From the Detection of Toxic Spans in Online Discussions to the Analysis of Toxic-to-Civil Transfer

Anonymous ACL submission

Abstract

We study the task of toxic spans detection, which concerns the detection of the spans that make a text toxic, when detecting such spans is possible. We introduce a dataset for this task, TOXICSPANS, which we release publicly. By experimenting with several methods, we show that sequence labeling models perform best, but methods that add generic rationale extraction mechanisms on top of classifiers trained to predict if a post is toxic or not are also surprisingly promising. Finally, we use TOXICSPANS and systems trained on it, to provide further analysis of state-of-the-art toxic to non-toxic transfer systems, as well as human performance on that latter task. Our work highlights challenges in finer toxicity detection and mitigation.

1 Introduction

In social media and online fora, toxic content can be defined as rude, disrespectful, or unreasonable posts that would make users want to leave the conversation (Borkan et al., 2019). Although several toxicity detection datasets (Wulczyn et al., 2017; Borkan et al., 2019) and models (Schmidt and Wiegand, 2017; Pavlopoulos et al., 2017c; Zampieri et al., 2019) exist, most of them classify whole posts, without identifying the specific *spans that make a text toxic*. But highlighting such toxic spans can assist human moderators (e.g., news portal moderators) who often deal with lengthy comments, and who prefer attribution instead of just a system-generated unexplained toxicity score per post. Locating toxic spans within a text is thus a crucial step towards successful semi-automated moderation and healthier online discussions.

To promote research on this new task, we release the first dataset of English posts with annotations of toxic spans, called TOXICSPANS.[1] We discuss how it was created and we propose an evaluation framework for toxic spans detection. We consider methods that (i) perform sequence labeling (tag words) or (ii) rely on an attentional binary classifier to predict if a post is toxic or not, then invoke its attention at inference time to obtain toxic spans as in rationale extraction. The latter approach allows leveraging larger existing training datasets, which provide gold labels indicating which posts are toxic or not, without providing gold toxic span annotations. Although sequence labeling performed overall better, the binary attentional classifier performed surprisingly well too, despite having been trained on data without span annotations.

We then study some characteristics of supervised and self-supervised toxic-to-civil transfer models (Laugier et al., 2021) by comparing them on several datasets, including a recently released parallel toxic-to-civil dataset (Dementieva et al., 2021) and the new TOXICSPANS dataset. Using the latter, we introduce a measure to evaluate the elimination of *explicit* toxicity, and we use this measure to compare the behavior and performance of toxic-to-civil models. Lastly, by applying toxic span detection systems, we assess the performance of human crowdworkers on the toxic-to-civil task.

2 Related work

Toxicity detection systems (Schmidt and Wiegand, 2017; Pavlopoulos et al., 2017c; Zampieri et al., 2019) are typically trained on datasets annotated at the post level (a text is annotated as toxic or not) (Wulczyn et al., 2017; Borkan et al., 2019). Our work differs from general toxicity detection in that we detect toxic *spans*, instead of assigning toxicity labels to entire texts. Toxic spans detection can be seen as a case of attribution or rationale extraction (Li et al., 2016; Ribeiro et al., 2016), but specifically for toxic posts, a task that has never been

[1] URL hidden to avoid revealing the identity of the authors. Part of the dataset was used in a challenge with the permission of the authors. We do not provide further information about the challenge to preserve anonymity. The full dataset and the code of this work will be released with a CC0 licence.

Gold Spans (set of char offsets)	Post
{55, 56, 57, 58, 59, 60}	What if his opinion is that most other commenters are **idiots**? :-)
{80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 176, 177, 178, 179, 180}	Survival of the fittest would not have produced you. You are alive because your **weak blood** is supported by welfare and food stamps. Please don't reference Darwin in your icon. **Loser**.
{ }	So tired of all these Portlanders moving to California and ruining the culture. When will it stop?!?

Table 1: Examples of toxic posts and their ground truth toxic spans (also shown in bold red). In the left column, toxic spans are shown as sets of character offsets. No toxic spans are included in the ground truth of the last post.

considered in general toxicity detection before.

Detecting spans, instead of entire posts, was recently also considered in propaganda (Martino et al., 2020) and hate speech detection (Mathew et al., 2021). Although the ground truth type is similar (spans), propaganda detection is a different task from ours. Hate speech is a particular type of toxicity (Borkan et al., 2019), which can be tackled by more general toxicity detectors (Van Aken et al., 2018), but not the other way round; i.e., we address a broader problem. This probably explains why a pattern-matching baseline, based on the data of Mathew et al. (2021), achieved only slightly better results than a random baseline on our dataset.

Suggesting civil rephrases of posts found to be toxic (Nogueira dos Santos et al., 2018; Laugier et al., 2021) is the next step towards healthier online discussions, and can be viewed as a form of style transfer (Shen et al., 2017; Fu et al., 2018; Lample et al., 2019). We show how toxic spans detection can also contribute in the assessment of toxic-to-civil transfer, linking the two tasks together for the first time.

3 The new TOXICSPANS dataset

We used posts (comments) from the publicly available Civil Comments dataset (Borkan et al., 2019), which already provides whole-post toxicity annotations. We followed the toxicity definition that was used in Civil Comments, i.e., we use 'toxic' as an umbrella term that covers abusive language phenomena, such as insults, hate speech, identity attack, or profanity. This definition of toxicity has been used extensively in previous work (Hosseini et al., 2017; Van Aken et al., 2018; Karan and Šnajder, 2019; Han and Tsvetkov, 2020; Pavlopoulos et al., 2020). We asked crowd annotators to highlight the spans that constitute "anything that is rude, disrespectful, or unreasonable that would make someone want to leave a conversation". Besides toxicity our annotators were also asked to select a subtype for each highlighted span, choosing between insult, threat, identity-based attack, profane/obscene, or other toxicity. Asking the annotators to also select a category was intended as a priming exercise to increase their engagement, but it may have also helped them align their notions of toxicity further, increasing inter-annotator agreement. For the purposes of our experiments, we collapsed all the subtypes into a single toxic class, and we did not study them further; but the subtypes are included in the new dataset we release.

