpus. We collected 1,000 tweets for each stopword. Thus, tweets, including more frequent stopwords, are collected more frequently. A full list of the stopwords and their frequency is shown in Appendix A.1. We used this approach to help mitigate biases found in *OLID*. *OLID* was collected using a predefined list of keywords that were more likely to retrieve offensive tweets. This caused offensive tweets in *OLID* to be explicit and easier to classify. In contrast, the tweets collected in *SOLID* contain implicit and explicit offensive text. This allows us to study the performance of models in hard classification cases.

We used the `langdetect` tool[5] to select English tweets and discarded tweets with less than 18 characters or less than two words. We substituted all user mentions with `@USER` for anonymization purposes. We also ignored tweets with URLs as those don't tend to be offensive and might be less self-contained, e.g., they could have a link to an article, image, video, etc. Understanding such tweets would require going beyond their purely textual content. In total, we collected over 12 million tweets. We kept 9 million as training data, and we created a new test dataset from a portion of the remaining 3 million tweets.

[3] https://developer.twitter.com/en/docs
[4] https://twython.readthedocs.io
[5] https://pypi.org/project/langdetect/

Model	Level A	Level B	Level C
Majority Baseline	0.419	0.470	0.214
BERT	0.816	0.705	0.568
PMI	0.684	0.498	0.461
LSTM	0.681	0.657	0.585
FastText	0.662	0.470	0.590

Table 3: Macro-F1 score of the models in the democratic co-training ensemble on the *OLID* test set.

5.2 Semi-Supervised Training Dataset

We used the democratic co-training setup described in Subsection 4.5 to create the semi-supervised dataset. We first trained each model on the *OLID* dataset using 10% of the training dataset for validation. The performance of the individual models on the *OLID* dataset is shown in Table 3. BERT is the best model for Level A. The PMI model performs almost on par with the LSTM model. We expect this is due to the size of the dataset and the fact that a simple lexicon of curse words would be highly predictive of the offensive content present in a tweet. The performance of the FastText model is the lowest by 2 points. BERT performs best for Level B, followed by the LSTM model. The task is more challenging at this level for the frequency and n-gram-based approaches of PMI and FastText.

Finally, the overall performance of the models at Level C decreases further. This is expected as the size of the dataset becomes smaller, and the task is a three-way classification, whereas Levels A and B are two-way. BERT and LSTM outperform FastText and PMI, with BERT being the best model. The decrease in the performance in the final level can lead to increased noise in the semi-supervised labels, but we use an ensemble of four models, and we provide the average and the standard deviation of the confidence across the models on each instance to mitigate this. As we show later, these scores can be successfully used to filter out a large amount of noise in the semi-supervised dataset, thus yielding performance improvements.

We compute the aggregated single prediction based on the average and the standard deviation of the confidences predicted by each of the models:

Level	Text	BERT	LSTM	FT	PMI	AVG	STD	Label	E/H
A	@USER he fucking kills me. he knew it was coming	0.919	0.958	0.852	0.509	0.809	0.177	OFF	E
	His kissing days are over, he's a pelican now!	0.659	0.304	0.568	0.523	0.514	0.131	NOT	H
	i think we're all in love with winona ryder	0.060	0.038	0.017	0.480	0.102	0.155	NOT	E
B	Guess I'll just never understand the fucking dynamics	0.901	0.569	0.001	0.617	0.522	0.327	UNT	H
	@USER Government is a bunch of bitches.	0.013	0.221	0.000	0.397	0.158	0.164	TIN	E
	@USER Give me the date. Fuck them other niggas Bro I'm irritated as fuck	0.882	0.666	0.983	0.701	0.808	0.131	TIN	E
C	@USER He was useless stupid guy	0.807	0.915	1.000	0.480	0.801	0.197	IND	E
	It's like mass shootings is the reg in this shit hole country!	0.826	0.479	0.693	0.570	0.642	0.131	OTH	H
	Getting these niggas tatted is a overstatement are ya dead serious	0.700	0.691	0.770	0.491	0.663	0.104	GRP	H

Table 4: Training data aggregation examples. Columns 3-6 show the confidence of each of the models with respect to the positive class in Levels A and B (OFF, UNT) and only for the corresponding class in C (one example for each of the classes: TIN, GRP, OTH). The label column shows manual annotations, and the last column shows whether the tweet is considered Easy (E) or Hard (H) based on its AVG confidence.

