Offensive tweet classification - An explainable pre-trained language model

Matej Kucera
s4551192
m.kucera@student.rug.nl

Abstract

Social media produces an enormous volume of content. The anonymity of the Internet allows a part of this content to be offensive. Due to the volume generated, an automated method to moderate offensive content on social media is needed. This paper uses the OLID dataset to train and evaluate multiple machine learning models in the task of offensive tweet classification. The RoBERTa model achieves an F1-score of 80% on unseen data. An investigation is carried out into the reasoning of the model. The results show that the model mainly focuses on individual words and that it does not identify veiled offense. This is proven by randomly shuffling the input without a significant F1-score decrease. Secondly, a list of words which the model considers offensive is compiled. These results explain how the model classifies results.

1 Introduction

Social media is an important platform for self-expression and freedom of speech. However they can be used anonymously or semi-anonymously. This can often lead to users feeling the freedom to spread toxic or offensive messages. It is therefore important that social media employs some form of filtering system which can catch offensive or abusive language and label it accordingly, either to mitigate the risk of other users seeing this content or to outright remove it.

This task can be relegated to human moderators, but doing so carries significant risk. Humans tasked with annotating offensive text may suffer from various mental problems due to the nature of the task. The very nature of potentially reading numerous abusive or offensive messages can influence these moderators negatively. Secondly, the amount of content to sift through is simply too overpowering for human moderation. Therefore it is preferred to find an automated system which can analyze text and decide whether it is offensive or not. As this is a Natural Language Processing (NLP) task, multiple Machine Learning (ML) models which are known to excel at NLP can be utilized.

Previous work has investigated various forms of tackling this problem. Some review studies have summarized and reviewed existing approaches to provide guidance for future research in this direction such as (Yin and Zubiaga, 2022) and (Chinivar et al., 2023). There are multiple datasets with labelled offensive texts to aid in this task, such as the OLID dataset (Zampieri et al., 2019a) which will be used in this study. All of the effort spent in this area of research leads to better online moderation and increased safety on social media.

To extend the existing work, it was decided to focus on ML explainability. To this end, a model which performs optimally on the OLID dataset will be found. To attempt to understand how it classifies text, the following research questions are posed:

1. **Which ML model achieves the highest F1-score in offensive tweet classification on the OLID dataset?**
2. **Which hyperparameter settings for the chosen model achieve the highest F1-score?**
3. **Does the model performance decrease on randomly shuffled text?**
4. **Which features does the model use to determine whether a piece of text is offensive or not?**

The answers to these questions will improve the understanding of the ML model and aid further fine-tuning and model selection. They will also provide some human-understandable explanations for how the ML model actually classifies tweets.

2 Related Work

The OLID dataset being used in this paper was the subject of a competition at the Workshop on Semantic Evaluation (May et al., 2019). Because of this, the dataset comes pre-divided into train, dev and test data. The results are reported in a summary paper (Zampieri et al., 2019b). These results can be used as a reference to evaluate the results obtained in this paper. The metric being reported is the macro average F1-score. This takes the non-weighted average of F1 scores for each class. Since the OFF class has fewer samples, this punishes lower F1 scores for this class more than a weighted average.

The best recorded macro average F1-score in the competition was 0,829 achieved by (Liu et al., 2019). The approach used the BERT Pre-trained Language Model (PLM) with extra pre-processing steps. This performance shows that the task is actually quite difficult. Some possible reasons for this are explored in section 3.

Other related work explores the possibility of using classical ML methods such as Naive Bayes (NB), Decision Trees (DT) and Support Vector Machines (SVM) among others. They find that SVM paired with a TF-IDF vectorizer is among the best performers with an F1-score of 0,94 (Hajibabaee et al., 2022). This is achieved on a different dataset so the score is not comparable, but it does show that the task of classifying offensive text is not a hard one and can be relatively well by simple ML models.

Other research has shown that the use of Long Short Term Memory (LSTM) models is also highly effective for this task. A paper on classifying offensive text on social media shows that an LSTM-based model achieves an F1-score of 0,926 on a dataset of Bengali offensive text from social platforms (Wadud et al., 2022).