Annotation From the original Civil Comments dataset (1.2M posts), we retained only posts that had been found toxic by at least half of the crowd-raters. This left approximately 30k toxic posts. We selected a random 11k subset of the 30k posts for toxic spans annotation. We used the crowd-annotation platform of Appen.[2] We employed three crowd-raters per post, all of whom were warned for explicit content. Raters were selected from the smallest group of the most experienced and accurate contributors. The raters were asked to mark the toxic word sequences (spans) of each post by highlighting each toxic span on their screen. If they believed a post was not actually toxic, or that the entire post would have to be annotated, they were instructed to select the appropriate tick-boxes in the interface, without highlighting any span.

It is not possible to annotate toxic spans for every toxic post. For example, in some posts the core message being conveyed may be inherently toxic (e.g., a sarcastic post indirectly claiming that people of a particular origin are inferior) and, hence, it may be difficult to attribute the toxicity of those posts to particular spans. In such cases, the posts may end up having no toxic span annotations, according to the guidelines given to the annotators; see the last post of Table 1 for an example. In other cases, however, it is easier to identify particular spans (possibly multiple per post) that make a post toxic, and these toxic spans often cover only a small part of the post (see Table 1 for examples).

[2] https://appen.com/

Agreement We measured inter-annotator agreement on 87 randomly selected posts of our dataset, using 5 crowd-annotators per post in this case. We calculated the mean pairwise (for a pair of annotators) Cohen's kappa per post, using character offsets as instances being classified as toxic (included in a toxic span) or non-toxic; we then averaged over the posts. Although our dataset contains only posts found toxic by at least half of the original crowd-raters, only 31 of the 87 posts were found toxic by all five of our annotators, and 51 were found toxic by the majority of our annotators; this is an indicator of the well-known subjectivity of toxicity detection. On the 31, 51, and 87 posts, the average kappa score was 65%, 55%, 48%, respectively, indicating that when the raters agree (at least by majority) about the toxicity of the post, there is also reasonable agreement regarding the toxic spans. Note that the toxic spans are typically short. This leads to class imbalance (most offsets are marked as non-toxic), increases agreement by chance (on the non-toxic offsets), and leads to low kappa scores (kappa adjusts for chance agreement). Another reason behind this modest (compared to other tasks) inter-annotator agreement is the inherent subjectivity of deciding if a post is toxic or not. Our kappa score is in fact slightly higher than in previous work on toxicity detection, classifying posts as toxic or not (Sap et al., 2020; Pavlopoulos et al., 2017a), and in that sense our inter-annotator agreement can be seen as an improvement.

Ground truth To obtain the ground truth of our dataset, we averaged the labels per character of the annotators per post. We used the following process: for each post t, first we mapped each annotated span of each rater to its character offsets. We then assigned a toxicity score to each character offset of t, computed as the fraction of raters who annotated that character offset as toxic (included it in their toxic spans). We retained only character offsets with toxicity scores higher than 50%; i.e., at least two raters must have included each character offset in their spans. Table 1 shows examples.

The dataset TOXICSPANS contains the 11,035 posts we annotated for toxic spans. The unique posts are actually 11,006, since a few were duplicates and were removed; a few other posts were used as quiz questions to check the reliability of candidate annotators and have also been removed.

Exploratory analysis Although we instructed the crowd-raters to click the appropriate tick-box and not highlight any span when the whole post would have to be highlighted, the ground truth of 34 out of the 11k posts covers the entire post. However, 14 out of the 34 posts are single-word texts, while the other posts are very short (Appendix A shows more details); it seems that in very short posts the raters sometimes did not realize they ended up highlighting the entire post. Furthermore, about 5k of the 11k posts have an empty ground truth set of toxic character offsets (as in the last post of Table 1), even though all the posts of our dataset had been found toxic by the original raters. This is partly due to the fact that we include in the ground truth only character offsets that were included in the toxic spans of the majority of our annotators. It also confirms it is not always possible to attribute (at least not by consensus) the toxicity of a post to particular toxic spans. In almost all posts, the ground truth covers less than half of the post; and in the vast majority, less than 20% of the post. A *dense toxic span* of a post is a maximal sequence of contiguous toxic characters. There exist posts with more than one dense toxic span, but most posts include only one. Table 2 provides further statistics.

4 Evaluation framework for toxic spans

For the newly introduced toxic spans detection task, we evaluate systems in terms of F_1 score, as in the work of Da San Martino et al. (2019). Given a test post t, let system A_i return a set $S^t_{A_i}$ of character offsets, for parts of the post found to be toxic. Let S^t_G be the character offsets of the ground truth annotations of t. We compute the F_1 score of system A_i with respect to the ground truth G for post t:

$$F_1^t(A_i, G) = \frac{2 \cdot P^t(A_i, G) \cdot R^t(A_i, G)}{P^t(A_i, G) + R^t(A_i, G)} \quad (1)$$

$$P^t(A_i, G) = \frac{|S^t_{A_i} \cap S^t_G|}{|S^t_{A_i}|}, R^t(A_i, G) = \frac{|S^t_{A_i} \cap S^t_G|}{|S^t_G|} \quad (2)$$

If S^t_G is empty for some post t (no gold spans are given for t), we set $F_1^t(A_i, G) = 1$ if $S^t_{A_i}$ is also empty, and $F_1^t(A_i, G) = 0$ otherwise. We average $F_1^t(A_i, G)$ over all test posts t to obtain a single score for system A_i. We use F_1 as the main evaluation measure in experiments reported below.

5 Methods for toxic spans detection

TRAIN-MATCH, classifies as toxic any tokens encountered inside toxic spans of the training data.

	Mean	Min	Max
Post length	208.14	4	1,000
Dense toxic span length	7.01	3	87
# Dense toxic spans	0.58	0	8

Table 2: TOXICSPANS statistics. Lengths in characters.

HATE-MATCH operates similarly but the lookup is within the hateful/offensive spans of the data of Mathew et al. (2021). A naive baseline, RAND-SEQ, randomly classifies tokens as toxic or not.

5.1 Supervised sequence labelling

Toxic spans detection can be seen as sequence labeling (tagging words). As a baseline of this kind, we employ SPACY'S Convolutional Neural Network, which is pre-trained for tagging, parsing, entity recognition (Honnibal and Montani, 2017). We call this model CNN-SEQ and fine-tune it on dense toxic spans, treated as 'entities'. We also train a bidirectional LSTM (BILSTM-SEQ),[3] and fine-tune BERT (Devlin et al., 2019) and SPAN-BERT (Joshi et al., 2020) for toxic spans (BERT-SEQ, SPAN-BERT-SEQ).[4] These methods require training data manually annotated with toxic spans.

