SRPOL DIALOGUE SYSTEMS at SemEval-2021 Task 5: Automatic Generation of Training Data for Toxic Spans Detection

Michał Satława, Katarzyna Zamłyńska, Jarosław Piersa,
Joanna Kolis, Klaudia Firląg, Katarzyna Beksa.
Zuzanna Bordzicka, Christian Goltz, Paweł Bujnowski
Samsung R&D Institute Poland
{m.satlawa;k.zamlynska;j.piersa}@samsung.com
joanna.kolis@gmail.com, {k.firlag;k.beksa}@samsung.com
{z.bordzicka;c.goltz;p.bujnowski}@samsung.com

Piotr Andruszkiewicz
Samsung R&D Institute Poland
Warsaw University of Technology
p.andruszki2@samsung.com

Abstract

This paper presents a system used for SemEval-2021 Task 5: Toxic Spans Detection. Our system is an ensemble of BERT-based models for binary word classification, trained on a dataset extended by toxic comments modified and generated by two language models. For the toxic word classification, the prediction threshold value was optimized separately for every comment, in order to maximize the expected F1 value.

1 Introduction

Freedom of speech is one of the most important human rights. However, because the definition of protected speech is not precise enough, it can be easily misinterpreted and misused. The problem is magnified in cyberspace, where anonymity and asynchronous communication contribute to toxic disinhibition. As a result, the Internet has become space where hatred, harsh criticism, rude language and threats may grow.

Currently, identification of such harmful content may depend mostly upon classification models that detect abusive comments or documents. However, SemEval-2021 Task 5: Toxic Spans Detection proposes detecting fragments of text that make it toxic, with the aim of supporting manual moderation of oftentimes lengthy comments. A successful solution to this problem would be a crucial step towards more constructive and inclusive online discussions. The task focuses on English, which is the most common language used on the Internet, as of January 2020 (Johnson, 2021).

In this paper we present the model we used for toxic span detection, the method we used to find optimal prediction threshold values, and two methods for producing new training examples with toxic spans annotation:

- Resampling the data: new examples are generated by substituting non-toxic words with predictions from a language model.
- Data generation: we trained a simple language model on existing examples, in order to generate new examples containing marked toxic spans.

A total of 91 teams made an official submission on the test set with the best submission achieving F1 score of 0.7083. Our approach was ranked as 11th with 0.6865 F1 score (for details see Section A of the Appendix).

2 Related Work

The interest in automatic identification of abusive language has increased among researchers due to the importance of public discussion in the Internet and its public impact. To our mind, the domain overlaps with other NLP tasks, such as sentiment analysis or comment classification.

In an earlier piece of work, (Yin et al., 2009) studied harassment detection on Web 2.0 datasets (i.e. Kongregate, Slashdot and MySpace) using TFIDF n-gram features and SVM models. In other research, (Sood et al., 2012a,b) analysed profanity detection in a community-based news site Yahoo!

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 974–983
Bangkok, Thailand (online), August 5–6, 2021.

Buzz with SVM and Levenshtein distance. In later studies, (Wulczyn et al., 2017) tried to understand personal attacks in English Wikipedia discussion comments using logistic regression and multilayer perceptron classifiers on character and word n-grams.

In the last years there were several contests focusing on various aspects of offensive language. One of them included hate speech (Bosco et al., 2018) and misogyny identification (Fersini et al., 2018) for Italian data. Another workshop concerning a similar topic was Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter at SemEval-2019 (Garibo i Orts, 2019). Recently, research has focused mostly on deep learning classification of cyber hate using "accept"/"non-offensive" or "reject"/"offensive" classes, e.g. (Pavlopoulos et al., 2017). However, more and more papers concern explainability components and reasons. For example, organizers of SemEval-2019 Task 6, (Zampieri et al., 2019) required identification of the offensive content type and the target of the offensive post. In another study, carried out by (Mathew et al., 2020), aside from simple performance metrics, more complex bias and explainability factors were used to evaluate various deep neural networks models (CNN-GRU, BiRNNs and BERT). A similar idea – to clarify rationales for hate speech – comes in SemEval-2021 Task 5: Toxic Spans Detection (Pavlopoulos et al., 2021). Our solutions to this challenge are presented in this paper.