Overall, it is clear that classical and novel approaches can achieve impressive results in offensive text classification. Depending on the dataset and implementation the task can be solved by simple models or be challenging for even the best state of the art NLP models.

3 Data

Offensive language only makes up around 3% of all twitter posts (Yin and Zubiaga, 2022). This makes collecting datasets difficult due to the inherent class imbalance. The OLID dataset tackles this problem by searching tweets for keywords which tend to be more associated with offensive language, such as “she is” or “gun control”. However, even with this approach they arrive at a ratio of about 1:2 of offensive tweets to non-offensive tweets. It is also unclear how they picked those specific phrases, as no explanation is given.

The data is provided as a set of tweets divided into two classes, offensive and not offensive. The labels are OFF and NOT, respectively. The authors report a Fleiss’ *kappa* value of 0.83, indicating a high level of annotator agreement. The text is cleaned by replacing all URL links by the “URL” token and any user mentions with the “@USER” token. It is unclear whether the cleaning step was applied before or after the manual annotation of the dataset. The dataset is imbalanced with approximately two thirds of the data being the NOT class for train, dev and test.

Upon manual inspection of the dataset, it was found that the labels are not always consistent and sometimes entirely wrong. Table 1 shows some examples of inconsistencies which were found in the train and dev parts of the dataset. The study which created this dataset states that offensive tweets are: “Posts containing any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct.”. However, as can be seen in Table 1, There are multiple instances where profanities such as “nigger” or “Fuckin” are labelled as not offensive. Secondly, there are some instances where a tweet not containing profanity or targeted offense (such as rows 10 and 11) is labelled as offensive.

There are also inconsistencies regarding some terms begin offensive or not. Rows 4 - 7 all contain the word “sexy” but only two are labelled as offensive. One could argue that “sexy” is only offensive if targeted at another user, but rows 4 and 5 disprove that immediately. A similar problem occurs with the phrase “gun control” in rows 1 - 3, where the labels are inconsistent as well. Rows 1 and 2 are labelled offensive even though they contain no profanities, hence suggesting that the phrase “gun control” itself may be considered offensive. But row 3 immediately disproves this theory.

No.	Tweet	Label
1	@USER California? How? Gun control laws should of prevented this	OFF
2	@USER I see you support gun control	OFF
3	@USER @USER I believe ALL of this stuff is to bring in Gun Control. Gun Control = Disarming America.	NOT
4	@USER She is sexy without even trying.	NOT
5	@USER Super sexy	NOT
6	@USER Hi sexy	OFF
7	@USER And I you ya sexy lady you	OFF
8	@USER I'm that nigger no lie	NOT
9	@USER Fuckin love king tulip	NOT
10	@USER Another lie. Anything for a diversion	OFF
11	@USER @USER Miriam	OFF

Table 1: Examples of inconsistent labels in the OLID dataset

It is unclear how many similar inconsistencies exist in the dataset. These examples were found by manually examining the data and picking some examples. The reported annotator agreement is very high, so the data was not expected to contain such obvious problems.

4 Method

To find the best performing model for the classification task, multiple ML models were used. First, a baseline was established by fine-tuning an SVM model with no additional pre-processing. Secondly, the SVM was enhanced with an optimized feature set to improve its performance. These models served as a baseline for the performance of classical ML models. The SVM is a model which learns a linear separation plane in a high-dimensional space to separate the data into classes. It can use the kernel trick to implicitly transform the data into a higher dimension before classifying to improve its power.

To test the power of LSTM-based models, a fine-tuned bi-directional LSTM model using GloVe embeddings (Pennington et al., 2014) was used. LSTM is a deep-learning model based on the Recurrent Neural Network (RNN) architecture (Hochreiter, 1997). Is is an improvement over the regular RNN because it includes an extra inter-layer connection which lets it carry long-term dependencies. This allows the LSTM to remember context and perform better on longer pieces of text where it can carry information forward through the layers.