SOLID $= \{(x'_i, p'_i)|i \in [1, |SOLID|])\}$, where p'_i = avg$(\{p'^j_i|j \in [1, N]\})$, std$(\{p'^j_i|j \in [1, N]\})$. In particular, we compute the scores based on the confidences for the positive class at Levels A and B and the confidences for the IND, GRP, and OTH classes at Level C. We performed the above aggregation step instead of just using the scores of each model to avoid over-fitting to any particular model in the ensemble. This helps to prevent biases on individual models in future uses of the dataset. Further, the standard deviation and the average scores can be used to filter instances that the models disagree on, thus reducing potential noise in the semi-supervised annotations.

The dataset is labeled using the semi-supervised manner by assigning a Level A label to all the tweets. Then, we select the subset of tweets that are likely to be offensive for all models (BERT and LSTM $\geq .5$, PMI and FT=OFF) as instances that should be assigned a label for Level B. We chose the tweets likely to be TIN at Level B with a standard deviation lower than 0.25 for Level C. Thus, only the instances that are most likely to be offensive are considered at Levels B and C, and only those that are most likely to be offensive and targeted are considered at Level C. The size and the label distribution across the datasets can be found in Table 2 and examples of tweets along with models' prediction confidences can be found in Table 4.

5.3 *SOLID* Test Dataset

The *OLID* test set is very small, particularly for Levels B and C. Therefore, we also annotated a portion of our held-out 3 million tweets to create a new *SOLID* test set to obtain more stable results and to analyze the performance in more detail.

First, all co-authors of this paper (five annotators) annotated 48 tweets that were predicted to be OFF in order to measure inter-annotator agreement (IAA) using $P_0 = \frac{agreement\_per\_annotation}{total\_annotations * num\_annotators}$. We found IAA to be *0.988* for Level A; an almost perfect agreement for OFF/NOT. The IAA for Level B was *0.818*, indicating a good agreement on whether the offensive tweet was TIN/UNT. Finally, for Level C, the IAA was *0.630*, which is lower but still considered reasonable as Level C is more complicated due to its 3-way annotation schema: IND/GRP/OTH. In addition, a tweet may address targets of different types (e.g., both an individual and a group), but only one label can be chosen.

After having observed high IAA, we annotated additional offensive tweets with a single annotation per instance. We divided our Level A data into four portions based on model confidence:

- *if* BERT $\geq .8 \wedge$ PMI $=$ OFF $\wedge$ FT $=$ OFF $\wedge$ LSTM $\geq .8$ *then* ***Easy*** **OFF** [2380 tweets]
- *else if* BERT $\geq .5 \wedge$ PMI $=$ OFF $\wedge$ FT $=$ OFF $\wedge$ LSTM $\geq .5$ *then* ***Hard*** **OFF** [835 tweets]
- *else if* BERT $\leq .2 \wedge$ PMI $=$ NOT $\wedge$ FT $=$ NOT $\wedge$ LSTM $\leq .8$ *then* ***Easy*** **NOT** [2500 tweets]
- *else if* BERT $< .5 \wedge$ PMI $=$ NOT $\wedge$ FT $=$ NOT $\wedge$ LSTM $< .5$ *then* ***Hard*** **NOT** [278 tweets]

Note, PMI $=$ OFF and FT $=$ OFF designates that the model's probability is higher for the OFF class than for the NOT class. We selected the rest of the thresholds after a manual examination of the confidence scores for each model. We chose the threshold where the model is confident and mostly correct.

We annotated 3,493 tweets for Level A. The

#	Type	Prediction	Tweet	Gold Label
1	Easy	OFF	this job got me all the way fucked up real shit	OFF UNT
2	Easy	OFF	@USER It's such a pain in the ass	OFF UNT
3	Easy	OFF	wtf ari her ass tooo big	OFF TIN IND
4	Easy	NOT	This account owner asks for people to think rationally.	NOT
5	Hard	OFF	It sucks feeling so alone in a world full of people	NOT
6	Hard	OFF	@USER We are a country of morons	OFF TIN GRP
7	Hard	NOT	Hate the sin not the sinner...	NOT
8	Hard	NOT	Somebody come get her she's dancing like a stripper	OFF TIN IND

Table 5: Example tweets from the *SOLID Test* dataset and its four subsets. Shown are the difficulty of each subset (Type), the ensemble model prediction for the examples in each subset (Prediction), an example tweet's text, and the manually annotated gold label.