3 Data

There are several abusive language detection datasets available (such as (Wulczyn et al., 2017; Borkan et al., 2019)). However, their purpose is toxicity detection of the whole text, so they do not contain information about the exact spans that make a text toxic. For such a task, the SemEval-2021 Toxic Spans Detection dataset was created. It consists of 10,629 English texts and their relative lists of toxic spans' indices (7,939 train, 690 trial, and 2,000 test texts with their respective spans). Examples from the dataset are presented in Table 1.

The SemEval-2021 Task 5 annotated data turned out to be a challenging material to work on in some respects.

It results mainly from somewhat inconsistent annotation which is visible in several dimensions, see Appendix B.

4 System Overview

4.1 Resampled Data

The texts from train dataset were tokenized and the tokens that were not labeled as toxic spans were replaced with tokens suggested by a RoBERTa (Liu et al., 2019) language model. We set the limit of substituted words on 1/3 of all tokens outside the toxic span, but no more than 10. The replacement was not applied to the punctuation. Also, it was not possible to change the word to the same one, or to the word preceeding the token in question. The tokens to be substituted are chosen randomly so it was possible to obtain different outcomes while processing the same text several times:

original text:

*Uh, yes he is **stupid**...as his actions have proven on countless occasions...*

generated:

*Uh, Donald Trump 's **stupid**...as recent history has shown in numerous occasions...*

*Uh, now that 's **stupid**...as my students proudly demonstrated in numerous occasions...*

*Uh, well that 's **stupid**...as Vikings fans themselves said in numerous occasions...*

4.2 Generated Data

The second method of expanding the original training dataset was inspired by the Ding et al. (2020) data augmentation method for the POS tagging. Firstly, the pairs of spans indices and texts from training data were encoded in linearized form, e.g. the original span and text pair:

*"[9, 10, 11, 12, 13]", "One less **idiot** to worry about."*

was transformed into text

*"**<s>** One less **<toxic>** idiot **</toxic>** to worry about. **</s>**"*

where <toxic>, </toxic> are special tokens which indicate the beginning and ending of a toxic span and <s>, </s> are special tokens indicating that this particular example is annotated. The set of such linearized training examples was extended by a dataset of the same size, containing unannotated examples from Civil Comment Dataset (Borkan et al., 2019) (with a different special starting token **<u>**). These examples were used to train a simple language model with 300-dimensional word embedding and two 256-dimensional LSTM layers. This language model was used to generate new sentences in linearized format, starting from the token **<s>** and randomly sampling next words from the

Text	Toxic span
What a pile of shit. I love Bruce and I could write a better case against Bruce than this rubbish !	[15, 16, 17, 18, 90, 91, 92, 93, 94, 95, 96]
What a jerk!	[7, 8, 9, 10]
Deep ecology madness is a sad sick religion.	[]

Table 1: SemEval-2021 Task 5 dataset examples.

model distribution until the ending token **</s>** is sampled or the generated text length exceeds 200 tokens. 10 000 of such generated examples were converted back into the format span + text and used for training the token classifier model.

4.3 Token Classification

To detect toxic spans within a text we use the Hugging Face (Wolf et al., 2019) implementation of the BERT model (Devlin et al., 2018). We approach the task as token classification. Therefore, there are additional layers on top of the model – a classification head (a one-dimensional linear layer on top of the hidden-states output), preceded by a 50% dropout layer.

4.4 Character Classification

The metric used for competition ranking was the mean value of character-level F1 score, as in (Da San Martino et al., 2019), so the token-level predicted probabilities from our model needed to be converted to character-level binary labels. The process included two stages.

- Assigning probabilities to text characters. Every character that is a part of a token is assigned the predicted probability of that token. All the other characters (e.g. whitespaces) are assigned 0.
- Choosing and applying the optimal threshold value for a given text example, based on the predicted characters probabilities. Characters with probabilities meeting this threshold are identified as being part of a toxic span.

The threshold value was optimized separately for every text example to maximize the expected value of character-level F1 score, where the F1 value for a predicted span is a random variable with respect to the distribution of golden spans.