5.2 Weakly supervised learning

We trained binary classifiers to predict the toxicity label of each post, and we employed attention as a rationale extraction mechanism at inference to obtain toxic spans, an approach Pavlopoulos et al. (2017b) found to work reasonably well in toxicity detection.[5] We experimented with two classifiers: a BILSTM with deep self-attention as in the work of Pavlopoulos et al. (2017b), but training with a regression objective and probabilistic labels following D'Sa et al. (2020) and Wulczyn et al. (2017); and BERT with a dense layer and sigmoid on the [CLS] embedding. To detect toxic spans, we used the attention scores of the BILSTM and the attention scores from the heads of BERT's last layer averaged over the heads, respectively. In both cases, we obtain a sequence of binary decisions (toxic, non-toxic) for the *tokens* of the post (inherited by their character offsets) by using a probability threshold (tuned on development data) applied to the attention scores. We refer to these two attention-based rationale extraction methods as BILSTM+ARE and BERT+ARE, respectively. These methods require training posts annotated only with toxicity labels per *post* (no toxic span annotations).

[3] We used the probabilistic ground truth for training and mean square error as the loss function of BILSTM-SEQ, which yielded best results in preliminary experiments.

[4] More details can be found in the Appendix A.3.

[5] See Wiegreffe and Pinter (2019); Kobayashi et al. (2020); Ferrando and Costa-jussà (2021) for a broader discussion of attention as an explainability mechanism.

	$F1$ (%)	P (%)	R (%)
BILSTM-SEQ	58.9	59.8	58.9
CNN-SEQ	59.3	60.7	59.0
BERT-SEQ	59.7	60.7	60.0
SPAN-BERT-SEQ	**63.0**	**63.8**	**62.8**
BILSTM+ARE	57.7	58.4	57.3
BERT+ARE	49.1	49.4	49.5
RAND	7.3	5.3	25.4
TRAIN-MATCH	41.0	39.1	48.7
HATE-MATCH	10.6	7.1	43.7

Table 3: F_1, Precision (P), Recall (R) of sequence labeling (1st zone), attentional (2nd), and look-up methods (3rd) in toxic spans detection. Average scores of a 5-fold Monte Carlo C-V shown. The standard error of mean is always lower than a percentage point. The ROC AUC scores of BILSTM and BERT (of ARE-based methods) in *toxic/non-toxic text classification* are 90.9% and 96.1%.

6 Experimental results for toxic spans

We used a 5-fold Monte Carlo cross-validation (5 random training/development/test splits) on the 11k posts of TOXICSPANS. In each fold, we use 10% of the data for testing, 10% for development, and 80% for training. In ARE-based methods, which rely on an underlying classifier to predict if a post is toxic or not, the classifier is trained on the training part of the fold (which contains only toxic posts, ignoring the toxic span annotations) and a randomly selected equal number of non-toxic posts from Civil Comments that are not included in our dataset. When measuring the (binary) classification performance of the underlying classifier, the classifier is evaluated on a new equally balanced test set of 3k randomly sampled unseen posts from Civil Comments.

Both look-up methods (TRAIN-MATCH, HATE-MATCH) outperform the random baseline (Table 3). However, TRAIN-MATCH performs much better, which agrees with our hypothesis that toxicity detection is a broader problem than hate speech detection. Both look-up methods are outperformed by the sequence labeling models (-SEQ), especially SPAN-BERT-SEQ, which is pre-trained to predict spans. These results show that the tokens of toxic spans are context-dependent and their meaning is not captured well by context-unaware look-up lexicons. An error analysis of the best-performing

You can stick your dick up anyone's butt. Why have any laws at all?
Not sure if "people are **dumb**" is the best descriptor, but you are correct that we tend to seek out and grasp at anything that supports our beliefs and hopes. Hence the proliferation of "fake news", which feeds those wants.
They can shuffle the cabinet seven ways from Sunday and it's still a cabal of **losers**.

Table 4: Examples of posts with toxic posts (ground truth in red) which SPAN-BERT-SEQ predicted (in bold) incorrectly. The ground truth is empty (no toxic spans) in the two last posts.

SPAN-BERT-SEQ showed that its most common mistakes are false negatives (e.g., incorrectly returning an empty span, 1st row of Table 4) and false positives (2nd and 3rd row). BERT+ARE performs worse than BILSTM+ARE, despite the fact that the underlying BERT classifier is much better (ROC AUC 96.1%) at separating toxic from non-toxic *posts* than the underlying BILSTM (90.9%). Interestingly, the BILSTM binary toxicity classifier with the attention-based toxic span detection mechanism (Pavlopoulos et al., 2017b) is close in performance with BILSTM-SEQ, despite the fact that the latter is directly trained on toxic span annotations, whereas the former is trained with binary post-level annotations only (toxic, non-toxic *post*).

Several large datasets with *post-level* toxicity annotations are publicly available (Pavlopoulos et al., 2019). Therefore, attribution-based toxic span detectors, such as BILSTM+ARE, can in principle perform even better if the underlying binary classifier is trained on a larger existing dataset. To investigate this, we increased the training set of the underlying BILSTM classifier of BILSTM+ARE. We added to the training set of each cross-validation fold 80k further toxic and non-toxic posts (still equally balanced, without toxic spans) from the dataset of Borkan et al. (2019), excluding posts used in TOXICSPANS. The ROC AUC score of the underlying BILSTM (in the task of separating toxic from non-toxic posts) improved from 90.9% to 94.2%, and the $F1$ score of BILSTM+ARE (in toxic spans detection) improved from 57.7% to 58.8%, almost reaching the performance of BILSTM-SEQ.[6]

7 Toxic spans in toxic-to-civil transfer

As shown in Section 6, a toxic span detection method can be used to highlight toxic parts of a post, to assist, for instance, human moderators. The new TOXICSPANS dataset and toxic span detection methods, however, can assist in more ways. This section describes how we combined the new dataset and the best-performing toxic span detector (SPAN-BERT-SEQ) to show how they can be useful in *toxic-to-civil text transfer* (Nogueira dos Santos et al., 2018; Laugier et al., 2021). In the context of detoxifying comments to nudge healthier conversations online, this task aims at suggesting civil rephrasings of toxic posts. More specifically, we study the following research question: "Can TOXICSPANS data and toxic span detectors be used to assess the mitigation of *explicit toxicity* in toxic-to-civil transfer?" To answer this question, we proceeded in two ways: (i) evaluating the transfer of toxic spans in *system*-detoxified posts, and (ii) studying any remaining toxic spans in *human*-detoxified posts.