Lastly, PLMs were used. PLMs are the state of the art in NLP and are therefore expected to outperform classical models. This comes at the cost of computational power, as PLMs require much more calculation and therefore are a lot more computationally expensive to run than classical models. The BERT model was the first PLM (Devlin et al., 2019), but there are many models improving upon it in various ways such as RoBERTa (Conneau et al., 2019) or DistilBERT (Sanh et al., 2019). Multiple PLMs were fine-tuned and compared to evaluate which one performs best since they were expected to outperform the other methods. BERT was used as a baseline PLM. RoBERTa claims better performance than BERT upon which it is based, so it was used as an improved version. Thirdly, DistilBERT was used as a more efficient PLM to investigate the performance achievable by a lighter PLM.

Once the best model was found, an investigation was carried out to find some clues as to how it classifies text. To answer research question 1, the inputs were pre-processed by randomly shuffling the word order and seeing whether this affects the accuracy. Considering the nature of the dataset, is was expected that this would not cause a large drop in performance since most of the data is classified as offensive based on profanities.

Secondly, to find out which words the model considers important for classification, the inputs were preprocessed by removing one word at a time and running the model on this data. If the removal of a word causes the model to classify an offensive tweet as non-offensive, it is clear that that word is an important feature. This investigation was also carried out for bigrams, but turned out not effective due to the nature of the data.

To implement all models, Python was used. The scikit-learn library (Pedregosa et al., 2011) contains implementations of all classical models. For the deep learning models, the tensorflow (Abadi et al., 2015), keras (Chollet and others, 2015) and transformers (Wolf et al., 2020) libraries were used. The nltk library was used for tokenization for classical models (Bird et al., 2009).

Computational resources were kindly provided by the University of Groningen. The Hábrók cluster was used for access to GPU compute resources for the training of PLMs.

The source code for all experiments can be found in the researcher's GitHub repository [1]. All random generators in all experimental runs were seeded to ensure reproducibility.

5 Results

The results will be presented for all algorithms mentioned in section 4. All F1 scores will be presented as unweighted macro average F1 scores on a withheld test set, which was not used for fine-tuning.

5.1 Baselines

For an initial baseline, a list of profanities was used. This represents the brute-force approach of simply checking of a text contains profanities from a predefined list (Anger, 2023). This method achieved a F1-score of 0,61.

For a better baseline, a fine-tuned SVM model with a bag-of-words vectorizer was used. The implementation used the SVC and CountVectorizer classes from the scikit-learn library. Table 2 shows the range of parameters tested. The best parameters were found to be $C = 0.2$ and kernel='linear', highlighted in the table. Other parameters were left as the default values. Several vectorizer parameters were tested as well, arriving at max_features=$50,000$. The word_tokenize function from the nltk library was used to tokenize each tweet. This model achieved an F1-score of 0,72.

Hyperparameter	Values Tested
C	0.1, **0.2**, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Kernel	**linear**, rbf, poly

Table 2: Hyperparameters tested for SVM model

The last baseline model was the SVM model with additional feature optimization. The optimization which were tried were:

- The SMOTE sampling technique (Chawla et al., 2002) - This technique can be used to fix imbalanced datasets. It combines oversampling of the minority class and undersampling the majority class to achieve better results than using each technique on their own, respectively. The imblearn library's SMOTE class was used (Lemaître et al., 2017).

- N-grams - The vectorizer can construct features for n-grams of varying lengths which can help with phrases which may be lost when vectorizing per-word. Bigrams, trigrams and a combination of both were tried.

- POS-tagging - adding a POS tag to each word can aid the model in understanding the context in which a word is used. This helps to distinguish different usages of the same word and can therefore be beneficial.

- TF-IDF vectorization - The TF-IDF formula provides a better set of features by accounting for the informational value of each word based on its frequency in the entire corpus. The TfidfVectorizer class from scikit-learn was used.

- class_weight - This is a parameter of the SVC class which can adjust the class weight based on frequency when given the 'balanced' value.

- max_df - This is a vectorizer parameter which removes words with a document frequency above the given value. Thsi can help filter out words which are so common that they carry no informational value.

It was found that none of these optimizations actually increased the F1-score of the model.