Formally, for a given sentence composed of n letters, lets denote predicted labels (0 and 1) as $s = (s_1, ..., s_n)$ and golden labels as $y = (y_1, ..., y_n)$. Then the F1 measure for such prediction is

$$F_1(s, y) = \frac{2\sum_{i=1}^{n} s_i y_i}{\sum_{i=1}^{n} s_i + \sum_{i=1}^{n} y_i}$$

If elements of s and y are indexed in order of decreasing probabilities of beeing toxic, and $S_{i:j} = \sum_{k=i}^{j} y_k$, then the expected value of F_1 for the prediction with k most probable letters classified as toxic, is

$$f(k) = \sum_{k_1=0}^{k} \sum_{k_2=0}^{n-k} \frac{2P(S_{1:k}=k_1)P(S_{k+1:n}=k_2)k_1}{k + k_1 + k_2}$$

We approximated the distribution of the golden spans y by assuming that probabilities of characters in the golden toxic span are independent and equal to probabilities predicted by the model. This allowed us to use $O(n^2)$ algorithm proposed by (Nan et al., 2012) to find the k which maximizes the $f(k)$.

4.5 Ensemble Models

On top of the plain prediction models we applied a selection of ensembles (Opitz and Maclin, 1999). Ensembles are aggregations of models solving the same problem, built with the hope that a panel of experts can give a better decision than a single one. In our case, the ensemble accepted character-level inputs from 2 to 9 participating models and returned a single character-level output. Among the available range, we used:

- set-theory union (i.e. at least one model declared a character as toxic),
- set-theory intersection (i.e. all models declared a character as toxic),
- majority voting (i.e. at least half of the models declared a character as toxic),
- F1-weighted voting (i.e. the vote was weighted with the model's F1 score and computed for the evaluation set),

The output of the ensemble still needed to be post-processed.

5 Experimental Setup

The officially released datasets (both train and trial) contained altogether 8,629 texts. From those datasets we generated nine random train/dev/test splits, with the ratio 80/10/10%.

From the train sets in each of the splits new data was generated using both methods described in 4.1: resampling data outside the span, and data augmentation. For resampling data, we used the BERT based model (uncased) model with BertTokenizerFast tokenizer or RoBERTa base model with RobertaFastTokenizer. The upper limit of changes in the text was equal to 10.

We trained the token classification model using only the train set, as well as the train set together with one or two sets of generated data. We used binary cross-entropy loss function and early stopping technique, ending the training after 3 consecutive epochs without the decrease of loss function on the dev set.

The other hyperparameters used in the training were: batch size: 8, dropout rate: 0.5, learning rate: 1e-05, and max token sequence length: 340. We wanted to compare models trained on dataset with diverse sizes and, due to resampling, a lot of similar examples. To keep the evaluations for the early stopping in short and uniform intervals across all models we set a fixed number of steps per epoch: 800.

All the nine models obtained from given cross-validation splits were used for ensemble models. The best results were obtained via intersection and majority voting over models trained on cross-validation splits.

The final stage was postprocessing the spans. Whitespaces and punctuation characters that were the only characters separating two parts of toxic spans were included in the toxic span, while all the other whitespaces and punctuation characters located at the ends of spans were removed.

6 Results

The results of the models submitted to SemEval-2021 competition are presented in Table 2. All of the three models were trained on the official dataset and on data generated with resampling method. The results shows that adding more components of our system improves the final result. The first model is only token classifier model trained on enlarged dataset. When it comes to the second one, we used models trained on 9 different train/dev splits and aggregated their results using ensemble. The last one was improved by the threshold optimization and additional data generated with language model.

We checked the results of token classifier trained with data obtain with the other tokenizers available, see Table 3. Our best result was obtained for XLMRobertaTokenizerFast and it exceeded our best result in the competition.

The results of the models trained on augmented datasets, either with or without optimization of the prediction threshold, are presented in Table 4. It can be observed that adding more noisy and automatically generated data to the training worsened the model results, making them more variable. The addition of the threshold that optimized the expected value of F1 metric independently on every test set example fixed both issues, producing models with higher and more stable results.

7 Conclusions

In many classification tasks we can observe a divergence of the objective function (e.g. F1 score), optimized loss function (e.g. cross-entropy) and applied prediction thresholds. Our results demonstrate that even when the distribution of golden classes is crudely approximated by the assumption of independent and underperforming underlying model, the F1-optimized threshold values perform better than commonly used and accuracy-optimized threshold of 0.5 in the setting with noisy and automatically augmented training data.

