Supplementary Notes to *Inducing a Lexicon of Abusive Words*

Michael Wiegand Josef Ruppenhofer Anna Schmidt
Clayton Greenberg

February 15, 2018

1 Introduction

This document provides more detailed information regarding certain aspects of our research for which there was not sufficient space in the main paper. We focus on four aspects:

- the creation of the base lexicon (§3), particularly, the annotation carried out via crowdsourcing
- the annotation of microposts with regard to explicitness (§4)
- the classifier configurations in our experiments on word-level classification (§5)
- the classifier configurations in our experiments on classifying microposts (§6), including a detailed discussion how we reconstructed the feature set from Nobata et al. [2016] (§6.2)
- some etymological information about the three recent abusive words mentioned in the introduction of the main paper (§7)

First, we begin with some general remarks on our annotation based on crowdsourcing.

2 Some General Remarks on Our Crowdsourcing Annotation

All our manual annotation for this research was produced with the help of crowdsourcing using *Prolific Academic*.[1] In *all* experiments, we had each instance rated by 5 different annotators. The most important requirements we specified for annotators in order to be eligible to participate in our survey were the following:

- Each annotator had to be a native speaker of English.

[1] `www.prolific.ac`

- Each annotator had to have a task approval rate of 90% or higher. (At the time we ran our experiments, this was the highest threshold possible on Prolific Academic.)
- Each annotator was not allowed to have any literacy difficulties.

For each annotation task we also produced some annotation guidelines (*they are also included in the supplementary data*). Annotators had to read those guidelines before the actual annotation. The guidelines were also made available as a separate pdf-file, so that the annotators could always refer to them during their annotation.

All guidelines were illustrated with examples. We also used these examples as test instances. They were randomly interspersed with the regular instances to be annotated. However, they were not marked as such so that the annotators could not recognize them. We only informed them in the guidelines that we had included test instances. The order of the instances presented to the annotators was randomized. If a significant proportion of test instances was incorrectly labeled by some annotator, then this meant that the annotator either had not read the annotation guidelines or had not understood the annotation task. Subsequently, the annotation of this annotator was rejected and not included in our final gold standard.

In all annotation tasks, the set of instances was organized in bins of about 100 to 200 instances. Each bin was considered a separate task posted on *Prolific Academic*. We felt that a larger amount of instances would result in a loss of concentration on the part of the annotators.

In order to have the most unbiased annotation possible, each annotator from *Prolific Academic* was only admitted to one particular annotation task of our work. For example, an annotator admitted to the task of annotating the abusive base lexicon would not have been allowed to participate in the annotation of evaluating explicitly abusive contexts and vice versa.

3 Creation of the Base Lexicon

The annotation of the base lexicon was done in two steps. In the first step, all words of the lexicon were judged (§3.1). In a second step, a subset of words was re-annotated in order to rule out inconsistent labeling (§3.2).

3.1 Main Annotation

Although the main annotation task was to decide for each given polar expression whether it was considered abusive or not, we also asked each rater to produce for each abusive word a small example sentence in which the given word is used to convey an abusive remark. These sentences were not used in our subsequent experiments. They were just meant as a further sanity check to test whether the annotators really had understood the task. For instance, if annotators produced sentences, which we felt were definitely not abusive or the abusive nature was conveyed by another word in that sentence than the given polar expression to be rated, then we rejected the annotation of these annotators and their annotation was not included in our final gold standard.

We chose to only consider words as abusive if 4 out 5 annotators were of the same opinion. We favoured this high threshold since we were interested in a list of (mostly) unambiguous abusive words. In the light of existing word lists of abusive words suffering from containing too many ambiguous entries [Davidson et al., 2017], we consider this decision vital. A list of such reliable words is also a pre-requisite of successfully inducing a larger lexicon of abusive words. The lower the precision of the initial base lexicon, the more likely noise will be added in the induction.

Since our base lexicon is fairly small (i.e. 1650 words), we could not afford to discard those words from the lexicon where several annotators (that is 2 or 3 annotators) considered a given word abusive. We accepted that our rating would produce some false negatives. However, in a second annotation step §3.2, we could systematically relabel a substantial amount of these instances.

