aschern at SemEval-2020 Task 11:
It Takes Three to Tango: RoBERTa, CRF, and Transfer Learning

Anton Chernyavskiy[1], **Dmitry Ilvovsky**[1] and **Preslav Nakov**[2]
[1]National Research University Higher School of Economics, Moscow, Russia
[2]Qatar Computing Research Institute, HBKU, Doha, Qatar
aschernyavskiy_1@edu.hse.ru, dilvovsky@hse.ru
pnakov@hbku.edu.qa

Abstract

We describe our system for SemEval-2020 Task 11 on *Detection of Propaganda Techniques in News Articles*. We developed ensemble models using RoBERTa-based neural architectures, additional CRF layers, transfer learning between the two subtasks, and advanced post-processing to handle the multi-label nature of the task, the consistency between nested spans, repetitions, and labels from similar spans in training. We achieved sizable improvements over baseline fine-tuned RoBERTa models, and the official evaluation ranked our system **3rd** (almost tied with the 2nd) out of 36 teams on the span identification subtask with an F1 score of 0.491, and **2nd** (almost tied with the 1st) out of 31 teams on the technique classification subtask with an F1 score of 0.62.

1 Introduction

The proliferation of disinformation online, commonly known as “fake news”, has given rise to a lot of research on automatic fake news detection. However, most of the efforts have focused on checking whether a piece of information is factually correct, and little attention has been paid to the propaganda techniques that malicious actors use to spread their message. SemEval-2020 Task 11 (Da San Martino et al., 2020a) aims to bridge this gap. It focused on detecting the use of propaganda techniques in news articles,[1] creating a dataset that extends (Da San Martino et al., 2019b), and offering two subtasks:

- **span identification (SI):** detecting the propaganda spans in an article;
- **technique classification (TC):** detecting the type of propaganda used in a given text span.

Below, we describe the systems we built for these two subtasks. At the core of our systems is RoBERTa (Liu et al., 2019), a pre-trained model based on the Transformer architecture (Vaswani et al., 2017). However, we improved over RoBERTa by adding extra layers in the neural network architecture, and we further added some post-processing steps. We further applied transfer learning between the two subtasks, and finally, we combined different models into an ensemble.[2]

2 Related Work

The dataset for the task comes from (Da San Martino et al., 2019b), which used a BERT-based model with multi-task learning and a gated architecture; the system can be tried online (Da San Martino et al., 2020c). There was also a related previous task on fine-grained propaganda detection (Da San Martino et al., 2019a), where the participants used Transformer-style models, LSTMs and ensembles (Fadel et al., 2019; Hou and Chen, 2019; Hua, 2019). Some approaches further used non-contextualized word embeddings, e.g., based on FastText and GloVe (Gupta et al., 2019; Al-Omari et al., 2019), or handcrafted features such as LIWC, quotes and questions (Alhindi et al., 2019). For the fragment-classification subtask (a combination of two subtasks of the current SemEval) the LSTM-CRF (Gupta et al., 2019) or biLSTM-CRF (Alhindi et al., 2019) models were applied besides BERT (Yu et al., 2019). Moreover, some efforts have been made to increase the size of the dataset using unsupervised language model pre-training (Yoosuf and Yang, 2019).

Finally, there is a recent survey on computational propaganda detection (Da San Martino et al., 2020b).

[1] The official task webpage: http://propaganda.qcri.org/semeval2020-task11/

[2] The code of our systems is available at http://github.com/aschern/semeval2020_task11

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1462–1468
Barcelona, Spain (Online), December 12, 2020.

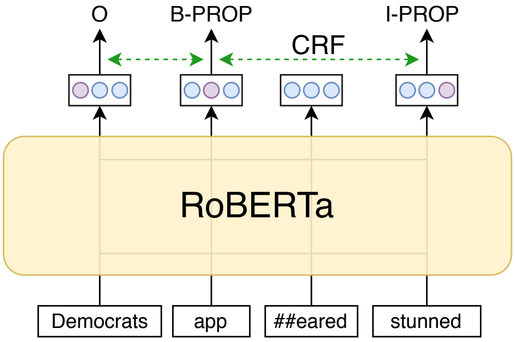

Figure 1: **SI subtask:** Our RoBERTa-CRF model with BIO encoding. It is trained end-to-end and the CRF model ignores non-starting word pieces (*##eared* in the example).