The implemented method of data augmentations, based on resampling non-toxic words proved to be effective by increasing the F1 score of token classifier.

Model	data augumentation	F1 on test data
BERT	resampled	0.6826
intersection ensemble over cv splits	resampled	0.6847
voting ensemble over cv splits, threshold optimization	resampled, generated	**0.6865**

Table 2: Results of our top models submitted to competition.

language model	tokenizer	F1 on trial set	F1 on test set
xlm-roberta-base	XLMRobertaTokenizerFast	0.6732	**0.6910**
facebook/bart-base	BartTokenizerFast	0.6610	0.6832
bert-base-uncased	BertTokenizerFast	0.6999	0.6684
bert-large-uncased	BertTokenizerFast	0.7007	0.6700
google/electra-large-generator	ElectraTokenizerFast	0.6984	0.6735

Table 3: Results for different tokenizers used in resampled data generation. The names of models and tokenizers are taken from https://huggingface.co/models .

data augumentation	prediction threshold	F1 mean	F1 std. deviation
resampled, generated	F1-optimized	**0.6700**	0.0115
	0.5	0.6643	0.0128
resampled	F1-optimized	0.6670	**0.0093**
	0.5	0.6644	0.0148
generated	F1-optimized	0.6643	0.0110
	0.5	0.6666	0.0105
none	F1-optimized	0.6664	0.0121
	0.5	0.6688	0.0107

Table 4: The results obtained using cross validation split and voting ensemble for different datasets. The token classifier was trained with F1-optimized threshold as well as fixed treshold.

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. *CoRR*, abs/1903.04561.

Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. 2018. *Overview of the EVALITA 2018 Hate Speech Detection Task*, pages 67–74. Torino: Accademia University Press.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. 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 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. Association for Computational Linguistics.

E. Fersini, P. Rosso, and M. Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) colocated with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain*, pages 214–228.

Joseph Johnson. 2021. Most common languages used on the internet as of january 2020, by share of internet users. https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/. Accessed: 2021-02-23.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Man-

dar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *CoRR*, abs/2012.10289.

Ye Nan, Kian Chai, Wee Lee, and Hai Leong Chieu. 2012. Optimizing f-measure: A tale of two approaches. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 1:1–8.

D. Opitz and R. Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169—-198.

Òscar Garibo i Orts. 2019. Multilingual detection of hate speech against immigrants and women in Twitter at SemEval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 460–463, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection (to appear). In *Proceedings of the 15th International Workshop on Semantic Evaluation*.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

S. Sood, J. Antin, and E. Churchill. 2012a. Profanity use in online communities. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1481–1490.

S. Sood, J. Antin, and E. Churchill. 2012b. Using crowdsourcing to improve profanity detection. In *Wisdom of the Crowd - Papers from the AAAI Spring Symposium*, AAAI Spring Symposium - Technical Report, pages 69–74. 2012 AAAI Spring Symposium ; Conference date: 26-03-2012 Through 28-03-2012.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian Davison, April Edwards, and Lynne Edwards. 2009. Detection of harassment on web 2.0. In *Proceedings of the Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009*, pages 1–7.

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

A Our models results within SemEval-2021 Task 5 scores

Figure 1 presents the outcome of SemEval-2021 Task 5 contest until the deadline and the green dot shows our best result at the time of publication.

B Original Dataset Inconsistency

Firstly, there are words annotated as toxic in some comments, while in the other ones they are left out, despite the similar context of the utterance. The words “stupidity" or “crooked" can serve as an examples, sometimes being omitted, sometimes being treated as full toxic spans and finally, sometimes being treated as parts of toxic spans, together with their modifiers (see Table 5).

Another issue related to inconsistency is the length of the annotated span. The majority of spans consist of one to three words, but there are also cases in which spans are longer, containing not just a toxic word, but also a longer phrase including the toxic word (see Table 6). As previously, in our opinion the discrepancies are not justified by context.

Other issues we found peculiar in the provided annotation include annotating non-toxic words while omitting toxic ones (see Table 7) and beginning or ending the annotation in the middle of a word (Table 8). Such cases do not appear as often as the aforementioned discrepancies, but are also present.

