Kungfupanda at SemEval-2020 Task 12: BERT-Based Multi-Task Learning for Offensive Language Detection

Wenliang Dai*, Tiezheng Yu*, Zihan Liu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{wdaiai,tyuah,zliucr}@connect.ust.hk, pascale@ece.ust.hk

Abstract

Nowadays, offensive content in social media has become a serious problem, and automatically detecting offensive language is an essential task. In this paper, we build an offensive language detection system, which combines multi-task learning with BERT-based models. Using a pre-trained language model such as BERT, we can effectively learn the representations for noisy text in social media. Besides, to boost the performance of offensive language detection, we leverage the supervision signals from other related tasks. In the OffensEval-2020 competition, our model achieves 91.51% F1 score in English Sub-task A, which is comparable to the first place (92.23% F1). An empirical analysis is provided to explain the effectiveness of our approaches.

1 Introduction

Nowadays, offensive content has invaded social media and becomes a serious problem for government organizations, online communities, and social media platforms. Therefore, it is essential to automatically detect and throttle the offensive content before it appears in social media. Previous studies have investigated different aspects of offensive languages such as abusive language (Nobata et al., 2016; Mubarak et al., 2017) and hate speech (Malmasi and Zampieri, 2017; Davidson et al., 2017).

Recently, (Zampieri et al., 2019a) first studied the target of the offensive language in twitter and (Zampieri et al., 2020) expand it into the multilingual version, which is practical for studying hate speech concerning a specific target. The task is based on a three-level hierarchical annotation schema that encompasses the following three general sub-tasks: (A) Offensive Language Detection; (B) Categorization of Offensive Language; (C) Offensive Language Target Identification.

To tackle this problem, we emphasize that it is crucial to leverage pre-trained language model (e.g., BERT (Devlin et al., 2018)) to better understand the meaning of sentences and generate expressive word-level representations due to the inherent data noise (e.g., misspelling, grammatical mistakes) in social media (e.g., twitter). In addition, we hypothesize that the internal connections exist among the three general sub-tasks, and to improve one task, we can leverage the information of the other two tasks. Therefore, we first generate the representations of the input text based on the pre-trained language model BERT, and then we conduct multi-task learning based on the representations.

Experimental results show that leveraging more task information can improve the offensive language detection performance. In the OffensEval-2020 competition, our system achieves 91.51% macro-F1 score in English Sub-task A (ranked 7th out of 85 submissions). Especially, only the OLID (Zampieri et al., 2019a) is used to train our model and no additional data is used. Our code is available at: https://github.com/wenliangdai/multi-task-offensive-language-detection.

2 Related Works

In general, offensive language detection includes some particular types, such as aggression identification (Kumar et al., 2018), bullying detection (Huang et al., 2014) and hate speech identification (Park

* These two authors contributed equally.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2060–2066
Barcelona, Spain (Online), December 12, 2020.

and Fung, 2017). (Chen et al., 2012) applied concepts from NLP to exploit the lexical syntactic feature of sentences for offensive language detection. (Huang et al., 2014) integrated the textual features with social network features, which significantly improved cyberbullying detection. (Park and Fung, 2017) and (Gambäck and Sikdar, 2017) used convolutional neural network in the hate-speech detection in tweets. Recently, (Zampieri et al., 2019a) introduce an offensive language identification dataset, which aims to detect the type and the target of offensive posts in social media. (Zampieri et al., 2020) expanded this dataset into the multilingual version, which advances the multilingual research in this area.

Pre-trained language models, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) have achieved great performance on a variety of tasks. Many recent papers have used a basic recipe of fine-tuning such pre-trained models on a certain domain (Azzouza et al., 2019; Lee et al., 2019; Beltagy et al., 2019) or on downstream tasks (Howard and Ruder, 2018; Liu et al., 2019b; Su et al., 2019).