3 Our Systems

In this section, we provide a general overview of our systems for the two subtasks. For both subtasks, we trained ensembles based on RoBERTa with some postprocessing.

3.1 Subtask 1: Span Identification

Model We addressed the span identification subtask as a sequence labeling problem. To that end, we transformed the initial span markup into a BIO tagging format (Begin, Inside, Outside), as our preliminary experiments had shown that it performed better than alternatives such as IO and BIOUL (Begin, Inside, Outside, Unit, Last) (Ratinov and Roth, 2009). As we have only one possible entity class `PROP`, each token can be assigned one of the following three labels: `O`, `B-PROP`, and `I-PROP`. Then, we fine-tuned a RoBERTa model to predict the above BIO tags for each token in the input sentence.

One problem with the above setup is that each token is classified independently of the surrounding tokens: while these surrounding tokens are taken into account in the contextualized embeddings that RoBERTa produces, there is no modeling of the dependency between the predicted labels: for example, logically `I-PROP` cannot follow `O`, but RoBERTa does not model it. Thus, we further added a linear-chain Conditional Random Field (CRF) model (Lafferty et al., 2001) as an additional layer, in order to model the dependency between the labels predicted for the individual tokens; this model can observe that the sequence `O I-PROP` is never present in the training data, and thus it can assign a very low probability to the transition from an `O` tag to an `I-PROP` tag.

We trained the resulting RoBERTa-CRF model in an end-to-end fashion as shown in Figure 1. The CRF receives the logits for each input token, and makes a prediction for the entire input sequence, taking into account the dependencies between the labels, similarly to (Lample et al., 2016). Note that RoBERTa works with byte pair encoding (BPE) units, while for the CRF it makes more sense to work with words. Thus, in the input to the CRF, we only used tokens that started a word, and we skipped any word continuation tokens, e.g., a token like *##smth* would not be passed to the CRF.

Post-processing We further applied two post-processing steps to obtain the final prediction from the token classification. First, we made sure that each predicted propaganda span began and ended by either a letter or a number (alphanumerical); otherwise, we shortened the span by advancing its beginning and/or by pushing back its ending by 1-2 characters until both the beginning and the ending characters became alphanumerical, which effectively defends against tokenization errors. Second, we checked if the span was preceded and followed by quotation marks, in which case we expanded it to include them; this is helpful as sometimes propaganda techniques contain text in quotations.

Ensemble Finally, in order to increase the stability of the model, we created an ensemble of two models that have the same architecture, but are trained using different random seeds. At test time, each classifier made an independent prediction, and then we took the union of the predicted spans. In case of overlaps between spans, we created a superspan spanning the union of the respective overlapping spans.

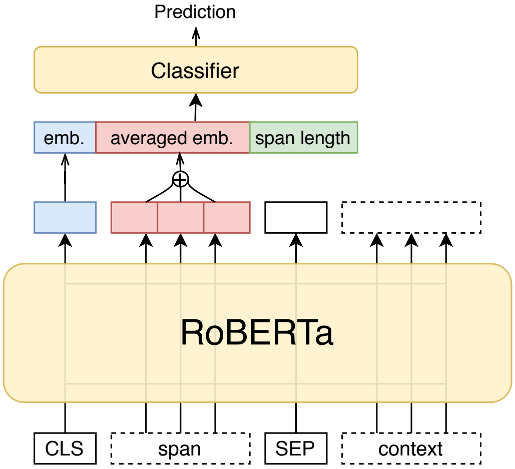

Figure 2: **TC subtask:** Our RoBERTa-based model takes as an input a target span and its context (the sentence where the span originated). The classifier takes as an input the embedding of the [CLS] token, the averaged embedding of all span tokens, and the span length.