[6] Appendix A reports results for less added data.

Dataset	**Parallel (P)**	**Non-Parallel (NP)**	
Attribute	Toxic-to-Civil pairs	Toxic	Civil
Train	2,222	90,293	5,653,785
Dev	278	4,825	308,130
Test	278	4,878	305,267
Av. len.	19.8 (toxic)	19.4	21.9

Table 5: Statistics for the parallel (P) and non-parallel (NP) datasets, used to train the SED-T5 and CAE-T5 toxic-to-civil models, respectively. Average lengths are reported by counting SentencePiece (Kudo and Richardson, 2018) tokens.

7.1 System-detoxified posts

We first compare the performance of two toxic-to-civil transfer models, CAE-T5 and SED-T5, both based on the T5 transformer encoder-decoder architecture (Raffel et al., 2019); they both fine-tune the weights of the same pre-trained model, namely T5-large. CAE-T5 (Laugier et al., 2021) is a self-supervised Conditional Auto-Encoder, fine-tuned on a large non-parallel (NP) dataset based on pre-processed posts from the Civil Comments (CC) dataset, the dataset (with post level annotations) that TOXICSPANS was also based on. SED-T5 is a Supervised Encoder-Decoder. We fine-tuned it on a smaller parallel (P) dataset created by Dementieva et al. (2021), consisting of pairs of comments: a toxic comment and a detoxified paraphrase written by a crowd-worker.

Table 5 summarizes statistics of the two datasets (P, NP) and highlights a trade-off between the level of supervision and number of samples: there is a 1:40 ratio between toxic comments in P (direct supervision, parallel data) and NP (indirect supervision, no parallel data). Table 6 shows our exper-

imental results. Following Laugier et al. (2021), we report accuracy (ACC), perplexity (PPL), similarity (SIM) and the geometric mean (GM) of ACC, 1/PPL, SIM. Accuracy measures the rate of successful transfers from toxic to civil, and computes the fraction of posts whose civil version is classified as non-toxic by a BERT toxicity classifier.[7] Perplexity is used here as a measure of fluency and it is computed with GPT-2 (Radford et al., 2019). Similarity measures content preservation between the original toxic text and its system-rephrased civil version (self-SIM) or between the original toxic text and its gold (human) civil rephrasing (ref-SIM, only for P); in both cases, it is computed as the cosine similarity between the single-vector representations of the two texts, produced by the universal sentence encoder (Cer et al., 2018).

As can be seen in Table 6, CAE-T5 has better aggregated results (higher GM) than SED-T5 in all three datasets, which are due to lower perplexity and (in NP and TOXICSPANS) higher accuracy. However, SED-T5 learned to preserve content better (higher SIM in all three datasets), because of the parallel data (P, with gold rephrases) it was trained on. By contrast, CAE-T5 was trained without parallel data (NP) using a cycle-consistency loss, which leads to more frequent hallucinations of content that was not present in the original post (Laugier et al., 2021). These hallucinations may also help CAE-T5 obtain better perplexity scores, by generating fluent civil 'rephrases' that do not preserve, however, the original semantics. Also, although the general trends are similar in all three datasets (SED-T5 preserves content better, CAE-T5 is better in perplexity and GM), there are several differences too across the three datasets. For example, CAE-T5 is much better than SED-T5 in accuracy (posts detoxified) on NP and TOXICSPANS, but both systems have the same accuracy on P; and the scores of the systems vary a lot across the three datasets.

These considerations motivated us to seek ways to further analyse the behavior of toxic-to-civil transfer models. TOXICSPANS and toxic span detectors are an opportunity to move towards this direction, by studying how well transfer models cope with *explicit toxicity*, i.e., spans that can be explicitly pointed to as sources of toxicity. We leave for future work the flip side of this study, i.e., studying cases where transfer models rephrase spans not explicitly marked (by toxic span detectors or human annotators) as explicitly toxic.

[7] We reused the BERT model of Laugier et al. (2021).

Evaluation Dataset	Metric	CAE-T5	SED-T5
Non-Parallel (NP)	ACC ↑	**75.0%**	52.2%
	ACC2 ↑	**83.4%**	67.3%
	PPL ↓	**5.2**	11.8
	self-SIM ↑	70.0%	**87.9%**
	GM (self) ↑	**0.466**	0.338
	ACC3 ↑	**86.7%**	64.1%
	ACC4 ↑	**83.2%**	59.5%
Parallel (P)	ACC ↑	94.3%	94.3%
	ACC2 ↑	**94.7%**	94.3%
	PPL ↓	**9.1**	38.3
	ref-SIM ↑	27.6 %	**65.3%**
	self-SIM ↑	32.6 %	**65.6%**
	GM (ref) ↑	**0.306**	0.252
	GM (self) ↑	**0.323**	0.252
	ACC3 ↑	**98.8%**	94.3%
	ACC4 ↑	**94.7%**	91.9%
TOXICSPANS	ACC ↑	**92.9%**	65.6%
	ACC2 ↑	**92.5%**	63.7%
	PPL ↓	**7.2**	24.9
	self-SIM ↑	34.5%	**82.1%**
	GM (self) ↑	**0.355**	0.279
	ACC3 ↑	**96.9%**	62.0%
	ACC4 ↑	**92.0%**	54.7%

Table 6: Automatic evaluation scores of CAE-T5 (trained on NP's training subset) and SED-T5 (trained on P's training subset), when the test sets are from NP, P, and TOXICSPANS. ACC2, ACC3, ACC4 also consider toxic spans (Section 7.2).

7.2 Explicit Toxicity Removal Accuracy

Recall that the accuracy (ACC) scores of Table 6 measure the percentage of toxic posts that the transfer models (CAE-T5, SED-T5) rephrased to forms that a (BERT-based) toxicity classifier considered non-toxic. One could question, however, if it is possible (even for humans) to produce a civil rephrase of a toxic post when it is impossible to point to particular spans of the post that cause its toxicity (as in the last post of Table 1). Detoxifying posts of this kind may constitute a mission impossible for most models (possibly even for humans); the only way to produce a non-toxic 'rephrase' may be to change the original post beyond recognition, which may be rewarding systems like CAE-T5 that often hallucinate in their rephrases, as already discussed.