3 Datasets

In this project, two datasets are involved, which are the dataset of OffensEval-2019 and OffensEval-2020 respectively. In this section, we introduce the details of them and discuss our data pre-processing methods. Table 1 shows the types of labels and how they overlap.

Task A	OFF			NOT
Task B	TIN		UNT	NULL
Task C	IND	GRP	NULL	

Table 1: Labels of three subtasks.

3.1 Offensive Language Identification Dataset (OLID)

The OLID (Zampieri et al., 2019b) is a hierarchical dataset to identify the type and the target of offensive texts in social media. The dataset is collected on Twitter and publicly available. There are 14,100 tweets in total, in which 13,240 are in the training set, and 860 are in the test set. For each tweet, there are three levels of labels: (A) Offensive/Not-Offensive, (B) Targeted-Insult/Untargeted, (C) Individual/Group/Other. The relationship between them is hierarchical. If a tweet is offensive, it can have a target or no target. If it is offensive to a specific target, the target can be an individual, a group, or some other objects. This dataset is used in the OffensEval-2019 competition in SemEval-2019 (Zampieri et al., 2019c). The competition contains three sub-tasks, each corresponds to recognizing one level of label in the dataset.

3.2 Semi-Supervised Offensive Language Identification Dataset (SOLID)

A semi-supervised offensive language detection dataset (SOLID) (Rosenthal et al., 2020) for English is proposed in the OffensEval-2020 competition in SemEval-2020 (Zampieri et al., 2020). Similar to OLID (Zampieri et al., 2019b), it still has three levels but the data in level A is separated from levels B and C. In level A, there are 9,089,140 tweets, in levels B and C, there are different 188,973 tweets. For each entry of data in level A, the mean and standard deviation of confidence scores generated by the democratic co-training approach are provided as supervision. Because of this, the data is more noisy than OLID. For level B and C, the data is still manually annotated.

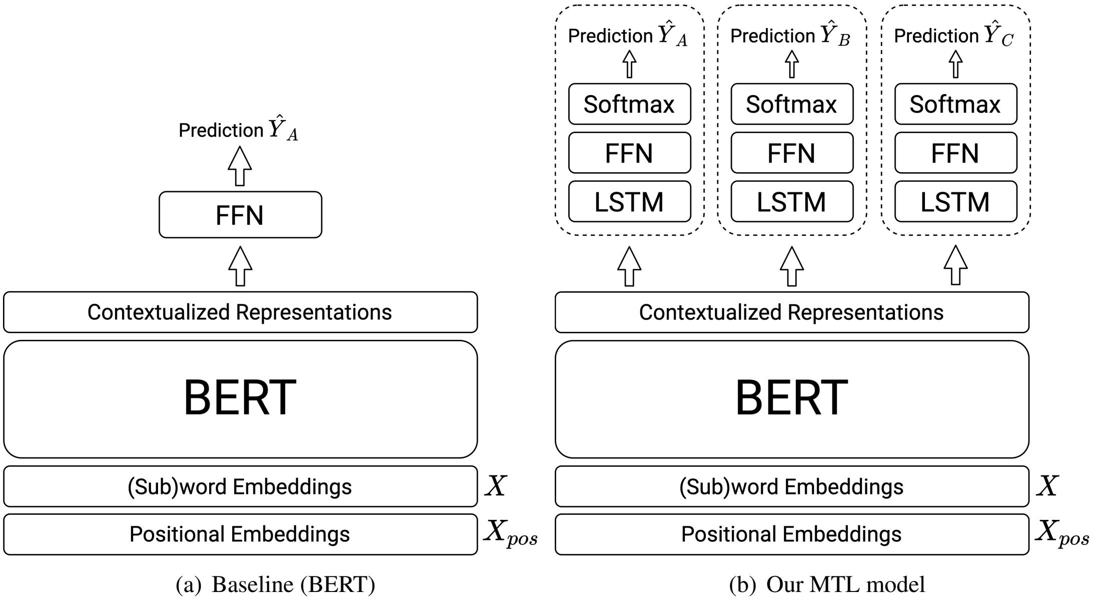
3.3 Data Pre-processing

Data pre-processing is crucial to this task as the data from Twitter is noisy and sometimes disordered. Moreover, people tend to use more Emojis and hashtags on Twitter, which are unusual in other situations.