Everything mentioned above might have been introduced to the dataset on purpose, as noise, in order to make the task more challenging to the models. However, we found the scale of the inconsistencies particular, the more so as it was not mentioned in the instructions for contest participants.

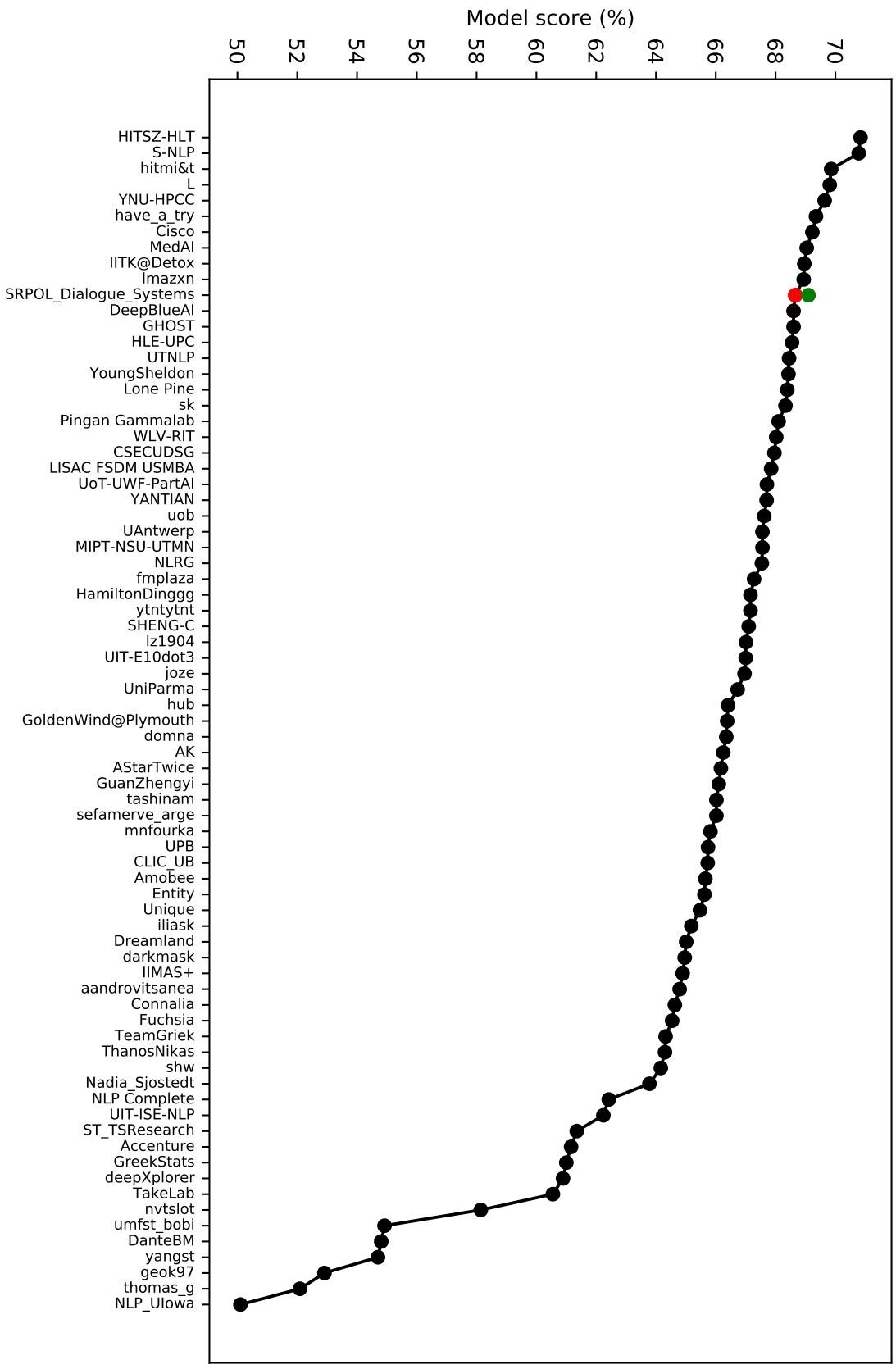

Figure 1: SemEval-2021 Task 5 results for models with scores $\geq 50\%$ on the test dataset. The red dot points our model outcome until the deadline (F1=68.65%) and the green dot shows our best result at the time of publication (F1=69.09%).