Hence, it makes sense to focus on posts that contain explicit toxic spans, marked by human annotators (for TOXICSPANS) or our best toxic span detector (SPAN-BERT-SEQ). Using these toxic spans, we define three additional variants of accuracy: ACC2 is the same as ACC, but ignores (in its denominator) posts that do not contain at least one toxic span; ACC3 also considers (in its denominator) only posts that contained at least one toxic span, but computes the fraction of these posts that had all of their toxic spans rephrased (even partly) by the transfer model;

ACC4 is a stricter version of ACC3 that requires the posts to also be judged non-toxic by the (BERT-based) toxicity classifier.

Table 6 shows that restricting ACC to consider only posts with at least one toxic post (ACC2) substantially improves the performance of both models on the NP dataset, indicating that it contains many 'mission impossible' instances (posts with no toxic spans) that the original ACC considers. By contrast, switching from ACC to ACC2 leads to mostly negligible changes on the P and TOXICSPANS datasets, which is in accordance with the fact that they contain fewer posts with no toxic spans (11.5% and 48.7%, respectively, compared to 67.4% for NP). Another interesting observation is that ACC4 is always substantially lower than ACC3 (for both systems, on all three datasets), indicating that the models often successfully detect toxic spans and try to rephrase them, but the rephrases are still toxic, at least according to the toxicity classifier.

7.3 Human-detoxified posts

In this experiment, we wished to study the extent to which *humans* rephrase known toxic spans, when asked to produce civil rephrases of toxic posts. We used the P dataset, the only one of the three considered that contains human rephrases.[8] Since P does not contain gold toxic spans, we again employed SPAN-BERT-SEQ to add toxic spans to the source posts and retained only the 1,354 (out of 2,778 in total) source-target pairs of posts with at least one toxic span in their source post.[9] In all but 6 of the 1,354 posts, the humans have rephrased (in the gold target post they provided) all the toxic spans of the source post. The 6 posts were mainly cases where the human changed the context to mitigate toxicity, while retaining the original toxic span. For example, "*he's not that* ***stupid***" became "*he's not* ***stupid***" (original toxic span shown in bold); in this case removing the 'that' from the context arguably makes the post less offensive. Overall, we conclude that humans did rephrase almost all cases of explicit toxicity in the toxic posts they were given.

We also applied SPAN-BERT-SEQ to the gold target (rephrased) posts that the humans provided to check if any explicit toxicity remained or was introduced by the rephrases. This flagged 93 gold target posts as comprising at least one toxic span. A manual inspection of the 93 posts revealed that they fall in two main categories. The first category comprises cases where a toxic span of the source post was rephrased, but the rephrase might not be considered totally civil; e.g., "*how* ***freaking narcissistic*** *do you have to be?*" became "*how* ***narcissistic*** *do you have to be?*", where SPAN-BERT-SEQ marked the 'narcissistic' of the rephrase as a toxic span. The second category comprises cases where SPAN-BERT-SEQ produced false positives; e.g., the source post "*most of the information is total* ***garbage***" became "*most of the information is totally* ***useless***", but SPAN-BERT-SEQ marked (arguably incorrectly) 'useless' as a toxic span.

8 Discussion

The posts we annotated for toxic spans were extracted from an already heavily studied public domain benchmark dataset (Civil Comments) that has been examined by thousands of teams in a Kaggle competition,[10] and that has been cited in over 50 academic publications. The Civil Comments dataset was filtered to remove any potential personally identifiable information before it was released. Our annotation cost was $21,089 for 59,486 judgements, paying $0.30 per item. All raters were warned for the explicit content of the job and only high accuracy raters were selected (70+%), based on performance on quiz questions. The most common countries of origin of our crowd-annotators were Venezuela and USA (Fig. 6 in Appendix A.1). In the contributor satisfaction survey, 51 participants gave an overall task rating of 3.6/5.0, with pay and test question fairness rated slightly higher than ease of job and clarity of instructions.

We note that it is more difficult and costly (approximately 3 times more) to manually annotate toxic spans, instead of just labeling entire posts as toxic or not. This is why we also explored adding rationale extraction components on top of toxicity classifiers trained on existing much larger datasets. We showed that BILSTM+ARE has the potential to reach the performance of BILSTM-SEQ, which is important for future work aiming to build toxic span detectors without any toxic span annotations in the training data. This may be particularly useful in low-resourced languages with limited resources for text toxicity (Zampieri et al., 2020).

Having two separate systems, one for toxicity detection and one for toxic spans identification, is more easily compatible with existing deployed toxi-

[8] We used all the P data, since no training was involved.

[9] The most frequent spans were 'sh*t', 'st*p*d', 'f*ck'.

[10] shorturl.at/hqEJ3

city detectors. One can simply add a component for toxic spans at the end of a pipeline for toxicity detection, and the new component would be invoked only when toxicity would be detected, leaving the rest of the existing pipeline unchanged. Since the vast majority of posts in real-world applications is non-toxic (Borkan et al., 2019), this pipeline approach would only increase the computational load for the relatively few posts classified as toxic. Using only toxic posts in this study was also a way to simplify this first approach to toxic spans detection, assuming an oracle system achieved the first step (deciding which posts are toxic). However, we note that future work could study adding non-toxic posts to our dataset and requiring systems to first detect toxic posts, then extract toxic spans for toxic posts.

A direct comparison (in terms of size) of TOXICSPANS with other existing toxicity datasets is only possible if one focuses on the toxic class, typically the minority one, since our dataset contains only toxic posts. By adding non-toxic posts, much larger versions of our dataset can be compiled, of sizes similar to those of existing previous datasets (that provide post-level annotations only). Hence, our TOXICSPANS dataset will be accessible with the following versions. First, only toxic posts included (11,006 posts), which is the version we discuss in this work. Second, the previous version will be augmented with the same number of randomly selected non-toxic Civil Comments posts. Third, a version similar to the previous one, but where the ratio of toxic to non-toxic posts will be 1:40 to be closer to that of real-world datasets (325,499 posts).