Firstly, all characters are converted to lowercase, and the spaces at ends are stripped. Then, inspired by (Zampieri et al., 2019c; Liu et al., 2019a), we further process the dataset in five specific aspects:

Emoji to word. We convert all emojis to words with corresponding semantic meanings. For example, 👍 is converted to *thumbs up*. We achieve this by first utilizing a third-party Python library [1], and then

[1] https://github.com/carpedm20/emoji



(a) Baseline (BERT)

(b) Our MTL model

Figure 1: (a) BERT baseline model. (b) Our MTL model. The bottom is a shared BERT backbone, and the upper parts are separate modules dedicated for each sub-task.

removing useless punctuation in it.

Hashtag segmentation. All hashtags in the tweets are segmented by recognizing the capital characters. For example, *#KeithEllisonAbuse* is transformed to *keith ellison abuse*. This is also achieved by using a third-party Python library [2].

User mention replacement. After reviewing the dataset, we find out that the token *@USER* appears very frequently (a single tweet can have multiple of them), which is a typical phenomenon in tweets. As a result, for those with more than one *@USER* token, we replace all of them with one *@USERS* token. In this way, we remove the redundant words while keeping the key information, which is useful for recognizing targets if there is any.

Rare word substitution. We substitute some out-of-vocabulary (OOV) words with their synonyms. For example, every *URL* is replaced with a special token, *http*.

Truncation. We truncate all the tweets to a max length of 64. Although this can get rid of some information in the data, it lowers the GPU memory usage and slightly improves the performance.

4 Methodology

We propose a Multi-Task Learning (MTL) method (Figure 1(b)) for this offensive language detection task. It takes good advantage of the nature of the OLID (Zampieri et al., 2019b), and achieves an excellent result comparable to state-of-the-art performance only with the OLID (Zampieri et al., 2019b) and no external data resources. A thorough analysis is provided in Section 5.2 to explain the reasons of not using the new SOLID dataset created in OffensEval-2020 (Zampieri et al., 2020).

4.1 Task Description

The OffensEval-2020 (Zampieri et al., 2020) is a task that organized at SemEval-2020 Workshop. As mentioned in Section 3.2, it proposes a semi-supervised multilingual offensive language detection dataset which contains five different languages. It has three sub-tasks: (A) Offensive Language Detection; (B) Categorization of Offensive Language; (C) Offensive Language Target Identification. In this paper, we mainly focus on the sub-task A of the English data (Rosenthal et al., 2020).

[2] https://github.com/grantjenks/python-wordsegment

4.2 Baseline

We re-implement the model of the best performing team (Liu et al., 2019a) in OffensEval-2019 (Zampieri et al., 2019c) as our baseline. As illustrated in Figure 1(a), (Liu et al., 2019a) fine-tuned the pre-trained model, BERT (Devlin et al., 2018), by adding a linear layer on top of it.

BERT. Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2018) is a large-scale masked language model based on the encoder of Transformer model (Vaswani et al., 2017). It is pre-trained on the BookCorpus (Zhu et al., 2015) and English Wikipedia datasets using two unsupervised tasks: (a) Masked Language Model (MLM) (b) Next Sentence Prediction (NSP). In MLM, 15% of input tokens are masked, and the model is trained to recover them at the output. In NSP, two sentences are fed into the model and it is trained to predict whether the second sentence is the actual next sentence of the first one. As shown in (Devlin et al., 2018), by fine-tuning, BERT achieves superior results on many NLP downstream tasks.

