NLPR@SRPOL at SemEval-2019 Task 6: Linguistically enhanced deep learning offensive sentence classifier

Alessandro Seganti[1], Helena Sobol[1], Iryna Orlova[1], Hannam Kim[2], Jakub Staniszewski[1], Tymoteusz Krumholc[1], Krystian Koziel[1]
[1] Samsung R&D Institute Poland
[2] Samsung Electronics, Korea
a.seganti@samsung.com

Abstract

The paper presents a system developed for the SemEval-2019 competition Task 5 *hatEval* Basile et al. (2019) (team name: *LU Team*) and Task 6 *OffensEval* Zampieri et al. (2019b) (team name: *NLPR@SRPOL*), where we achieved 2^{nd} position in Subtask C. The system combines in an ensemble several models (LSTM, Transformer, OpenAI's GPT, Random forest, SVM) with various embeddings (custom, ELMo, fastText, Universal Encoder) together with additional linguistic features (number of blacklisted words, special characters, etc.). The system works with a multi-tier blacklist and a large corpus of crawled data, annotated for general offensiveness. In the paper we do an extensive analysis of our results and show how the combination of features and embedding affect the performance of the models.

1 Introduction

In 2017 two-thirds of all adults in the United States have experienced some form of online harassment (Duggan, 2017).[1] This, together with various episodes of online harassment, boosted research on the general problem of recognizing and/or filtering offensive language on the Internet. Still, recognizing if a sentence expresses hate speech against immigrants or women, understanding if a sentence is offensive to a group of people, an individual or others – these tasks continue to be very difficult for neural networks and machine learning models to accomplish. In order to do this, various implementations have been proposed; for the most successful recent approaches see Pitsilis et al. (2018); Founta et al. (2018); Wulczyn et al. (2017); Waseem and Hovy (2016); Park and Fung (2017); Davidson et al. (2017). Most of them use various combination of features to recognize these characteristics.

This article presents a system that we have implemented for recognizing if a sentence is offensive. The system was developed for two SemEval-2019 competition tasks: Task 5 *hatEval* "Multilingual detection of hate speech against immigrants and women in Twitter" Basile et al. (2019) (team name: *LU Team*) and Task 6 *OffensEval* "Identifying and categorizing offensive language in social media" (Zampieri et al., 2019b) (team name: *NLPR@SRPOL*). Table 1 shows the results that we achieved with our system in the SemEval-2019 competitions.

Competition	Placement
Task 6-A	8^{th} position
Task 6-B	9^{th} position
Task 6-C	2^{nd} position
Task 5-A	8^{th} position (ex aequo)

Table 1: SemEval-2019 results.

In order to create a highly accurate classifier, we combined state-of-the-art AI with linguistic findings on the pragmatic category of impoliteness (Culpeper, 2011; Jay and Janschewitz, 2008; Brown and Levinson, 1987). We achieved this by deciding on the factors that point to the impoliteness of a given expression (for the blacklists) or the entire sentence (for corpus annotation). Such factors led us to divide the blacklist into "offensive" and "offensive in context", as most linguistic studies of impoliteness focus on various aspects of the context. Furthermore, linguistic research made it possible to arrive at a maximally general definition of offensiveness for the crowdsourced annotators.

[1] Due to the topic of the SemEval-2019 Tasks 5 and 6, the present paper contains offensive expressions spelled out in full. These are solely illustrations of the problems under consideration. They should not be interpreted as expressing our views in any way.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 712–721
Minneapolis, Minnesota, USA, June 6–7, 2019.

The article is organized as follows. Section 2 presents the current state of the art for offensive sentence classification. Section 3 explains the architecture of our system (features, models and ensembles). Section 4 describes the datasets and how they were created. Section 5 shows the results of the SemEval-2019 tasks in detail, motivating which combination of features and models was the best. Finally, section 6 offers conclusions together with our plans for future research.

2 Related work

In recent years, the problem of recognizing if a sentence is offensive or not has become an important topic in the machine learning literature. The problem itself has different declinations depending on the point of view. Currently there are three main areas of research in this topic in the machine/deep learning community:

1. Distinguishing offensive language from non-offensive language;
2. Solving biases in deep learning systems;
3. Recognizing more specific forms of offensivness (e.g. racism, sexism etc.).