Comment text (annotated spans in bold italics)
we dont need to move from anything other than left wing stupidity like yours (no annotation)
keep living in your left wing ***stupidity***, there will be a market for our OIL from the SANDS for generations
oh more ***left wing stupidity*** ... colleges need to focus on educating and leave policy to the grown ups, like trump who is not banning anyone but suspending things
Unfortunately these people refuse to stand up to oppose the crooked self and friends rulers who don't hesitate to kill to kill them. (no annotation)
They were just ***crooked*** liars looking out for their own interests. Trump is way beyond that.
Now we will have endless ***crooked honkey*** lawyers talking unsuspecting "useful ***idiots***" into filing outrageous claims.

Table 5: Examples of inconsistent annotation.

Comment text (annotated spans in bold italics)
I am watching Goodale on TV giving statements about this. ***He is such an ass***. (17)
Saw him on the news. What an arrogant ***arse***! (4)
Trump's a liar (14); ***Trump's lawyer is a liar*** (23). Get them outta there! Lock them up!
There is no indication that Trump will change from the ***racist*** (6), ***sexist*** (6), ***demagogue*** (9) that he is.
The dotard almost forgot to sign it until his minder pulled him back (68).
Who gives a flying freak about what Kim and Park are making or not making? Really? This is what the SA chooses to write editorials about (136)?
Trump simply does not care. "I am president, and to ***hell*** with all of you".

Table 6: Examples of inconsistency in regard to span length. The numbers in brackets indicate the number of characters within a given toxic span.

Comment text (annotated spans in bold italics)
I am totally with you on this continued stupidity that has been US policy, and now Hawaiis effort. I was a military officers wife expecting my second son, stationed in Germany at this time. As I recall, Kennedy gave Kruschev an opportunity to " save face". I doubt that Trump has the intelligence or skill, nor has he picked an intelligent UN rep. Or cabinet. This is where investigative journalism makes the difference. I am glad that Civil Beat is ***challenging*** the status quo mentality.
Patriotism: the last refuge of scoundrels..
Is this important news? Is it even ***news***? Who needs to know this ***crap***?
Yeah John, you tell all those ***Chinese*** and ***Muslims*** you hate to get the hell out of Canada.

Table 7: Examples of annotation of non-toxic words while omitting potentially toxic ones.

Comment text (annotated spans in bold italics)
It's so weird to have an actual s***tupid narcissistic*** childish person in the white house.
Yep you ar***e a sore loser***
The hy***pocrisy*** of the christian right is astounding.

Table 8: Examples of annotations starting or ending in the middle of a word.

Original annotation	Our re-annotation
“Watch people die from taking away their healthcare" DING DING DING! Winner of ***stupid*** post of the day award!	***“Watch people die*** from taking away their healthcare" DING DING DING! ***Winner of stupid post of the day award***!
They’re not patriots. They’re vandals, ***thieves***, and ***bullies***. They’ve plastered a facade of patriotism over their outrage at being expected to obey the law.	They’re not patriots. ***They’re vandals, thieves, and bullies.*** They’ve plastered a facade of patriotism over their outrage at being expected to obey the law.
Rabidly ***anti-Canadian*** troll.	***Rabidly anti-Canadian troll.***

Table 9: Examples of differences between the original annotation and our re-annotation. The annotated spans are in bold italics.

C Re-annotation

Because of inconsistencies in the original annotation, we decided to re-annotate the trial and test datasets and compare the differences. For this task we invited a team of 14 language experts (all of them with a master’s degree in linguistics). They received the same instruction as the one provided by the contest organizers. Each text was evaluated by three language experts, as in the original dataset creation process.

The differences between the original annotation and the one provided by our language experts are quite noticeable. Firstly, we believe our annotation is more coherent and involves fewer mistakes described earlier. Moreover, our language experts marked fragments as toxic more often: whole sentences, paragraphs or even whole comments, as well as more single words (see Table 9). The goal of our re-annotation was to evaluate the quality of the original datasets annotation and check if our understanding of toxicity is equivalent to the one of contest organizers.