4.3 Multi-task Offense Detection Model

In recent years, multi-task learning (MTL) technique is used in many machine learning fields to improve performance and generalization ability of a model (Kang et al., 2011; Long and Wang, 2015; Kokkinos, 2016; Güler et al., 2018; Liu and Zhao, 2018; Dankers et al., 2019). Generally, MTL has three advantages. Firstly, with multiple supervision signals, it can improve the quality of representation learning, because a good representation should have better performance on more tasks. Secondly, MTL can help the model generalize better because multiple tasks introduce more noises and prevent the model from over-fitting. Thirdly, sometimes it is hard to learn features by one task but easier to learn by another task. MTL provides complementary supervisions to one task and makes it possible to eavesdrop other tasks and get more information.

For this task, MTL is a very effective strategy. As mentioned in Section 3.1 and shown in Table 2, the three labels in OLID are hierarchical and they are designed to be inclusive from top to bottom. This makes it possible for one sub-task to eavesdrop information form the other tasks. For example, if a tweet is labelled as Targeted in sub-task B, then it must be classified to Offensive in sub-task A.

Our MTL architecture is shown in Figure 1(b). The bottom part is a BERT model, which is shared among all three sub-tasks. The upper parts are three separate modules dedicated for each sub-task, each module contains a Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997). The input X is first fed into the shared BERT, then each sub-task module takes the contextualized embeddings generated by BERT and produces a probability distribution for its own target labels. The overall loss L is calculated by $L = \sum_i^I w_i L_i$. Here, $I = \{A, B, C\}$ and w_i is the loss weight for each task-specific Cross-Entropy loss L_i, where $\sum_i^I w_i = 1$. The loss weights are chosen by cross validation.

5 Experiments

During the training phase, we evaluate our models on the test set of OLID (OffensEval-2019). As a reference, we also evaluate them on the test set of SOLID (OffensEval-2020), which is only released after the submission date.

5.1 Experimental Settings

To find the optimal architecture for this task within the models we have, we set up five different experiments (Table 2). For the first two, we train our baseline model on OLID and SOLID separately. As SOLID's labels are AVG_CONF scores between 0 to 1 rather than binary classes, we set the threshold as 0.3 to convert SOLID to a classification dataset. We set this threshold value by first training a BERT classifier on the OLID dataset, and then do predictions on SOLID and choose the best threshold by cross validation. Besides, we also conduct an experiment that pre-train the baseline model on SOLID and fine-tune on OLID by utilizing the pre-train strategy discussed in Session 5.2. Finally, we train our Multi-task Offense Detection Model only on OLID and fine-tune the hyper-parameters based on Sub-task A (i.e. the

Model	F1 - OLID	F1 - SOLID
BERT (OLID)	0.8203	0.9088
BERT (SOLID)	0.7280	0.9060
BERT(Pre-trained with MSE loss)	0.8138	0.9107
BERT + MTL	0.8341	0.9139
BERT + MTL (Ensemble)	**0.8382**	**0.9151**

Table 2: Experimental results on sub-task A. The evaluation metric is macro F1 score, which is official in OffensEval-2020.

goal is to leverage the information in level B and C to improve level A). To further improve the generalization performance of our method, we ensemble five MTL models with different parameter initialization and generate final results through majority voting.

To evaluate the performance of each model, we use macro-F1 which is computed as a simple arithmetic mean of per-class F1-scores. Since OLID released its test set last year, we use this test set as our validation set and optimize the hyper-parameters manually over the successive runs on it. For our best MTL model, we set the learning rate as 3e-6 and batch size as 32, the loss weights of subtasks A, B, C are 0.4, 0.3, 0.3 respectively. We train the model with maximum 20 epochs and utilize an early stop strategy to stop training if the validation macro-F1 doesn't increase in three continuous epochs. Our code is implemented in PyTorch and all experiments are run on a single GTX 1080Ti.

5.2 Result Analysis

The results on Table 2 show the macro-F1 scores on OLID and SOLID's test set and they are consistent except the model with pre-training. Our ensembled MTL model achieves the best performance in both two test sets.