The main problem with each of the tasks is the amount of data available to researchers for experimenting with their systems. This – together with the fact that it is difficult to clearly define what is offensive/racist/sexist or not – makes the three problems listed above very difficult for a deep learning system to solve.

Articles have showed that there is a strong bias in text and embeddings, and have tried to solve this bias using different techniques (Zhao et al., 2018; Dixon et al.; Bolukbasi et al., 2016). Furthermore, thanks to a dataset defined in Waseem and Hovy (2016) and Waseem (2016), various works have gone in the direction of recognizing sexism and racism in tweets (Pitsilis et al., 2018; Park and Fung, 2017).

Another field of work was recognizing offensiveness in the Wikipedia internal discussion forum dataset (Wulczyn et al., 2017). This dataset has led to other articles making systems for distinguishing between offensive and non-offensive language (Founta et al., 2018; Pitsilis et al., 2018; Kumar et al., 2018; Gröndahl et al., 2018; Li, 2018; Park and Fung, 2017; Aken et al., 2018).

Linguistic expertise enhanced the functionality at two stages: sentence annotation (described in detail in Section 4) and active creation of blacklists (Section 3). The completion of these tasks breaks new ground, as there exist no corpus linguistic studies on the generality of offensive language, to the best of our knowledge. Recent approaches of narrower scope are Dewaele (2015) and McEnery (2006).

3 System description

Our system is composed of three major elements, described below:

- Features – common to all models;
- Various models – neural networks or not;
- Ensemble.

3.1 Features

This section describes the features that we used and explains their role. We implemented the following features:

- Number of blacklisted words in the sentence;
- Number of special characters, uppercase characters, etc.;
- A language model taught to recognize offensive and not offensive words.

Blacklisted We used two kinds of blacklisted expressions: "offensive" and "offensive in context". The "offensive in context" expressions are offensive in specific contexts and unoffensive otherwise, e.g. *bloody* or *pearl necklace*. This dictionary was compiled by crowdsourcing and contains about 2,300 words (+ variations). The blacklist consists of swear words, invectives, profanities, slurs and other impolite expressions.

Special characters, uppercase, etc. We checked the graphemic characteristics of the written text and we gave this as a feature to the model. We mainly used the non user related features defined in Founta et al. (2018).

Language model Inspired by the work of Yu et al. (2018), we decided to train a language model on both offensive and non-offensive words. For this purpose, we trained two character based language models, one on the offensive dictionary (described above) and the other from a corpus of non-offensive words. After training them we used the difference in perplexity of each input word as a feature for the model.

3.2 Models

We trained various models and then combined them in an ensemble. This section outlines the models that were part of the ensemble.

Embeddings Both the Neural networks and the machine learning models used embeddings. We used the following embeddings: ELMo (Peters et al., 2018), fastText (Bojanowski et al., 2017), custom embeddings, and Universal Sentence Encoder (Cer et al., 2018). For fastText, we used the 1 million word (300d) vectors trained on Wikipedia 2017, below called fastText 1M.

The custom embedding was built by training a fastText embedding on our corpus. We then combined the 1M fastText embeddings with these custom embeddings using Truncated SVD after concatenating their columns (this was done inspired by the work (Speer et al., 2017)). Building custom embeddings was important for the offensive word classification because the original version of the fastText 1M embeddings contained around 50% of the words in the corpus while after adding the custom embeddings, only 30% of the words were out of the vocabulary. Below, this combination of embeddings is called "combined".

Neural networks We used two types of neural network models:

- LSTM models (Hochreiter and Schmidhuber, 1997);
- Transformer models (Vaswani et al., 2017).

For both models, we used multi-head attention and we tried different embeddings. In most cases, the Transformer models had better results than the LSTM models, and this is what we used in the submissions. The parameters of the models are described in Appendix A.1. In both models, the Features described in Section 3.1 are concatenated to the output of the model.