3.2 Subtask 2: Technique Classification

Model The technique classification subtask is a multi-class *multi-label* problem, as it asks to predict one or more labels per span. As only a small number of examples in the training dataset have multiple labels (propaganda techniques), we reduced the problem to a multi-class *single-label* problem. In particular, at training time, we converted all multi-label training examples into single-label ones by creating multiple copies of each multi-label example, one copy for each of the labels; at testing time, we considered the n-best predicted labels and we decided whether to predict more than one label in a post-processing step.

Our model for the TC subtask is based on RoBERTa, and it takes the following input: `[CLS] <span> [SEP] <sentence>`, where `<sentence>` is the sentence from which the span was extracted. Then, we added a softmax layer on top of the embedding for the `[CLS]` token to make a classification prediction, and we fine-tuned the model using the training data.

We further developed a variant of the model, which concatenates (*i*) the RoBERTa embedding for the `[CLS]` token, (*ii*) the averaged embedding of the remaining tokens, and (*iii*) the length of the span. Length is important, as some propaganda techniques such as *Loaded Language* and *Name Calling* are typically short, while others such as *Causal Oversimplification* are generally longer. Finally, we added an extra fully-connected layer with the size of the original RoBERTa embeddings, as shown on Figure 2.

We further used transfer learning from the span identification subtask: we first trained the model using the data for the span identification subtask, and then we continued training for the technique classification subtask. As a result, the embeddings model how propagandistic these tokens are, which in turn can help discriminate between different types of propaganda for the TC subtask, as some propanganda techniques, such as *Loaded Language* and *Name Calling*, are short and their tokens are likely to be highly propagandistic, while other propaganda techniques are long, such as *Red Herring* and *Causal Oversimplification*, and many of the tokens in their spans are not propagadistic by themselves.

Post-processing While developing our system, we noticed that our model struggled with *Repetition*, as repetitions can occur over long distances that go beyond the maximum span length that RoBERTa can handle: 512 tokens. Thus, we added some special post-processing to handle this technique. Since all candidate spans are given for the TC subtask, we compared these spans looking for possible repetitions. We compared the spans looking for exact matches after removing punctuation, filtering out stopwords, and applying the Porter stemmer (Porter, 1980). We assigned a *Repetition* label in case the span matched at least two other spans. If it matched only one other span, we further required that the classifier predicted *Repetition* with a probability greater than 0.001. If it matched no other spans, we assigned *Repetition* a probability of 0.0, unless the classifier had predicted *Repetition* with probability of 0.99 or higher.

We further checked for each test span whether it can be found as a span in the training data (as above, for the matching, we ignored punctuation, stopwords, and we used stemming), and if so, we first collected all the propaganda techniques that the span has been seen with in the training data (note that the span might occur multiple times in training, possibly with different annotations for the different instances, and some instances could have multiple techniques assigned), and then we boosted the corresponding predicted probabilities by 0.5. This works well for spans that are likely to express the same propaganda technique(s) regardless of the context.

Next, we modeled the local consistency of the predicted spans. We observed that some long propaganda spans could contain a subspan with a different propaganda technique. We further noticed that not all combinations were equally likely, e.g., *Causal Oversimplification* was generally long and it could contain a subspan of *Loaded Language*, but it was very unlikely to see these propaganda techniques nested the other way around. Thus, we collected all possible span-subspan combinations of propaganda techniques observed in training, and we tried to discourage any other combinations. For any span-subspan combination that was not observed in training, we assigned 0.0 to the smallest of the two probabilities for the considered spans, unless the new maximum probability for the affected span would drop more than twice as a result.

Moreover, we modeled the multi-label nature of the TC subtask. We took advantage of the fact that in case a given span had to be assigned n propaganda techniques ($2 <= n <= 14$), this span would be repeated n times in the test input, i.e., the number n was known, and only the actual propaganda techniques were to be predicted. We handled this by assigning the top-n propaganda techniques for such a span, according to the calculated probabilities (after they have been potentially altered by the previous post-processing steps).