As shown in Section 7, the TOXICSPANS dataset and toxic span detectors can also help study and evaluate explicit toxicity removal when rephrasing toxic posts to be civil. In this case, toxic spans can be used to get a better understanding of how toxic-to-civil models operate, by showing the toxic spans and their context, along with their rephrases.

9 Intended use and misuse potential

The toxic span detection systems we consider are trained (the sequence-labeling ones) and tested (all systems) on posts with binary ground-truth character offset labels (toxic or not), reflecting the majority opinion of the annotators (Section 3). This runs the risk of ignoring the opinions of minorities, who may also be minorities among crowd-annotators. To address this issue, we also release the toxic spans of all the annotators and the pseudonymous rater identities, not just the spans that reflect the majority opinion, to allow different label binarisation strategies and further studies.

Toxic span detection systems are intended to assist the decision making of moderators, not to replace moderators. When they operate correctly, systems of this kind are expected to ease decision making (reject/accept a post). Incorrect results could be of two types; toxic spans that were not highlighted and non-toxic spans that were highlighted. Mistakes of both types, especially of the first one, may mislead a moderator working under pressure.

As with other content filtering systems (e.g., spam filters, phishing detectors), toxic span detectors may trigger an adversarial reaction of malicious users, who may study which types of toxic expressions evade the detectors (esp. publicly available ones) and may gradually start using more implicit toxic language (e.g., irony, false claims), which may be more difficult to detect. However, this is a danger that concerns any toxicity detection system, including systems that classify user content at the post level (without detecting toxic spans).

10 Conclusions and future work

We studied toxicity detection, which aims to identify the spans of a user post that make it toxic. Our work is the first of this kind in general toxicity detection. We constructed and release a dataset for the new task, along with baselines and models. Fine-tuning the SPAN-BERT sequence labelling model of Joshi et al. (2020), yielded the best results. A post-level BILSTM toxicity classifier that was combined with an attention-based attribution method, not trained on annotations at the span level, performed well for the task. By leveraging the dataset of posts annotated as toxic or non-toxic (without spans), we showed that this method can reach the performance of a BILSTM sequence labelling approach that was trained on the more costly toxic spans annotations. This result is particularly interesting for future work aiming to perform toxic spans detection by using only datasets with whole-post toxicity annotations. In a final experiment, we examined toxic-to-civil transfer, showing how toxic spans can help shed more light on this task too, by helping assess how well systems and humans address explicit toxicity. *All our code and data will be publicly available.* In future work we plan to study toxic span detection in multiple languages and in context-dependent toxic posts.

References

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 *WWW*, pages 491–500, San Francisco, USA.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *EMNLP-IJCNLP*, pages 5640–5650.

Daryna Dementieva, Sergey Ustyantsev, David Dale, Olga Kozlova, Nikita Semenov, Alexander Panchenko, and Varvara Logacheva. 2021. Crowdsourcing of parallel corpora: the case of style transfer for detoxification. In *Proceedings of the 2nd Crowd Science Workshop: Trust, Ethics, and Excellence in Crowdsourced Data Management at Scale co-located with 47th International Conference on Very Large Data Bases (VLDB 2021 (https://vldb.org/2021/))*, pages 35–49, Copenhagen, Denmark. CEUR Workshop Proceedings.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, Minneapolis, Minnesota.

Ashwin Geet D'Sa, Irina Illina, and Dominique Fohr. 2020. Towards non-toxic landscapes: Automatic toxic comment detection using DNN. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 21–25, Marseille, France. European Language Resources Association (ELRA).

Javier Ferrando and Marta R. Costa-jussà. 2021. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 434–443, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xiaochuang Han and Yulia Tsvetkov. 2020. Fortifying toxic speech detectors against veiled toxicity. In *EMNLP*, pages 7732–7739, Online.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. In *arXiv preprint*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.

Mladen Karan and Jan Šnajder. 2019. Preemptive toxic language detection in Wikipedia comments using thread-level context. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 129–134, Florence, Italy.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Léo Laugier, John Pavlopoulos, Jeffrey Sorensen, and Lucas Dixon. 2021. Civil rephrases of toxic texts with self-supervised transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1442–1461, Online. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A survey on computational propaganda detection. In *IJCAI*, pages 4826–4832.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Arxiv (accepted at AAAI)*.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short*

Papers), pages 189–194, Melbourne, Australia. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017a. Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35, Vancouver, BC, Canada. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017b. Deep learning for user comment moderation. In *Proceedings of the 1st Workshop on Abusive Language Online*, pages 25–35, Vancouver, Canada.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017c. Deeper attention to abusive user content moderation. In *EMNLP*, pages 1125–1135, Copenghagen, Denmark.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *ACL*, pages 4296–4305, Online.

John Pavlopoulos, Nithum Thain, Lucas Dixon, and Ion Androutsopoulos. 2019. Convai at semeval-2019 task 6: Offensive language identification and categorization with perspective and bert. In *SemEval*, Minneapolis, USA.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?" Explaining the predictions of any classifier. In *SIGKDD*, pages 1135–1144, San Francisco, USA.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *ACL*, pages 5477–5490, Online. Association for Computational Linguistics.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Betty Van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online*, pages 33–42, Brussels, Belgium.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *WWW*, pages 1391–1399, Perth, Australia.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *SemEval*.

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). *arXiv preprint arXiv:2006.07235*.

A Appendix

A.1 Exploratory analysis of TOXICSPANS

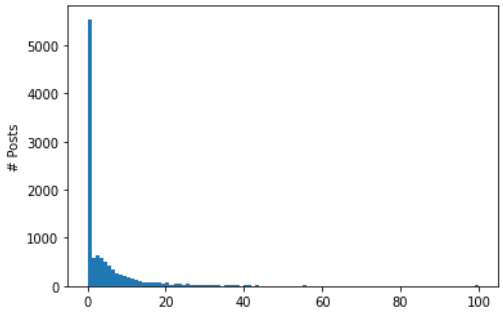

Figure 1: Distribution of the percentage of characters of each post that are covered by the ground truth spans.

Figure 1 shows the distribution of the percentage of character offsets of each post that are included in toxic spans. Figure 2 illustrates the distribution of dense toxic spans per post. Figure 3 shows the most frequent toxic spans in the dataset (after lower-casing each post) and their frequencies. Figure 4 shows the most frequent multi-word toxic spans (again after lower-casing). Figure 5 illustrates the distribution of the size (in words) of those posts whose ground truth covers the whole post. Figure 6 shows the frequencies of the countries of origin of the TOXICSPANS crowd-annotators.