[1] https://github.com/siued/Learning-From-Data

5.2 LSTM

The GloVe pre-trained embeddings were used for this experiment (Pennington et al., 2014). Since the dataset contains tweets, the twitter.27B embeddings were used. Since there are multiple dimensionalities, the 100D embeddings were used for a combination of efficiency and performance.

Table 3 shows the various hyperparameters which were tested. The parameters of the final, best performing LSTM model are highlighted in boldface. Notably, the best performing model was one with a single LSTM layer containing just one LSTM unit.

Hyperparameter	Values Tested
Learning Rate	1e-4, **5e-4**, 1e-5
Weight Decay	0.0, **0.1**, 0.2, 0.3
Momentum	0.0, **0.5**, 0.9, 0.99
LSTM Layers	**1**, 2, 3, 4
LSTM units per layer	**1**, 4, 8, 16, 32, 64
Dense Layers	**0**, 1, 2, 3
Batch Size	8, 16, 32, **64**, 128
Epochs	10, 20, **50**
Dropout Rate	**0.0**, 0.1, 0.2, 0.3
Early Stopping Patience	1, **3**, 5, 10

Table 3: Hyperparameters tested for LSTM model

The model achieved an F1-score of 0,71.

5.3 PLM

PLMs are the state-of-the-art NLP models. They are trained on large corpora of textual data in a self-supervised way. This provides a good initialization point for any downstream task, leveraging the pre-trained weights as better-than-random initializers. The models were further trained on the training portion of the dataset while using the dev portion as validation data. All hyperparameters which were fine-tuned are shown in Table 4. The best model achieved an F1-score of 0,80 with the parameters highlighted in bold in the table using RoBERTa (Conneau et al., 2019).

Notably, DistilBERT achieved F1 scores which were near identical to the two bigger models. However since efficiency is not a concern for this task in this experiment, the best performing model was chosen.

5.4 Model explainability

It is possible that due to flaws in training data, the model isn't actually finding veiled threats or offense and that it is simply identifying profanities. To test this hypothesis and answer research question 1, an experiment was conducted where the words of each input were randomly shuffled. This can show whether context actually matters or whether the model simply looks for profanities regardless of context. The resulting F1-score of 0,78 (averaged over three random seeds) shows that the hypothesis is indeed correct.

Secondly, knowing that the word order doesn't matter allows further exploration. If a word is removed from the tweet, it might change its label from OFF to NOT. In this case, it is clear that the model considers this word offensive and that the tweet is considered not offensive if this profane word is removed. Using this technique, it was possible to find a list of words which have a high probability of making a tweet offensive.

To achieve this, each tweet t from the dev portion of the dataset was split into n words, then n new tweets t_n were generated by removing the nth word from t. The best performing RoBERTa model was trained on the train set. The model was then used to predict a label for t, as well as each of t_n. If the predicted label of t was OFF, but the label of any of t_n was NOT, the nth word of t was considered an offensive word. This procedure was repeated for the entire dev set. In the end, the number of times each word caused a OFF → NOT switch was calculated and normalized by dividing by the total number of occurrences of that word. The result is a probability p for each word, which represents the likelihood that removing that word from a tweet makes the tweet no longer offensive.

Upon inspection of the list of words with $p > 0.15$, the complete contents of which can be seen in section 8. The threshold was picked by inspecting the data, whence it was found that non-profane but common words like "@USER" appeared with $p \leq 0.10$. As expected, the majority of the words are simple profanities. There are some words which are not profane marked by this approach like "yet" or "job", but the majority are profanities. This results shows that the model really is looking mainly for profanities and not really marking veiled offense or multi-word insults.

To test this hypothesis further, this list of words was used for brute-force classification by simply checking whether any word form the list is contained in a tweet. On the unseen test set (as this list of words was generated using the train and dev sets only) this approach achieves an F1-score of 0,66. This outperforms the list of profanities ob-

Parameter	Values
Model Name (lm)	**`"roberta-base"`**, `"bert-base-uncased"`, `"distilbert-base-uncased"`
Learning Rate	1e-6, **5e-6**, 1e-5, 3e-5, 5e-5
Batch Size	8, 16, 32, 64, **128**
Epochs	2, 3, **4**
Max Length	32, **64**, 128, 256

Table 4: Hyperparameters and PLMs tested

tained from (Anger, 2023) by 0,05 while being approximately twelve times shorter.