3.2 Revision

By manually inspecting the labels of the annotation (§3.1), we found many words that are semantically similar to each other but were not assigned the same class label. In order to rule out annotation inconsistencies, we ran a second elicitation study specifically addressing these cases. We presented raters lists of similar words. (Each list typically comprised 3 to 5 instances.) These similar words always contained one word which had been assigned a class label other than the remaining words of that list. For example, we consider the 4 nouns *braggart*, *huckster*, *hypocrite* and *shyster* as semantically similar words, yet while *braggart*, *hypocrite* and *shyster* were labeled as abusive, *huckster* was not. The annotators were not given the labels of the individual words but the label of the majority of those similar words (in the above example the label given to the annotators would have been *abusive*). They were to specify the word on the list they thought does not correspond to the level of abuse that is conveyed by the other words on the list.[2] The words of that list were always presented in random order. The annotators also had the option to label that list as consistent. In that case, none of the given words was chosen as the odd one out. If 4 out of 5 annotators agreed on the odd one out (this should have coincided with the word that we initially thought as the word being assigned an incorrect class label), this would have confirmed that the original annotation was correct. If, however, there was no such agreement, then this would have meant that our initial intuition about the inconsistent label given in the initial annotation (in the above example it would have been the original label *non-abusive* assigned to *huckster*) was indeed inconsistent. In those cases, we replaced the label of the odd one out.[3]

This revision resulted in about 10% of the instances being re-labeled. We also examined the performance of our supervised classifiers trained on the labels before and after this revision. We found that overall classification benefited from this revision by approximately 2% points in F-score.

[2] If they thought that more than one word on that list notably deviated from the remaining words, they were to specify the word they considered least similar to the remaining words.

[3] There were only very few cases, in which the annotators agreed on the inconsistent labeling of a word on the similarity list that we did not initially have in mind. However, even in those cases, we would also have followed the judgment of the annotators.

4 Annotation of Microposts with regard to Explicit Abusiveness

Two of the four existing datasets containing annotated microposts that we consider in our work, that is the datasets from Warner and Hirschberg [2012] and from Waseem and Hovy [2016], were re-annotated with regard to explicit abusiveness. The results of our cross-domain experiments was that all classifiers produced lower classification scores on these two datasets including those trained on predictive word lists. After visual inspection we found that in those particular datasets, the amount of abusive utterances that comprise explicit abusive words were disproportionately lower than on the remaining datasets. Unlike the other two datasets, a substantial amount of abusive remarks were conveyed by sarcasm (1), metaphorical language (2), conspiracy theories (3), allusions (4), jokes (5) or displaying prejudices (6).

(1) #adviceforyoungfeminists Be sure to employ double standards to excuse your bigotry. No one will notice

(2) I would rather brush my teeth with sandpaper then watch football with a girl!!

(3) When Walt Disney died in 1966, the last barrier to the total Jewish domination of Hollywood was gone, and Jews were able to grab ownership of the company that Walt built. Since then they have had everything their way in the movie industry.

(4) We are in an era where everyone has come to understand what the real #Islam is all about.

(5) How does a black woman fight crime? She has an abortion.

(6) Islam tells women to stay at home.

Since we consider these utterances to be inaccessible for any cross-domain classifier, we tried to reduce those datasets so that the abusive remarks exclusively comprise explicit abusive words. We asked annotators from *Prolific Academic* to mark those abusive microposts that they consider to be explicit. An explicitly abusive remark should contain at least one abusive word. The annotators had to specify the abusive word in an explicitly abusive utterance. They were not given a full reference list of abusive words (this is actually the ultimate goal of this research) but only some reference sentences for implicitly abusive remarks and explicitly abusive remarks.