OpenAI GPT One of the models that we used was the OpenAI GPT (Radford et al., 2018). We used the GPT model in its original form, without changing any parameters. Our results show that this model works very well when there is enough data for finetuning. However, small classes – as in Task 6 Subtask C – pose a problem (see Section 5).

Machine learning models We used two machine learning models:

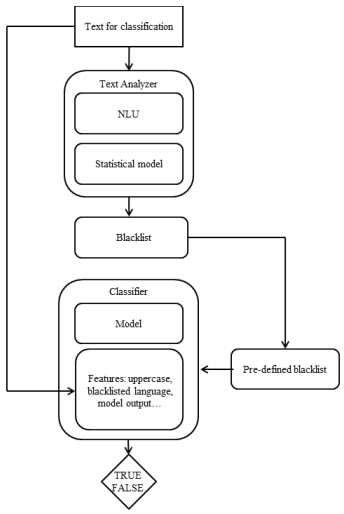

Figure 1: Pipeline for the offensive sentence classifier.

- Random forest;
- SVM.

For these models, we built a pipeline where:

- In a first step we either compute the embeddings of the sentences or get the Td-Idf score.
- In a second step we concatenate the result of the first step with the Features described in Section 3.1 (if used).
- We run the classifier.

As embeddings we used only the Universal Encoder, and with good results.

Ensemble For the ensemble we used a voting classifier with soft voting (based on the probability returned by each model). For each subtask, we show which combination of models gave the best results.

The pipeline for the entire offensive sentence classifier is shown in Figure 1.

4 Data/Datasets

4.1 Preprocessing

Preprocessing plays a crucial role in the analysis of potentially offensive sentences, because most inputs use highly non-standard language. Hence, preprocessing was mainly focused on normalizing the language for simplifying the model work. We applied the following preprocessing:

- Substituting user names with <USER> tokens;
- Removing all links;
- Normalizing words and letters;

- Normalizing spacing and non-standard characters;
- Over/Downsampling of the classes;

After the preprocessing, we split by space and used each token as an input to the models.

Normalizing words and letters We have a dictionary containing common spelling variants of words found in our corpus. We used this to change words to the "canonical" form. Examples of such variants can be seen in Table 2.

Word	Common variants
fuck	fvck, fok, fucc, phuk
nigger	n1gga, n1gr, niigr, nuggah, nigg3r
boob	booob, booooooob
motherfucker	Mutha Fukker, Motha Fuker
ass	a55, 455 ("leetspeak" variants)
assclown	ãs̈s̈ĺσẇη (vulgarity obfuscation)

Table 2: Common spelling variants.

Over/Downsampling For each Task/Subtask, we systematically oversampled the classes to obtain a balanced dataset. This was especially important for Task 6/Subtask C, which introduced 3 highly unbalanced classes. For most subtasks we did two things at the same time:

- Downsampled the majority class when there was too much difference from the other classes;
- Oversampled the minority class after downsampling.

4.2 Datasets

In this section we give a high level overview of the datasets we used for training our models for the SemEval-2019 tasks. Detailed statistics are presented in Appendix A.2. For training the model, we used several openly available datasets:

- *Hate Sonar* gathered from Twitter (Davidson et al., 2017);
- 2 related hate speech datasets from Twitter (Waseem and Hovy, 2016), (Waseem, 2016);
- Insulting internet comments (Impermium, 2012);
- Attacking, aggressive, toxic and neutral comments from Wikipedia Talk Pages (Wulczyn et al., 2017);
- *Vulgar Twitter* (Cachola et al., 2018);

our own custom-built corpora and datasets provided by the SemEval organizers.

From the sources listed above, we added a total of 20,399 sentences to the SemEval-2019 corpus for Task 5, and 97,759 sentences to the one for Task 6.

Custom Offensive language corpus Our custom dataset was built by crowdsourcing and by crawling content from the Internet. The dataset is balanced, with 49,179 not offensive and 48,580 offensive comments. Around half of the dataset was labeled by linguists, who were asked to look for "general offensiveness". This could take various forms:

- Expletives, swear-words, offensive terms;
- Rude meaning;
- Meaning that is harsh politically/ethically/ emotionally, and hence expression of hate/ disgust/disrespect;
- Uncomfortable topics related to the human genitals in a gross way;
- Hate speech, sarcasm, sexism, racism, violence, etc.;
- Discussion of drug use or other illegal actions;
- For any other reasons, children should not have access to the sentence.