An attempt was made to replicate the same experiment using bigrams, but the results did not prove useful as most of the resulting bigrams simply consisted of a profanity combined with a random word. This was not a surprising result as in the SVM experiment, it was found that adding bigram or even trigram features to the data did not improve the performance.

Answering research question 2, it was found that the model mainly focuses on individual profanities and seemingly does not consider context or indirect offensive language such as paraphrased profanities or otherwise concealed offense. This is most likely to be caused by inadequate training data, as the data does not allow the model to learn more complicated features which constitute these kinds of offensive language.

6 Discussion

The results show that the PLMs performed best on the task at hand. The best PLM found with an F1-score of 0,80 performs in the upper range of the results achieved in the competition (Zampieri et al., 2019b). Specifically, it places around the 10th position. As expected, classical models and LSTM-based models performed well, but not at the level of PLMs in this task.

The competition as well as the results of this paper using the OLID dataset notably achieved much lower results than similar work with different datasets as described in section 2. Since the task is essentially identical, this can be attributed to the dataset. As mentioned in section 3, the data contains noticeable inconsistencies. This is further proven by the answer to research question 1, which shows that the context of the data is ignored by the model in favor of scanning individual words. The answer was confirmed by averaging the result with three different random seeds, which showed a variance of 0,01 in the resulting F1 scores. As expected, the list of offensive words generated by the model in response to research question 2 is simply a list of profanities.

This is confirmed by the best performing LSTM model. The model uses a single LSTM unit, suggesting that this is a very simple task and larger LSTM models lead to overfitting. This suggests that higher performance is not achievable due to inherent flaws in the data, not due to the model's distinguishing power.

Overall, the results point to a flaw in the dataset as the achieved performance is not on par with similar tasks. This can be caused by inherent bias in the data due to the way it is sampled, label inconsistencies discussed earlier or other factors. The results show that PLMs should be considered for use as automatic moderation tools once better training data is available.

7 Conclusion

The experiment was successful in finding a well-performing ML model for the task at hand and answering the posed research questions. The best performing model was a fine-tuned RoBERTa PLM with a macro average F1-score of 0,80. The main findings are that in this task, the model focuses mainly on profanity and does not take into account context or other nuanced clues as the the offensiveness of a tweet. The results also take a step towards explaining how the model makes its decisions in producing a list of words which have a high probability of causing a tweet to be offensive according to the model.

Future research should focus on finding more advanced data gathering techniques which can deal with the inherent class imbalance while avoiding bias and providing valuable data. It is difficult to find data which contains more nuanced offensive language which could be used by PLMs to learn to recognize it. It is important to avoid sampling bias while searching for ways to reduce class imbalance. With better training data, models like the one presented in this paper are sure to improve their performance and to become better suited for automatic social media moderation.

References

[Abadi et al.2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Anger2023] Zac Anger. 2023. profane-words. https://github.com/zacanger/profane-words, aug. Accessed: 2024-10-31.

[Bird et al.2009] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with python*. O'Reilly Media, Sebastopol, CA, July.

[Chawla et al.2002] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June.

[Chinivar et al.2023] Sneha Chinivar, Roopa M.S., Arunalatha J.S., and Venugopal K.R. 2023. Online offensive behaviour in socialmedia: Detection approaches, comprehensive review and future directions. *Entertainment Computing*, 45:100544, March.

[Chollet and others2015] François Chollet et al. 2015. Keras. https://keras.io.

[Conneau et al.2019] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

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

[Hajibabaee et al.2022] Parisa Hajibabaee, Masoud Malekzadeh, Mohsen Ahmadi, Maryam Heidari, Armin Esmaeilzadeh, Reyhaneh Abdolazimi, and James H Jr Jones. 2022. Offensive language detection on social media based on text classification. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, page 0092–0098. IEEE, January.