Type	Model Prediction	Gold Label OFF	Gold Label NOT	Total
easy	OFF	2,187	193	2,380
easy	NOT	0	2,500	2,500
hard	OFF	670	165	835
hard	NOT	145	133	278
Total		**3,002**	**2,991**	**5,993**

Table 6: Statistics of the *SOLID Test* dataset grouped by difficulty (Type) and model prediction.

number of annotations at each level is shown above in square brackets. Furthermore, to create a complete test dataset for Level A (where we only labeled offensive tweets), we also took a random set of 2,500 *Easy* NOT tweets. The resulting test sizes are shown in Table 2. Of the 3,493 annotated tweets, 491 were determined to be NOT. In total, there are 5,993 tweets in our test set. In all cases, all three levels were annotated, but the decision of whether a tweet in Level B/C is *Easy* or *Hard* is still based on its Level A confidence.

Table 5 shows some tweets and whether they are *Easy* OFF/NOT (lines 1-4) or *Hard* OFF/NOT (lines 5-8), and Table 6 shows statistics regarding the Easy and Hard examples in the test dataset. Note that determining the labels for the *Hard* examples is not simple and the model does make incorrect predictions such as in lines 5 and 8 of Table 5. In fact, 25% of the *Hard* OFF tweets that we annotated were NOT. In contrast, 8% of the *Easy* OFF tweets were judged to be NOT.

6 Experiments and Evaluation

In this section we describe our experiments and results when training with *OLID* + *SOLID* data compared to just *OLID* on the *OLID* test set.

6.1 Experimental Setup

We used the BERT and FastText models from the semi-supervised annotation set up to estimate the performance improvements when training on the supervised dataset *OLID* together with the semi-supervised *SOLID*. The models in all sets of experiments were fine-tuned on a 10% validation split of the training set used during co-training. We explored different schemes to combine *OLID* and *SOLID*, as well as different thresholds for the confidence of the instances in *SOLID*. We achieved improvements for Levels B and C by upsampling the underrepresented classes: we sampled K instances of each class, where K is the number of instances for the most frequent class. We also removed the warm-up in Levels B and C, which improved the results further.

FastText. The FastText model used an external command-line tool without control over the training. Therefore, we merged the training splits of *OLID* and *SOLID*, randomly shuffled them, and trained models with the combined dataset. The FastText model has the same parameters used above in co-training.

BERT. Due to the computational requirements of BERT, we subsampled 20,000 tweets from *SOLID* in Level A and B for BERT. Including more semi-supervised instances did not improve the performance. During training, we used *SOLID* in the first epoch and *OLID* in the following two epochs for Level A. Using *SOLID* after training with *OLID* yielded worse results. We assume this can be explained by the fact that the semi-supervised dataset by construction contains labels that are not golden truth. It can be used as an initial step to guide the model towards a better local minimum. On the other hand, we conjecture that the supervised

Level	Baseline	BERT		FastText	
		OLID	*+SOLID*	*OLID*	*+SOLID*
A	0.419	**0.816**	0.809	0.662	**0.720**
B	0.470	0.687	**0.729**	0.470	**0.591**
C	0.214	0.589	**0.643**	**0.590**	0.515

Table 7: Macro-F1 score on the *OLID* test set for BERT and FastText with and without training on *SOLID* compared to the majority class baseline.

dataset is better suited for fine-tuning the model towards the local minimum with the gold data, particularly in Level A, where the training split of *OLID* is already sufficient for training BERT. For Levels B and C, we trained for two epochs with the training split of *OLID* and then for one epoch with *SOLID*. At Levels B and C, we observed that training with *SOLID* in the first epochs and then fine-tuning with *OLID* did not improve the performance. Furthermore, training with *OLID* and then using *SOLID* for the final epochs yielded substantial performance improvements. We assume this is due to the small training size of *OLID* which can cause the model to overfit to a suboptimal local minimum when used in the final training epochs.