A.2 Error analysis of SPAN-BERT-SEQ

We performed an error analysis on our best toxic spans detector (SPAN-BERT-SEQ). We analyzed its predictions on the first fold of the Monte Carlo Cross-Validation, which comprises 10% of the dataset or 1001 posts. We identified three main types of errors. The first, which is the most frequent one occurring in 235 out of 1001 posts (23.5%), comprises posts for which SPAN-BERT-SEQ failed to find all toxic spans. This type of error can be

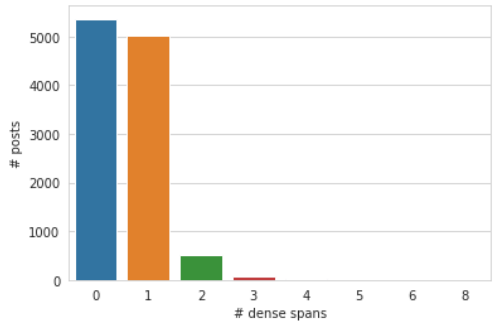

Figure 2: Distribution of the number of dense ground truth toxic spans per post in TOXICSPANS.

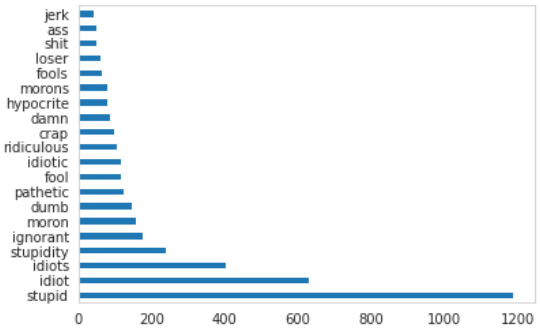

Figure 3: Most frequent toxic spans in TOXICSPANS.

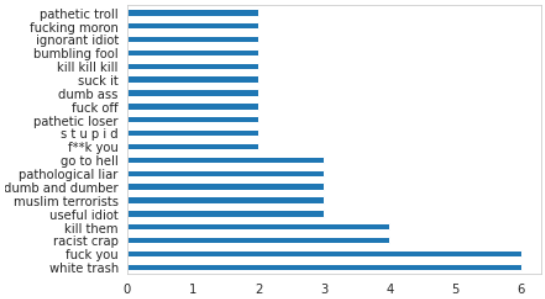

Figure 4: Most frequent multi-word toxic spans.

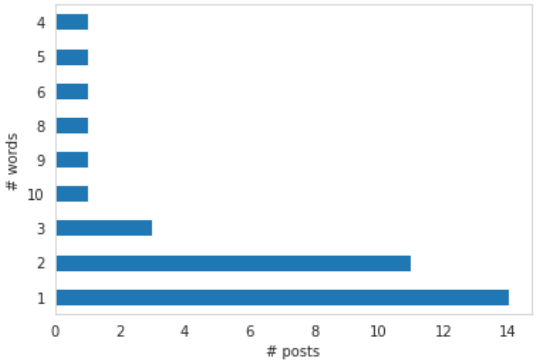

Figure 5: Distribution of size (in words) of posts whose ground truth covers the whole post.

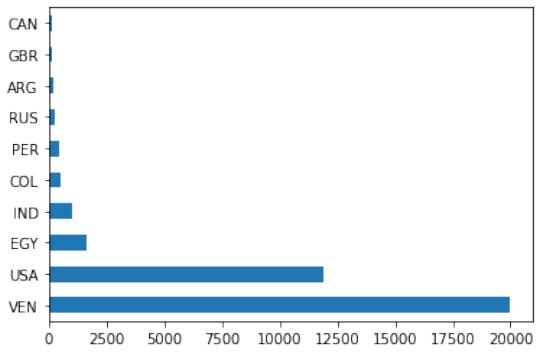

Figure 6: Frequency of annotations based on the country of origin of the crowd-annotators.

You can stick your d**k up anyone's butt.
Of course they do. Stupid people really have to meet everyone else half way if they don't want to be called stupid, starting with not saying stupid things.

Table 7: Examples posts where SPAN-BERT-SEQ incorrectly predicted no spans. Ground truth in red.

Play **stupid** games, win stupid prizes.
I always smile when I've been called **stupid** by a fool.

Table 8: Examples posts where SPAN-BERT-SEQ predicted some, but not all of the gold spans. Ground truth in red. Predictions of SPAN-BERT-SEQ in bold.

divided in two sub-types: the first sub-type comprises posts for which SPAN-BERT-SEQ predicted no spans at all (Table 7), while the second sub-type comprises posts for which SPAN-BERT-SEQ predicted some, but not all of the gold spans (Table 8). The first sub-type occurs more often, with 217 out of the 235 total occurrences of the first error type, while the second sub-type occurs only a few times (18 out of 235). The second type of error, which is the second most frequent one, occurred in 173 out of the 1001 posts (17.3%). It occurs when the ground truth of a post is empty, but SPAN-BERT-SEQ predicts at least one toxic span (Table 9). The last type of error occurs rarely (only 10 out of 1001 posts) when the ground truth of a post is not empty, and SPAN-BERT-SEQ predicts more (or larger) toxic spans than it should (Table 10).

A.3 Experimental Settings

Sequence labelling

BILSTM-SEQ was implemented in KEARS, version 2.7.0.[11] We used word embeddings of size 200 and hidden states of size 128; mean squared error (MSE) loss; the Adam optimiser; learning rate 0.001; post padding; maxlen and batch size 128; training for max. 100 epochs. We used early stopping with 5 epoch patience, monitoring the validation loss. The classification threshold was set to 0.5. CNN-SEQ was trained for 30 epochs; we used 0.5 recurrent dropout; progressively increasing batch size from 4 to 32 with step 1. All the other hyper-parameters were set to their default values. BERT-SEQ was implemented using the huggingface transformers library.[12] We used the bert-base-cased model, binary cross entropy loss; the Adam optimiser; learning rate $2 \cdot 10^{-5}$; maxlen 128; batch size 32; training for max. 100 epochs; early stopping with 5 epoch patience, monitoring validation loss. The classification threshold was 0.5.