Pre-train vs. No pre-train on SOLID. Since the SOLID (Zampieri et al., 2020) contains more than 9 million samples with the AVG_CONF score. To make full use of the dataset, we conduct pre-train strategy which let the model pre-trained on SOLID and then fine-tuned on the Offensive Language Identification Dataset(OLID) (Zampieri et al., 2019b). To pre-train the model on SOLID, we regard the Sub-task A as a regression problem based on the AVG_CONF score. Instead of setting a threshold to divide the data into two classes(OFF, NOT), we directly apply Mean Square Error(MSE) loss function on AVG_CONF. However, our result shows that conducting pre-training makes little difference. We believe it is because the SOLID contains lots of noisy data which is also the reason why the baseline model trained on SOLID is much worse than on OLID.

BERT and Multi-Task Learning From the result, we find that incorporating BERT and multi-task learning can help improve the macro-F1 score of Sub-task A a lot. This can be attributed to two reasons. Firstly, BERT model is pre-trained on a huge corpus which helps to produce more meaningful representations for the input text. Meanwhile, the large model size increases the learning ability for the task. Secondly, with the large capacity of BERT, through multi-task learning, sub-task A can get more information from the other shared part of the model, and it will be more certain to some cases. For example, if the label of sub-task B is *NULL*, then label of sub-task A must be *NOT*. If the label of sub-task B is *TIN* or *UNT*, then the label of sub-task A must be *OFF*.

6 Conclusion and Future work

From all of our experiments, we conclude that MTL improves the performance of Sub-task A in both OLID and SOLID. Moreover, our finding shows that pre-training Sub-task A as a regression task doesn't improve the model's performance. We think that there are several paths for further work. Firstly, more studies about the combination of the sub-tasks can be investigated for MTL. This can show us more about the interaction between sub-tasks, and how much does one influence another. Secondly, as mentioned in (Kokkinos, 2016), simultaneously updating the model's parameters during MTL can have negative effects on optimization as the total gradients are too noisy. This becomes more significant when the number of tasks is large or the batch size is small. As a result, asynchronous optimizations for each task may provide a more stable gradient descent.

Acknowledgements

This work is funded by MRP/055/18 of the Innovation Technology Commission, the Hong Kong SAR Government.

References

Noureddine Azzouza, Karima Akli-Astouati, and Roliana Ibrahim. 2019. Twitterbert: Framework for twitter sentiment analysis based on pre-trained language model representations. In *International Conference of Reliable Information and Communication Technology*, pages 428–437. Springer.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. 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 3606–3611.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE.

Verna Dankers, Marek Rei, Martha Lewis, and Ekaterina Shutova. 2019. Modelling the interplay of metaphor and emotion through multitask learning. 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 2218–2229, Hong Kong, China, November. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. *CoRR*, abs/1802.00434.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Qianjia Huang, Vivek Kumar Singh, and Pradeep Kumar Atrey. 2014. Cyber bullying detection using social and textual analysis. In *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*, pages 3–6. ACM.

Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 521–528, USA. Omnipress.

Iasonas Kokkinos. 2016. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *CoRR*, abs/1609.02132.

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

Nayeon Lee, Zihan Liu, and Pascale Fung. 2019. Team yeon-zi at semeval-2019 task 4: Hyperpartisan news detection by de-noising weakly-labeled data. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1052–1056.

Yu Liu and Guanlong Zhao. 2018. Pad-net: A perception-aided single image dehazing network. *CoRR*, abs/1805.03146.

Ping Liu, Wen Li, and Liang Zou. 2019a. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2019b. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. *arXiv preprint arXiv:1911.09273*.

Mingsheng Long and Jianmin Wang. 2015. Learning multiple tasks with deep relationship networks. *CoRR*, abs/1506.02117.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 467–472.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. In *arxiv*.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing question answering system with pre-trained language model fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 203–211.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019c. 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, June. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR*, abs/1506.06724.