Selecting *SOLID* Instances. We filter the training instances from *SOLID* to be the most confident examples based on the average probability score provided in *SOLID* when training with FastText and BERT. We choose the threshold for the average confidence score based on the validation dataset as follows:

Level A: $avg(\text{OFF}) < 0.20 \cup avg(\text{OFF}) > 0.70$

Level B: $avg(\text{UNT}) < 0.35 \cup avg(\text{UNT}) > 0.65$

Level C: $avg(\text{IND}) > 0.80 \cup avg(\text{GRP}) > 0.70 \cup avg(\text{OTH}) > 0.65$

To select a label for each instance, we choose: NOT when $avg(\text{OFF}) < 0.20$, otherwise OFF in Level A; UNT when $avg(\text{UNT}) > 0.65$, otherwise TIN in Level B; the class with the highest probability in Level C.

6.2 *OLID* Results

In this section we describe our results on the *OLID* test set using just *OLID* and adding *SOLID*. The results are shown in Table 7.

The results for Level A are improved only with FastText, which is a weaker model (see Table 3). It leverages a large performance improvement when trained with *OLID+SOLID*. On the other hand, the BERT model already achieves high performance without augmenting *OLID* with *SOLID* because *OLID* is large enough for Level A. As a result, including semi-supervised data did not improve the performance. Our findings are in line with the study of Longstaff et al. (2010), who observed that democratic co-training performs well when the initial classifier's accuracy was low.

	Model	Baseline	BERT		FastText	
			OLID	*+SOLID*	*OLID*	*+SOLID*
A	Full	0.338	0.922	**0.923**	0.856	**0.860**
	Easy	0.400	0.983	**0.983**	0.936	**0.940**
	Hard	0.444	0.557	**0.570**	0.525	**0.536**
B	Full	0.236	0.559	**0.666**	0.355	**0.493**
	Easy	0.232	0.569	**0.677**	0.349	**0.509**
	Hard	0.234	0.542	**0.649**	0.363	**0.467**
C	Full	0.203	0.627	**0.645**	0.387	**0.504**
	Easy	0.201	0.635	**0.644**	0.378	**0.504**
	Hard	0.205	0.616	**0.649**	0.397	**0.505**

Table 8: Experimental results (macro-F1 scores) on the *SOLID Test* dataset, and on its Easy and Hard subsets, compared to the majority class baseline.

The *OLID* training dataset is smaller for Level B, and the task is more complex. Moreover, the FastText model here performs on par with the majority class baseline. Augmenting *OLID* with *SOLID* yields performance improvements for both models. We achieve an improvement of 0.042 points for BERT and a large margin of improvement of 0.121 points for FastText.

Finally, in Level C, the supervised *OLID* data is even smaller, and the complexity of the subtask is more pronounced, mainly due to it having three possible labels. Interestingly, using *SOLID* for FastText does not yield better results. This might be due to the model already achieving high performance on par with BERT (see Table 3), while democratic co-training performs well when the initial classifier's performance is low. Additionally, this may be due to the instability of the test set for Level C, which is very small. On the other hand, the *SOLID* data helps the BERT model by a large margin of 0.054 points.

6.3 *SOLID* Results

In the previous section, we showed noticeable improvements on the *OLID* dataset using *SOLID*. However, *OLID* is small (particularly for Levels B and C). Showcasing the performance on a larger test set, *SOLID* test, is important for estimating the models' stability. We also focus on *Easy* vs. *Hard* examples (based on the confidence computed during co-training) to gain better insight into why

some tweets are easier to predict as offensive than others. Results are shown in Table 8 and significantly beat the majority baselines.

The overall Level A results on *SOLID* test are 92.3% and 86.0% macro-F1 for BERT and FastText, respectively, with a small improvement when *OLID* is augmented with *SOLID* for FastText only. This is consistent with what we found on the *OLID* test set. Note that the full results for Level A are much better than on the OLID dataset in Table 7. We expect that this is partially due to our selection of tweets for the new test set, indicating that there are more *Easy* tweets in it. Similar findings to the full test set occur with the *Easy* tweets, but the scores are even higher. On the other hand, for the *Hard* tweets, the results are much lower at 57% and 53.6% for BERT and FastText, respectively. Using *SOLID* yields a nice improvement for both models on the *Hard* tweets, which was not evident in the *OLID* test set in Table 7.