To each sentence, the linguists assigned one of the three labels:

- OFF – offensive sentence,
- NOT – not offensive sentence,
- Nonsense – random collection of words or non-English (removed from the corpus).

In cases of disagreement between linguists, we chose the most popular label, if applicable, or obtained an expert annotation. We calculated Fleiss' kappa for inter-annotator agreement (Fleiss, 1971), which extends Cohen's kappa to more than two raters (Cohen, 1960). For random ratings Fleiss' $\kappa = 0$, while for perfect agreement $\kappa = 1$. Our κ was equal to 0.62, which falls in the "substantial agreement" category, according to Landis and Koch (1977).

The remaining part of the corpus was assessed automatically with a blacklist-based filter.

Dataset for Task 6 The OLID dataset (Zampieri et al., 2019a) contains Offensive and Not Offensive sentences. The Offensive sentences are further categorized into:

- TIN – targeted insults and threats,
- UNT – untargeted.

and the targeted (TIN) category was further subdivided into:

- IND – individual target,
- GRP – group target,
- OTH – a target that is neither an individual nor a group.

Our full offensive language corpus, described in the previous subsection, was used for this task. The OFF sentences were further annotated for the two categories while the NOT sentences were not further annotated. All the additional classes were added automatically by a wordlist-based annotator.

Dataset for Task 5 The dataset for Task 5 (Basile et al., 2019) contained the classes:

- HATE – hate speech against women or immigrants,
- NOHATE – no hate speech against women or immigrants.

together with other subclasses. Given that we participated in the Task 5 Subtask A, we annotated our corpus only with these two labels. Using a mixture of automated and manual annotation, we were able to add around 30k sentences from our dataset for this task.

5 Results

SemEval In Table 3 we show the average F1 of our models for all the SemEval-2019 Tasks and Subtasks. These results were obtained by using an ensemble of models and in Table 4 we show which model was used inside which ensemble. The acronyms used in the table correspond to:

- **GPT** : OpenAI's GPT model
- **RF**: Random Forest
- **T**: Transformer model
- **U**: Universal encoder
- **EL**: ELMo embeddings
- **CO**: Combined embeddings (see 3.1 for an explanation of this)
- **F**: Features.

Given the short amount of time, during the SemEval competition we were unable to test all the combinations of models and data preparation types to choose the best combination for the Ensemble. We thus selected the models in the ensemble by experimenting with part of the models. This is the main reason why only one model used in Task 5 contains additional features (the TELUF model).

Ensemble	Competition	Macro F1
6-A	Task 6-A	0.80
6-B	Task 6-B	0.69
6-C	Task 6-C	0.63
5-A	Task 5-A	0.51

Table 3: SemEval-2019 results breakdown.

Task	6-A	6-B	6-C	5-A
GPT	✓		✓*	✓
RF	✓	✓	✓	✓
RF+U	✓	✓		✓
T+EL	✓		✓	✓
T+CO+U	✓		✓*	✓
T+EL+U+F				✓

Table 4: Ensemble detail. The models marked with * have been trained with an unbalanced dataset.

After the competition, we tried the models contained in the Ensembles on all the tasks; detailed results are presented in Table 9 in the Appendix. It is important to note though that the results in the Appendix cannot be directly compared with the ones of the SemEval competition because although the models were the same, the Test data was different (the golden data has not been released yet).

From the results we clearly see that we have two "data regimes": in the *low data regime* (Task 6 Subtask B and C), Random forest (with or without the Universal embeddings) is the best choice. However, in *the big(ger) data regime*, Fine tune is the best model. Also in the low data regime each model works best with a different data preparation strategy: GPT with unbalanced data, the Transformer with oversampled and downsampled data while Random forest with oversampled data.