A micropost was considered as explicitly abusive if 4 out of 5 annotators judged so on that instance. For the new dataset, we still adhered to a binary classification scheme. A micropost is either explicitly abusive or not. We excluded the instances that were originally labeled as abusive but were not judged explicitly abusive in this elicitation study. The reason for excluding these instances is that we did not want classifiers to be penalized if they labeled an instance as abusive but this micropost was not judged as *explicitly abusive*. Maintaining these instances in the dataset would have meant that they would then be counted as a false positive (since only explicitly abusive instances are considered abusive). After removing these instances we adjusted the resulting

dataset in such a way that the original class distribution was preserved. In order to achieve this, we randomly removed non-abusive microposts until the original class distribution was restored.

Due to the large size of the dataset from Waseem and Hovy [2016], we only had annotators rate a sample of 1000 abusive microposts. However, we saw that these 1000 instances were representative of the entire sample of abusive remarks.[4]

5 Classifiers for Word-Level Experiments

5.1 Support Vector Machines (SVM)

For our word-level experiments, we chose SVM and used SVM^{light} in its ***default configuration***. Parameter tuning did not increase the overall performance on this particular dataset. This also meant that we did not have to set aside data for a dedicated development set. (Note that on the experiments not dealing with word-level data, i.e. §6, a development set was used.)

5.2 Processing Wiktionary

Several of our well-performing features were derived from *Wiktionary*. In order to automatically navigate through that resource, as an API we employ *JWKTL* [Zesch et al., 2008].

5.3 Weak Supervision

For our weak supervision baseline, in which we exclusively employed graph-based labeled propagation, we made use of the Adsorption propagation algorithm as implemented in *junto* [Talukdar et al., 2008]. For all different sets of seeds we experimented with, we only consider the ***default configuration*** of that tool.

With regard to the choice of seed class instances, we ran more extensive experiments on the negative class seed instances.[5] Apart from high-frequency words, we also examined

- a random sample of words *and*
- words that possess an *actor* sentiment view (see section on *sentiment views* in the main paper).

The latter option was motivated by the fact that among actor-view words on our base lexicon, there is a very low proportion of abusive words, i.e. 90.3% (Table 3 in the main paper). Thus, a priori a randomly extracted actor-view word is very unlikely to be abusive. Our exploratory experiments with all three types of negative class seed instances revealed that high-frequency words performed best. Due to the limited space in our main paper, we, therefore, only listed the performance using the most effective set of negative class seed instances.

[4] This was mostly achieved by forcing the two predominant subtypes of abuses on this dataset, that is, racism and sexism, to be equally represented on the final sample.

[5] For the positive class seed instances the choice of using the output of our surface patterns is self-evident. It is the only feature that does not rely on some lexical resource; its output can be directly considered as a set of abusive words. Other effective features, such as WordNet glosses, first need to be trained by a classifier in order to be used.

In our exploratory experiments, we, of course, compared negative class seed instances of the same size. As a size we arbitrarily chose the number of actor-view words that are available to us. Since we wanted to avoid overfitting this parameter value was also chosen in our final experiments on the amounts of high-frequency words for negative class seed instances.

5.4 Deriving Word-Level Classification from Labeled Microposts

For the projection-based approach to produce a ranking of negative polar expressions using labeled microposts (MICR:proj), we modified the code of the implementation of *Non-Linear Sub-Space Embedding (NLSE)* [Astudillo et al., 2015]. Our modification meant removing the hidden layer from NLSE and reducing the dimensionality of the projected embeddings to a scalar. All remaining components of that software were left to its ***default configuration***.

As both methods to produce a word-level classification derived from labeled microposts (i.e. MICR:pmi and MICR:proj) produce a ranking, we had to set empirically a cut-off value in order to convert the ranking into a binary classification. Since these two methods were only thought as baselines, we set the cut-off values to produce the best possible F-score.[6] For the PMI-based method (MICR:pmi), the ***cut-off value*** was set to 250, where as for the projection-based method (MICR:proj) the value was set to 400.