To provide further insight into why the results are so high for *Easy* OFF tweets in Level A, we implemented a curse word baseline using the absence or presence of 22 curse words like *fuck*, *bitch*, and *nigga*. A full list of the curse words used in the baseline can be found in Appendix A.1. We found that most *Easy* tweets were classified correctly with this baseline with 93.6% F1-score. In contrast, the curse word baseline was not effective on the hard examples, just like the BERT and FastText models. It achieved a macro-F1 score of 58%, which is one point higher than the BERT result. The BERT and FastText models are clearly overfitting to the curse words. The *hard* tweets are offensive due to other language use such as negative biases rather than the appearance of a curse word such as examples 6 and 8 in Table 5. Classifying these tweets successfully remains an open challenge.

The difference between *Easy* OFF/NOT and *Hard* OFF/NOT tweets is less pronounced for Levels B and C. The curse word imbalance may have a small impact on the lower levels as UNT tweets are more likely to contain curse words. In all cases, including *SOLID* with *OLID* for Levels B and C yields a nice improvement, indicating that the larger test set can better showcase the improvements, leading to more stability. The results for Levels B and C vary greatly for the two models compared to those on the *OLID* test set in Table 7, which points to the challenges of having a very small test set.

7 Conclusion and Future Work

We presented *SOLID*, a large-scale semi-supervised training dataset for offensive language identification, which we created using an ensemble of four different models. To the best of our knowledge, *SOLID* is the largest dataset of its kind, containing nine million English tweets. We have shown that using *SOLID* yields noticeable performance improvements for Levels B and C of the OLID annotation schema, as measured on the *OLID* test set. Furthermore, in contrast to using keywords, our approach allows us to distinguish between *Hard* and *Easy* offensive tweets. The latter enables us to have a deeper understanding of offensive language identification and indicates that detecting *Hard* offensive tweets is still an open challenge. Our work encourages safe and positive places on the web that are free of offensive content, especially non-obvious cases (i.e., *Hard*). *SOLID* is the official dataset of the SemEval shared task OffensEval 2020 (Zampieri et al., 2020). In the future, we would like to provide insights and methods for categorizing *Hard* tweets.

Acknowledgements

We would like to thank the anonymous reviewers who provided us with constructive and insightful feedback to further improve the quality of the paper. PA has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199. This work is also part of the Tanbih mega-project, developed at the Qatar Computing Research Institute, HBKU, which aims to limit the impact of "fake news," propaganda, and media bias by making users aware of what they are reading.

Ethics Statement

Dataset Collection The *OLID* and the *SOLID* datasets were collected using the Twitter API[6]. The *OLID* dataset was collected using keywords that would be more likely to be accompanied with offensive tweets (Zampieri et al., 2019a). The *SOLID* dataset was collected using frequent stop words (See Section 5) intending to aid in detecting offensive tweets in English only, We follow the terms of use outlined by Twitter[7]. Specifically, we only

[6] http://developer.twitter.com/en/docs

[7] http://developer.twitter.com/en/developer-terms/agreement-and-policy

download public tweets and we provide only the user ids of the tweets to ensure that deleted tweets will no longer be available in our dataset. Further, in all our examples in this paper, we anonymize the user names in the tweets. Since no private information is stored, IRB approval is not required. All annotations were performed internally by the authors of the paper.

Biases *SOLID* is a large-scale semi-supervised dataset for offensive language detection We note that determining whether a piece of text is offensive can be subjective, and thus it is inevitable that there would be biases in our gold-labeled data. It is expected that these biases will, therefore, also be present in the semi-supervised dataset we generated from such tweets.

While we cannot ensure that no biases occur in the gold data, we address these concerns by following a well-defined schema, which sets explicit definitions for offensive content during annotation. Our high inter-annotator agreement makes us confident that the assignment of the schema to the data is correct most of the time.