Ablation studies In this part we show the results of ablation studies on the transformer and random forest models. In this study, we want to understand how far the final result was influenced by the linguistically based features and preprocessing we defined in this article. All the results obtained in this section have been computed on a Test set

Model	Task 6 A	Task 6 C
T + CO	0.73	0.44
T + CO + U	0.71	0.52
T + CO + F	**0.75**	0.45
T + CO + U + F	0.74	0.47
RF	0.7	**0.54**
RF + F	0.68	0.43
RF + U	0.72	0.48
RF + U + F	0.69	0.38

Table 5: Macro F1 for selected Transformer models with different combinations of features

created from the Train set shared in the SemEval-2019 tasks (as in Appendix A.3). As we discussed in the previous section, the Tasks were characterized by a "low" (Task 6 C) and a "big" data regime (Task 6 A), thus we compare the ablation study results for these two extreme regimes.

In a first study we wanted to understand how the features influenced the results. For this reason, we tried some combinations of Features, Embeddings and Models on both Task 6 Subtask A and Task 6 Subtask C; the relevant macro F1 results are shown in Table 5. The table shows that, in the big data regime, the Random Forest works best when only the Universal Encoder is used, while the Transformer model improves its performance when the features are added. On the other hand, in the low data regime, we see that the plain Random Forest outperforms all the other combinations. This is probably because the more things we add, the more the model needs to learn, and with little data this is simply not possible.

In a second study we wanted to understand how much the normalization defined in Section 4.1 affected the performance of the model. For this reason, we trained again the best models in Table 5 for both Subtasks with an unnormalized version of the dataset. The results are that for Subtask A, the model **T + CO + F** F1 decreased from 0.75 to 0.73 while for Subtask C, the **RF** F1 decreased from 0.54 to 0.44.

The results of this section seem to point to the fact that the features we added and the normalization we used are beneficial for the performance of the models. Further work will be devoted to understanding this point though.

6 Conclusions

The article presented our approach to making a classifier recognizing offensive expressions in text. It showed how our architecture is suitable for multiple (related) offensive sentence classification tasks. It also showed how we built the features and the data that the model used for learning. Thanks to our system, we were 2^{nd} in the SemEval-2019 Task 6/Subtask C. In the article we also showed with ablation studies that the linguistic features proposed and the embeddings added improve the performance of the models we used.

In the future, we will extend our system to recognize a wider set of features. We are currently working on analyzing the linguistic differences between the offensive corpus and the non-offensive corpus. Specifically, we think that by analyzing the differences, we should be able to build a "white-list" of terms that can be used as features that will help the classifier understand which sentences are less likely to be offensive.

Acknowledgments

We would like to thank G. Knor, P. Przybysz, P. Andruszkiewicz, P. Bujnowski and most of the AI Team in Samsung R&D Institute Poland for all the helpful discussions that made this article possible.

References

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. arXiv:1809.07572.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Association for Computational Linguistics, location = Minneapolis, Minnesota.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some Universals in Language Usage*.

Isabel Cachola, Eric Holgate, Daniel Preoiuc-Pietro, and Junyi Jessy Li. 2018. Expressively vulgar: The socio-dynamics of vulgarity and its effects on sentiment analysis in social media. In *Proceedings of the 27^{th} International Conference on Computational Linguistics*, pages 2927–2938.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. arXiv:1803.11175.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Jonathan Culpeper. 2011. Politeness and impoliteness. In *Pragmatics of Society*, pages 391–436.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*, pages 512–515.

Jean-Marc Dewaele. 2015. British *bollocks* versus American *jerk*: Do native British English speakers swear more – or differently – compared to American English speakers? *Applied Linguistics Review*, 6(3):309–339.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

Maeve Duggan. 2017. Online harassment 2017.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Antigoni-Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athina Vakali, and Ilias Leontiadis. 2018. A unified deep learning architecture for abuse detection. arXiv:1802.00385.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All you need is "love": Evading hate speech detection. arXiv:1808.09115.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Impermium. 2012. Detecting insults in social commentary.

Timothy Jay and Kristin Janschewitz. 2008. The pragmatics of swearing. *Journal of Politeness Research*, 4:267–288.