6 Classification of Microposts

6.1 Tuning Hyper-Parameters of Classifiers

In the following, we will discuss the configuration of the classifiers we used in the classification of microposts. Here, we have to distinguish between in-domain classification and cross-domain classification. For *in-domain classification*, we tuned a classifier on some development data (10% of the respective dataset which was excluded in the cross-validation experiments). For *cross-domain classification*, we did not tune the classifiers on the source domain since it would have meant tuning the parameters to the *wrong* domain. Either we ran a classifier in its default configuration or, if this was not possible, we determined a parameter setting which scored reasonably well on most domains. We also chose a setting that worked best for all classifiers/feature sets and did **not** favour one particular classifier/feature set. For each classifier, a single configuration was used for all cross-domain experiments.

6.1.1 Support Vector Machines (SVM)

For SVM on in-domain classification, we only considered the *cost-factor* by which training errors on positive examples outweigh errors on negative examples. In SVM^{light}, this corresponds to the j-parameter. This parameter turned out to

[6] The better these methods perform, the less there is a need for a lexicon of abusive words, as we propose this in this paper. A strong performance of a method that derives word-level information from microposts means that all that one actually needs to accomplish this task would be labeled microposts. In other words, there would not be any need for some annotation on the word level and dedicated features.

be crucial for our experiments since it prevents classifiers from producing a majority classifier. All our datasets have an imbalanced class distribution where the abusive comments always represent the minority class.

For cross-domain classification, we could not run the classifier in a default configuration because of the high risk of producing majority classifiers. We chose for all experiments the setting $j = 2$. Even with that parameter setting, we occasionally produced majority classifiers. However, other parameter settings would have strongly preferred some individual classifier/feature set and therefore would have resulted in a biased evaluation.

6.1.2 FastText

FastText [Joulin et al., 2017] has been promoted as a simple and efficient baseline system. Due to its model's simplicity, the tool is not dependent on extensive parameter-tuning. We therefore decided to run this classifier for all experiments with its ***default configuration***. The only exception is that we used ***pre-trained embeddings***[7], which meant a consistent improvement for all experiments in which we employed that classifier.

6.1.3 Recurrent Neural Networks (RNN)

Due to the high amount of parameters in deep learning, we contacted the authors of Pavlopoulos et al. [2017] (the most recent publication that examines abusive language detection with the help of deep learning) to inquire their precise parameter settings for RNN from their experiments. They stated that ***drop-out*** need not be used. Moreover, the ***number of epochs*** should be kept to 1. As a ***learning rate*** 0.01 (the default setting of *keras*[8]) was recommended. After some initial testing, we confirmed that this setting was also generally applicable to our dataset. So, for our in-domain experiments, we decided only to tune ***batch size*** and the ***number of hidden units***. For cross-domain classification, we set both batch size and number of hidden units to 50. This configuration, or a very similar one, was optimal on most domains.

6.2 Reconstructing the Feature Set from Nobata et al. [2016]

For supervised classification of abusive language, we consider the feature set by Nobata et al. [2016] as the most sophisticated that has been reported for this task. Therefore, we use this feature set a strong baseline. Since several components were not publicly available, we had to make certain modifications in order replicate the respective features.

6.2.1 Word Embeddings

For word embeddings, we re-used those embeddings we also used for our word-level experiments, that is, a concatenation of two embeddings which have been induced on the two large corpora we employ in our work (i.e. *Amazon Review Corpus – AMZ* [Jindal and Liu, 2008] and the *Web As a Corpus – WAC*

[7] We use the same embeddings we employed in other experiments and classifiers, i.e. the combination of word2vec-embeddings induced from the *Amazon Review Corpus – AMZ* [Jindal and Liu, 2008] and the *Web As a Corpus – WAC* [Baroni et al., 2009].

[8] `https://keras.io/`

[Baroni et al., 2009]). Unlike Nobata et al. [2016], we do not possess any larger unlabeled corpora from the domains from which our datasets originate on which we could have induced more domain-specific embeddings. Our word embeddings were induced with word2vec [Mikolov et al., 2013]. We used that tool in its ***default configuration***, that is 200 dimensions, continuous bag of words, context window of 5 tokens.

6.2.2 Comment Embeddings

Due to the unavailability of the specific method to induce embeddings for comments[9] (Nobata et al. [2016] refer to this embedding model as *comment2vec*), we used the implementation proposed by Lau and Baldwin [2016] (in its default configuration).