Using semi-supervised techniques to create a large dataset, *SOLID*, can cause the biases found in the gold data to be expanded further. We mitigate this in two ways. First, we gather tweets based on the most frequent words in English to ensure a random set of initial tweets. Next, we construct an ensemble of models with diverse inductive biases to label the target tweet, which can help to ameliorate the individual model biases and to produce predictions with a lower degree of noise. At test time, we aim to have a meaningful ratio of offensive and non-offensive tweets based on a random collection of tweets. We also label all test offensive tweets manually. The aim of these steps was to help reduce the potential biases. Please refer to Section A.2 of the Appendix for some analysis that indicates the diversity of the models.

We acknowledge that current semi-supervised techniques do not address the problem of the bias inherent in the semi-supervised data coming from the supervised source model(s), which can also be studied in future work. Further, we acknowledge that biases can still exist in the ratio of offensive/non-offensive tweets. The size of the data and the method of collection for the SOLID dataset mean that biases are hard to avoid.

In addition, offensive language can vary depending on demographics, such as the gender of the targeted individual and the target can even be a particular gender group. Such biases that are present in natural language data (Olteanu et al., 2019) is an attractive future study.

Misuse Potential Most datasets compiled from social media present some risk of misuse. We therefore ask researchers to be aware that the *SOLID* dataset can be maliciously used to unfairly moderate text (e.g., a tweet) that may not be offensive based on biases that may or may not be related to demographics and other information within the text. Intervention with human moderation would be required in order to ensure this does not occur.

Intended Use We present *SOLID* to encourage research in automatically detecting and stopping offensive content from being disseminated on the web. Such systems can be used to alleviate the burden for media moderators, which can suffer from psychological disorders due to the exposure of extremely offensive content. Improving the performance of offensive content detection systems can decrease the amount of work for moderators, but human supervision is required for more intricate cases and to ensure that the system is not causing harm. With the possible ramifications of a highly subjective dataset, we distribute *SOLID* for research purposes only, without a license for commercial use. Any biases found in the dataset are unintentional, and we do not intend to cause harm to any group or individual.

We believe that this dataset is a useful resource when used in the appropriate manner with great potential to improve the performance of current offensive content detection systems.

References

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 the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. 2019. CONAN -

COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy. Association for Computational Linguistics.

Çağrı Çöltekin. 2020. A corpus of Turkish offensive language on social media. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6174–6184, Marseille, France. European Language Resources Association.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, page 693–696, Berlin, Heidelberg. Springer-Verlag.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

European Commission. 2020. Proposal for a regulation of the european parliament and of the council on a single market for digital services (digital services act) and amending directive 2000/31/ec. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52020PC0825.

Paula Fortuna, José Ferreira, Luiz Pires, Guilherme Routar, and Sérgio Nunes. 2018. Merging datasets for aggressive text identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 128–139, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

HM Government. 2019. Online harms white paper. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/793360/Online_Harms_White_Paper.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Georgios Kostopoulos, Stamatis Karlos, and Sotiris Kotsiantis. 2019. Multiview learning for early prognosis of academic performance: A case study. *IEEE Transactions on Learning Technologies*, 12(2):212–224.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 1–5, Marseille, France. European Language Resources Association (ELRA).

Brent Longstaff, Sasank Reddy, and Deborah Estrin. 2010. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–7.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

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*.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*, pages 1–12.

Tawfik A. Mohamed, Neamat El Gayar, and Amir F. Atiya. 2007. A co-training approach for time series prediction with missing data. In *Multiple Classifier Systems*, pages 93–102, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2021. Arabic offensive language on Twitter: Analysis and experiments. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 126–135, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Alexandra Olteanu, C. Castillo, Fernando D. Diaz, and E. Kiciman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Social Science Research Network*, 2:13.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4675–4684, Hong Kong, China. Association for Computational Linguistics.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Zesis Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive language identification in Greek. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5113–5119, Marseille, France. European Language Resources Association.

Rahat Ibn Rafiq, Homa Hosseinmardi, Richard Han, Qin Lv, and Shivakant Mishra. 2018. Scalable and Timely Detection of Cyberbullying in Online Social Networks. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 1738–1747, New York, NY, USA. Association for Computing Machinery.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5838–5844, Online. Association for Computational Linguistics.