Ensemble Finally, we used model combination. In particular, we combined the simple RoBERTa model with the more sophisticated one from Figure 2: we took the posterior probabilities they produced for all propaganda techniques, and we passed them to a logistic regression classifier to make the final decision. We tuned the parameters of the classifier on part of the development dataset.

4 Experimental Setup

Data We experimented with the training, the development and the test datasets provided for SemEval-2020 Task 11, which contain 371, 75, and 90 news articles with 6,128, 1,063 and 1,790 spans, respectively. We randomly selected 20% of the training dataset for local evaluation when developing our models.

Evaluation measures The official evaluation measures are a “normalized” version of the F1 score for the span identification subtask, and micro-averaged F1 score for the technique classification subtask. A detailed description of the evaluation measures can be found in the SemEval-2020 task 11 paper (Da San Martino et al., 2020a).

Parameter settings We used the RoBERTa-large model and the following hyper-parameters, which we selected using validation on a subsample of the training data: learning rate of 2e-5, batch size of 24, and RoBERTa’s default optimizer with 500 warm-up steps. We further found that an uncased model should be used for the span identification subtask, but that a cased model worked better for the technique classification subtask. We trained the models for 30 epochs, we saved a checkpoint every two epochs, and we selected the best checkpoint on the validation subsample.

5 Results

5.1 Subtask 1: Span Identification

On the development set, our fine-tuned BIO-encoded RoBERTa-large model achieved an F1 score of 47.8; see Table 1. Adding a CRF layer pushed F1 to 48.8, and also yielded higher stability of the results, i.e., less variation across reruns. The ensemble of these two models improved F1 score to 49.6. Finally, adding post-processing of punctuation and quotation symbols pushed the final F1 score to 49.9.

System	F1
RoBERTa-large with BIO encoding (fine-tuned)	0.478
+ CRF layer	0.488
+ ensemble	0.496
+ post-processing	0.499

Table 1: **SI subtask (dev):** An incremental analysis of our system.

Rank	Team	F1	Prec.	Recall
1	Hitachi	51.55	56.54	47.37
2	ApplicaAI	49.15	59.95	41.65
3	**aschern**	49.10	53.23	45.56
4	LTIatCMU	47.66	50.97	44.76
36	baseline	0.31	13.04	0.16

Table 2: **SI subtask (test):** Top-4 teams on the test set.

The official results on the blind test dataset are shown in Table 2. We can see that our official submission is ranked third and is almost tied with the second one, with a difference of only 0.05 F1 points absolute.

5.2 Subtask 2: Technique Classification

On the development set, our fine-tuned RoBERTa-large model achieved an F1 score of 62.18; see Table 3. Adding length and averaged span embeddings, improved the results only marginally. Marginal was also the improvement from multi-label correction and from bonus for span labels seen on training, which together only added 0.5 F1 points absolute. However, handling of *Repetition* yielded a huge improvement of over 3.5 F1 points absolute. Further adding checking for unseen span-subspan combinations yielded marginal gains. Finally, using an ensemble improved the F1 score by 1.5 F1 points absolute to 68.10.

The official results on the blind test dataset are shown in Table 4. We can see that our official submission is ranked second and is almost tied with the first team, with a difference of only 0.06 F1 points absolute.

6 Conclusion and Future Work

We described the system we developed for the SemEval-2020 Task 11 on Detection of Propaganda Techniques in News Articles. We developed ensembles of RoBERTa-based neural architectures, additional CRF layers, transfer learning between the two subtasks, and advanced post-processing to handle the multi-label nature of the task, the consistency between nested spans, repetitions, and labels from similar spans in training. We achieved sizable improvements over baseline RoBERTa models, and the official evaluation ranked our system **3$^{\mathbf{rd}}$** (almost tied with the 2nd) out of 36 teams on the span identification subtask, and **2$^{\mathbf{nd}}$** (almost tied with the 1st) out of 31 teams on the techniques classification subtask.