[11] https://keras.io/

[12] https://huggingface.co/transformers/

This outlet should hire some editors. Nobody I've crossed paths with would green light this **crap**.
Actually, Seaton is a wealthy man and can do without his day job quite easily. If he would just get rid of that friggin' **stupid** cap....
In other word, blah, blah, blah, blah. It's **bullshit**. Deal with it. No proof=doesn't exist.
Or maybe we should place a tax on **stupid** ideas like yours

Table 9: Examples posts where the ground truth was empty, but SPAN-BERT-SEQ incorrectly predicted at least one span. Predictions of SPAN-BERT-SEQ in bold.

People don't normally take it to heart when an **idiot calls someone stupid**.
$10B a GW avg compared to $2.5B a GW for a 2nd Candu nuke at LePreau. **Stupid is as stupid** does I guess.
All **useless piles of crap**.
oh no, this isn't even in the top 10 **moronic statements by this babbling fool**.

Table 10: Examples posts where the ground truth was not empty, and SPAN-BERT-SEQ incorrectly predicted more (or larger) toxic spans. Ground truth in red. Predictions of SPAN-BERT-SEQ in bold.

SPAN-BERT base (cased) was fine-tuned in the same way that Joshi et al. (2020) fine-tunes it on SQUAD 2.0 (Rajpurkar et al., 2018) with the format mapping presented in Table 11. At training time, we ignore posts with more than one dense toxic span, since the SQUAD 2.0 format allows for only one dense answer span in the context. We trained with a learning rate $2 \cdot 10^{-5}$, for 4 epochs with training batches of size 32.

Post-level classifiers with attribution

BILSTM+ARE was implemented in KERAS, like BILSTM-SEQ. We used maxlen of 128; post padding; early stopping with patience 5 epoch, monitoring the validation loss; Adam optimizer with 0.001 learning rate; MSE loss. The text classification threshold was 0.5. BERT+ARE was implemented with Huggingface Transformers similarly to BERT-SEQ. We used maxlen of 128; post padding; early stopping with patience 5 epoch,

SQUAD 2.0	TOXICSPANS
Context	Post
Question	Empty string
is_impossible boolean	toxic_spans_is_empty boolean
Answer	Toxic span

Table 11: Mapping between the SQUAD 2.0 format and TOXICSPANS examples.

monitoring the validation loss; Adam optimizer with $2 \cdot 10^{-5}$ learning rate; binary cross-entropy loss. The text classification threshold was 0.5. In both models, the attention threshold (above which a token is considered toxic) was fine-tuned on the development set of each Monte Carlo C-V fold.

Further implementation details can be found in our code repository, which will be made publicly available in the camera-ready version of this paper.

A.4 Improving BILSTM+ARE with more training of the underlying BILSTM

Figure 7 shows the improvement in the F1 score of BILSTM+ARE when increasing the training set of the underlying BILSTM with 5k, 10k, 20k, 40k, 80k more posts (always balanced toxic and non-toxic) with post-level annotations only (no toxic span annotations). The dashed lines represent the sequence labeling methods, which cannot benefit directly from training data without toxic span annotations. Similarly, Fig. 8 shows the corresponding improvement in the ROC AUC score of the underlying BILSTM in the toxic/non-toxic text classification task.

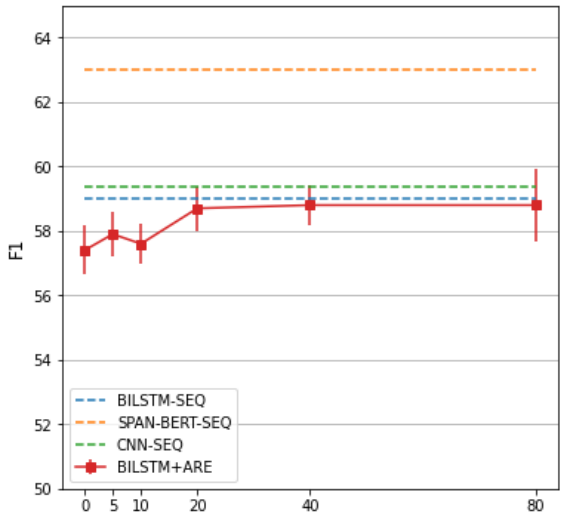

Figure 7: Improvement in the F1 of BILSTM+ARE when increasing the training set of its underlying BILSTM with posts tagged at the post-level (toxic/non-toxic, no toxic spans). Standard error of mean shown as error bars.

A.5 Toxicity scores of posts with and without explicit toxicity

We applied the BERT-based text toxicity classifier (Laugier et al., 2021), which we also used in Section 7, to the 2,778 posts of the P dataset, dividing them in two sets: posts that comprised at least one toxic span detected by SPAN-BERT-SEQ (1,354 posts with explicit toxicity) and the rest (implicit toxicity). The BERT-based toxicity classifier considered more toxic (higher average toxicity score) the 1,354 posts of the first set compared to the second one, i.e., it was more confident that the posts of the first set (explicit toxicity) were toxic, as one might expect. By resampling 1,000 subsets (of 50 posts each) from the two sets, we confirmed that this is a statistically significant difference ($P = 0.001$). The difference of the average predicted toxicity score between the two sets is 14% (from 0.94 down to 0.80).

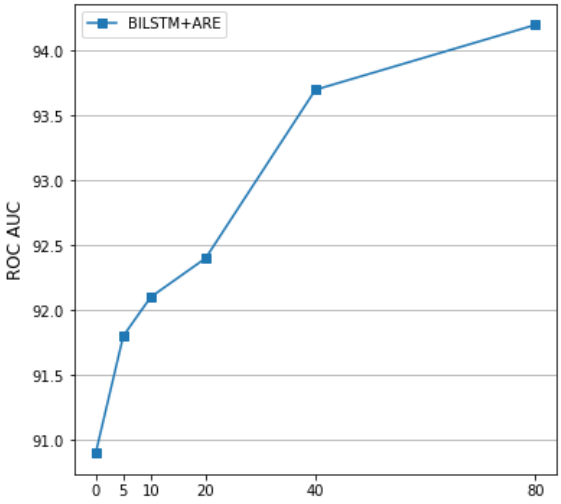

Figure 8: Improvement in the ROC AUC of BILSTM+ARE in the toxic spans detection task, when increasing the training set of its underlying BILSTM with posts tagged at the post-level (no toxic spans).