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*, pages 1–11.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Siyuan Li. 2018. Application of recurrent neural networks in toxic comment classification. Master's thesis, University of California, Los Angeles.

Tony McEnery. 2006. *Swearing in English. Bad language, purity and power from 1586 to the present*. Routledge, London and New York.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT 2018*, pages 2227–2237.

Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. arXiv:1801.04433.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 4444–4451.

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*, pages 5998–6008.

Zeerak Waseem. 2016. Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the* 26^{th} *International Conference on World Wide Web*, pages 1391–1399.

Xiaodong Yu, Stephen Mayhew, Mark Sammons, and Dan Roth. 2018. On the strength of character language models for multilingual named entity recognition. In *Proceedings of the 2018 conference on empirical natural language processing (EMNLP 2018)*.

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. In *Proceedings of NAACL*.

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 *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853.

A Supplementary Materials

A.1 Model parameters

The following parameters were used for all LSTM and Transformer models in the results Section 5:

- keep probability: 0.8;
- LSTM units: 100;
- L2 regularization: 0;
- fully connected size: 256 or 128;
- multihead attention:
  - attention size: 5 or 2,
  - attention head: 4

For GPT, we used a learning rate of 6.25e-5 and an L2 regularization of 0.01.

A.2 Data

In Table 6 we show the amount of data that was contained in our corpus (overall). In Table 7 and 8 we show the data for Task 5 and Task 6. For a description of how these corpora were built and annotated, see Section 4.2.

Source	NOT	OFF	Total
Custom corpus	16,545	12,938	29,483
Kaggle	2,629	3,463	6,092
Twitter	917	23,438	24,355
Wikipedia	29,088	8,741	37,829
Total	49,179	48,580	97,759

Table 6: Statistics for our offensive language corpus. The Kaggle dataset was collected by Impermium (2012). The Twitter dataset was compiled from 4 sources: Davidson et al. (2017), Cachola et al. (2018), Waseem and Hovy (2016) and Waseem (2016). The Wikipedia dataset was collected by Wulczyn et al. (2017).

Class	Total
HATE	16,508
NOHATE	11,154

Table 7: Statistics for the additional corpus for SemEval-2019 Task 5.

A.3 Model results

In this section we show the detailed results of all the models for all the SemEval-2019 tasks. For each Task, we extracted a test set from the Train data released by SemEval. We compared the models to one of the current state of the art defined in Park and Fung (2017); the results shown here are obtained by averaging the best F1 for each class (not a single model). The data by Waseem and Hovy (2016) for comparing to the state-of-the-art model has been kindly shared by the authors of Park and Fung (2017). In the table we marked with

Class	Targeting	Target	Total
OFF	TIN	IND	18,506
		GRP	6,761
		OTH	1,025
		Total	34,669
	UNT	–	6,234
	Total	–	59,837
NOT	–	–	64,773

Table 8: Statistics for the additional corpus for SemEval-2019 Task 6.

- No additional mark: the normalized data with oversampling and downsampling as described in Section 4.
- **FULL**: the normalized data with oversampling but no downsampling.
- **UNB**: the normalized data without oversampling or downsampling.

The model acronyms are the same as the ones used in Section 5.

Model	5-A	6-A	6-B	6-B FULL	6-C	6-C FULL	6-C UNB	SOTA
RF	0.7	0.62	**0.61**	0.58	0.44	**0.54**	0.45	0.78
RF + F	0.68	0.68	0.59	0.54	0.32	0.43	0.41	-
RF + U	0.72	0.69	0.6	0.55	0.39	0.48	0.46	0.74
GPT	**0.77**	**0.77**	0.58	**0.6**	0.42	0.49	**0.51**	**0.81**
T + CO + U	0.74	0.71	0.58	**0.6**	**0.52**	0.45	0.5	0.73
T + EL	0.73	0.73	0.58	0.58	0.49	0.5	0.45	0.74
SOTA	-	-	-	-	-	-	-	0.78

Table 9: Macro F1 for all the models on all the Tasks and on the state-of-the-art (SOTA) data.