In future work, we plan to explore other neural architectures such as T5 (Raffel et al., 2019) and GPT-3 (Brown et al., 2020). We further want to explore transfer learning from other tasks such as argumentation mining (Stede et al., 2018) and offensive language detection (Zampieri et al., 2019; Zampieri et al., 2020).

7 Acknowledgments

Anton Chernyavskiy and Dmitry Ilvovsky performed this research within the framework of the HSE University Basic Research Program, funded by the Russian Academic Excellence Project '5-100'.

Preslav Nakov contributed as part of the Propaganda Analysis Project (`propaganda.qcri.org`), part of the Tanbih megaproject (`tanbih.qcri.org`), developed at the Qatar Computing Research Institute, HBKU. Tanbih aims to limit the effect of "fake news", propaganda, and media bias by making users aware of what they are reading, thus promoting media literacy and critical thinking.

System	F1
RoBERTa-large (fine-tuned)	62.18
+ length + averaged span embeddings	62.37
+ post-processing: multi-label correction	62.75
+ post-processing: bonus for span labels seen on training	62.85
+ post-processing: handling of *Repetition*	66.42
+ post-processing: checking for unseen span-subspan combinations	66.60
+ ensemble	**68.10**

Table 3: **TC subtask (dev):** An incremental analysis of our system on the development set.

Rank	Team	F1
1	ApplicaAI	62.07
2	**aschern**	62.01
3	Hitachi	61.73
4	Solomon	58.94
30	baseline	25.20

Table 4: **TC subtask (test):** Results on the test set (top-4 teams).

References

Hani Al-Omari, Malak Abdullah, Ola AlTiti, and Samira Shaikh. 2019. JUSTDeep at NLP4IF 2019 task 1: Propaganda detection using ensemble deep learning models. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 113–118, Hong Kong, China.

Tariq Alhindi, Jonas Pfeiffer, and Smaranda Muresan. 2019. Fine-tuned neural models for propaganda detection at the sentence and fragment levels. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 98–102, Hong Kong, China.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Arxiv preprint*, abs/2005.14165.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019a. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 162–170, Hong Kong, China.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019b. 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 '19, pages 5636–5646, Hong Kong, China.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020a. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020b. A survey on computational propaganda detection. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence*, IJCAI-PRICAI '20, Yokohama, Japan.

Giovanni Da San Martino, Shaden Shaar, Yifan Zhang, Seunghak Yu, Alberto Barrón-Cedeño, and Preslav Nakov. 2020c. Prta: A system to support the analysis of propaganda techniques in the news. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, ACL '20, pages 287–293, Seattle, WA, USA.

Ali Fadel, Ibraheem Tuffaha, and Mahmoud Al-Ayyoub. 2019. Pretrained ensemble learning for fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 139–142, Hong Kong, China.

Pankaj Gupta, Khushbu Saxena, Usama Yaseen, Thomas Runkler, and Hinrich Schütze. 2019. Neural architectures for fine-grained propaganda detection in news. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 92–97, Hong Kong, China.

Wenjun Hou and Ying Chen. 2019. CAUnLP at NLP4IF 2019 shared task: Context-dependent BERT for sentence-level propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 83–86, Hong Kong, China.

Yiqing Hua. 2019. Understanding BERT performance in propaganda analysis. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 135–138, Hong Kong, China.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, pages 260–270, San Diego, CA, USA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Arxiv preprint*, abs/1907.11692.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

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*, abs/1910.10683.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Boulder, CO, USA.

M. Stede, J. Schneider, and G. Hirst. 2018. *Argumentation Mining*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pages 5998–6008. Long Beach, CA, USA.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned BERT. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 87–91, Hong Kong, China.

Seunghak Yu, Giovanni Da San Martino, and Preslav Nakov. 2019. Experiments in detecting persuasion techniques in the news. In *Proceedings of the NeurIPS 2019 workshop on AI for Social Good*, Vancouver, Canada.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 1415–1420, Minneapolis, MN, USA.

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. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '20, Barcelona, Spain.