[Hochreiter1997] S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

[Lemaître et al.2017] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

[Liu et al.2019] Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In Jonathan May, Ekaterina Shutova, Aurelie Herbelot, Xiaodan Zhu, Marianna Apidianaki, and Saif M. Mohammad, editors, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

[May et al.2019] Jonathan May, Ekaterina Shutova, Aurelie Herbelot, Xiaodan Zhu, Marianna Apidianaki, and Saif M. Mohammad, editors. 2019. *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

[Pedregosa et al.2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[Sanh et al.2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

[Wadud et al.2022] Md. Anwar Hussen Wadud, Muhammad Mohsin Kabir, M.F. Mridha, M. Ameer Ali, Md. Abdul Hamid, and Muhammad Mostafa Monowar. 2022. How can we manage offensive text in social media - a text classification approach using lstm-boost. *International Journal of Information Management Data Insights*, 2(2):100095, November.

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

[Yin and Zubiaga2022] Wenjie Yin and Arkaitz Zubiaga. 2022. Hidden behind the obvious: Misleading keywords and implicitly abusive language on social media. *Online Social Networks and Media*, 30:100210, July.

[Zampieri et al.2019a] 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.

[Zampieri et al.2019b] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In Jonathan May, Ekaterina Shutova, Aurelie Herbelot, Xiaodan Zhu, Marianna Apidianaki, and Saif M. Mohammad, editors, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

8 Appendix

The rest of this page is intentionally left blank.

Profane Words Identified by RoBERTa				
dumb	fuck	Despicable	our””	sucks
(us)	commies	precious	atheists	anyone
horrible	chicks	lying	fucking	damn
suck	ass	Demon-craps	Fucked	shit
messed	rape	slaying	down.	intimidate
evil	outfit.	corrupt	streets	fucked
liberals.	down?	job	lies	F*******
Flawed	overgeneralization.	crazy	religion.	Fascists
idiotic	sick	hell	bastards	stupid.
LIES!!....They’re	morons.	brat	Ass	ASS
lie.	fool!!!!	idiocy	embarrassment...	mouth
trashpeople.	stupid	terrible.	wilful	violence.
ignorance	Asshole!	SHITlibs	phony	liar
suicide	raped	witch	traitor	slurs
shame	slanderous	bullshit	fool	too?
produce	criminals	devils.	butt	FUCK
hatred	bullies.	maniac!	shitheads.	guy.
Chicago	Dems	USA.	country,	saw
capital	Dick,	yet	control,	peewee
Herman...	weird	dumbest	vile	BITCH?””””
disgusting.	abuse	sexual	mindless	fbi
cocks!	pissing	fuckwits.	stupidity	pussy
bitch	idiot!!!	murdered	laws.	blames
crap?	racism	Hypocrisy	shit?	nuts
find	hate.	hateful	disgusting	liar!
piss	bully	hated	Fuckin	jester
f*ck	awful!	confederate	Toad	#MoronsAreGoverningAmerica
choking	idiot	bloody	murder	dealing
boyfriend	drug	Burger	worked	a**hole.
Cowards.	shitting	fraud	jail	BADASS!
idiot.	assholes	Maddow’s	toxic	slut
Fuck	lunatic	bald	crush	Human
Illegal	nut	mongering	coward	criminal
Utter	garbage.	hypocrite	prostitute...for	stupid,
nasty	brats	joke	cruel	”...
American	epithet	Criminal	NUTS	daddy
Fredo.	prison.	Pathetic	Lying	trashy
men?	idiots	collapse	embarrassment	porn
f*cking	#Bullshit	Gays?	FUCKING	DIRTY
DESPERATE	narcissist.	Impeach	life””	pedo.
hates	pussy””	deep	pay	#fanniegate
Biglly!	crooks?	#LockThemAllUp	Got	guilty
devil””.”””	garbage	gypsies	midget	poor
#traitor	#Islamists	cunt.	crimes	LIAR,
fuckin	cowards.	Liars	Pussy	

Table 5: Profane Words Identified by RoBERTa