Niloofar Safi Samghabadi, Adrián Pastor López Monroy, and Thamar Solorio. 2020. Detecting early signs of cyberbullying in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 144–149, Marseille, France. European Language Resources Association (ELRA).

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive language and hate speech detection for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3498–3508, Marseille, France. European Language Resources Association.

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 shared task on the identification of offensive language. In *Proceedings of the GermEval 2018 Workshop (GermEval)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 656–666, Montréal, Canada. Association for Computational Linguistics.

Mengfan Yao, Charalampos Chelmis, and Daphney Stavroula Zois. 2019. Cyberbullying ends here: Towards robust detection of cyberbullying in social media. In *The World Wide Web Conference*, WWW '19, page 3427–3433, New York, NY, USA. Association for Computing Machinery.

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 the 2019 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

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 the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

Yan Zhou and Sally Goldman. 2004. Democratic co-learning. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '04, page 594–202, USA. IEEE Computer Society.

A Appendices

A.1 Data Collection and Analysis

Section 5.1 we described our method for collecting tweets. We collect tweets using the most frequent English words based on the large monolingual Project Gutenberg corpus.[8] Table 9 shows the top-20 most frequent words in the corpus and their frequency which we used to collect tweets. The normalized value is the percentage of the total frequency for all top 20 words. We randomly pick a number between 0 and 1, and choose the word based on the normalized value. For example, .45 would be "and".

In Section 6.3, we discussed the simple curse word baseline used to analyze the *Easy* OFF/NOT tweets. Table 10 lists the 22 curse words used in the baseline.

A.2 Implementation Details

The fine-tuning of the models was performed on a 10% split from the *OLID* dataset. All models were trained on an NVIDIA Titan X GPU with 8GB of RAM.

[8] https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists#Project_Gutenberg

w	frequency	norm.	w	frequency	norm.
the	56,271,872	0.20	it	8,058,110	0.79
of	33,950,064	0.32	with	7,725,512	0.82
and	29,944,184	0.43	is	7,557,477	0.85
to	25,956,096	0.52	for	7,097,981	0.87
in	17,420,636	0.58	as	7,037,543	0.90
i	11,764,797	0.63	had	6,139,336	0.92
that	11,073,318	0.67	you	6,048,903	0.94
was	10,078,245	0.70	not	5,741,803	0.96
his	8,799,755	0.73	be	5,662,527	0.98
he	8,397,205	0.76	her	5,202,501	1.00

Table 9: The top-20 most frequent English words (*w*). *Norm.* is the normalized value based on the total frequency of all the top words. The random number generated between 0 and 1 determines which word is chosen.

ass	arse	wtf	lmao	fuck
bitch	nigga	nigger	cunt	effing
shit	hell	damn	crap	bastard
idiot	stupid	racist	dumb	f*ck
pussy	dick			

Table 10: The 22 common offensive terms used in the curse word baseline.

The evaluation metric used for all experiments is macro F1 from scikit-learn.[9]. The performance of the models in the ensemble used for semi-supervised labelling is provided in Table 11.

Model	A	B	C
BERT	0.788	0.610	0.577
PMI	0.772	0.595	0.536
LSTM	0.599	0.599	0.579
FastText	0.672	0.489	0.456

Table 11: F1 score performance of each model used in the ensemble on the validation dataset of Levels A, B, and C.

In Table 12 we show the agreement of the models for the task prediction. For Levels A and B, it is more common that all four models agree, while in Level C, there are more cases when at least one model disagrees with the rest models. Furthermore, in Level A, there are almost no cases when the decision is tied with two models disagreeing with the other two. Finally, as in Level C, the performance of the models is lower, the disagreement between the models in the ensemble is the largest and it is least common for all four models to agree on

[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

a prediction. Given the observed agreement rates, we conclude that there is considerable variance in the predictions across the models, especially for the lower levels. We assume this indicates that the separate models have different rationales to a certain degree, which can be avoided by the ensemble combination of the models.

N	A	B	C
4	0.517	0.598	0.249
3	0.392	0.275	0.417
2	0.091	0.127	0.335

Table 12: Percentage of instances where N models agree for a predicted label of an instance, N $\in \{2, 3, 4\}$, for Levels A, B, and C.