6.2.3 Linguistic Features

Among their linguistic features, Nobata et al. [2016] propose to use the *length of a comment in tokens*. We experienced a notable drop by using this very feature, so we excluded it for our final evaluation.

6.2.4 Syntactic Parsing

Nobata et al. [2016] employed the ClearNLP v2.0 dependency parser[10] for their experiments. We could not use this particular parser since one of our datasets (i.e. the dataset from [Waseem and Hovy, 2016]) represents tweets from Twitter. Standard parsers are known to perform very poorly on these types of texts. Therefore, we decided to use the Tweebo parser [Kong et al., 2014]. Since many of our datasets comprise similarly fragmented short texts as tweets, we also parsed those with Tweebo. The shortcoming, however, of this parser is that it only provides *unlabeled* dependency relations. Still, we think that the information provided by this parser is useful, since Nobata et al. [2016] motivate the usage of dependency-parse information by its ability to capture long-range dependencies. For instance, with dependency parsing, one might capture the predictive ngram *Jews_pigs* in the abusive utterance (7). In order to capture such long-range dependency, it does not require the presence of a dependency label (such as *nsubj*, *dobj* or *xcomp*). In other words, even Tweebo should be able to provide such information.

(7) Jews are lower class pigs.

6.2.5 The classifier

Finally, as a classifier, we incorporated this feature set into an SVM. As a tool we used again SVM^{light}. Nobata et al. [2016] employed Vowpal Wabbit's regression model[11]. We also tested this tool with the given feature set extensively. However, on several of our datasets, we obtained much inferior results than we obtained with SVM. Since the feature set of Nobata et al. [2016] is thought as a strong baseline, we decided in favor of carrying out our evaluation with SVM.

[9] By *comments* we understand an entire micropost we want to classify.
[10] `http://clearnlp.wikispaces.com/depParser`
[11] `https://github.com/JohnLangford/vowpal_wabbit`

Thus we get the best possible results with this feature set on the datasets of our evaluation.

6.3 Lexicon Expansion Based on Generic Word Embeddings

In order to demonstrate the effectiveness of our expanded lexicon using our proposed feature set, we also examined a more generic lexicon expansion. Like our proposed lexicon, it is trained on our newly created base lexicon. However, it is trained exclusively on word embeddings. Regarding the word embeddings, we used the induced embeddings we also employed in other experiments of this paper. For the expansion, we run experiments on algorithms implemented in the *SocialSent* package [Hamilton et al., 2016]. We compared the two competing algorithms *Densifier* [Rothe et al., 2016] and *SentProp* [Hamilton et al., 2016]. Since overall the latter outperformed the former on our data, we only listed the performance of *SentProp* in the main paper.

7 Etymological Information about the Three Recent Abusive Words Mentioned in the Introduction of the Main Paper

In the introduction of the main paper, we argued that one reason why the process of building a lexicon of abusive words is not a one-time annotation effort is that new abusive words constantly enter natural language. We illustrated this with three examples of recent abusive words. Below you find some etymological information about these words.

- **remoaner:** A Remainer; one who complains about or rejects the outcome of the 2016 EU referendum on the UK's membership of the European Union. A blend of *moan* and *Remainer*.[12] (*from Wiktionary*)
- **gimboid:** An incompetent person coined in the British television series *Red Dwarf*, possibly from *gimp* + *-oid*. (*from Wiktionary*)
- **twunt:** A combination swearword, in this case *twat* meets *cunt*. Thought to have been invented by humourist Chris Morris for the Channel 4 series 'Jam' (2000). Now a good description of any cunt who's a twat, or indeed vice-versa. (*from Urban Dictionary*)

References

Ramón F. Astudillo, Silivio Amir, Wang Lin, Mario Silva, and Isabel Trancoso. Learning Word Representations from Scarce and Noisy Data with Embedding Subspaces. In *Proceedings of the Annual Meeting of the Association for*

[12] This is one of the most recently coined abusive words. The crowdsourced dictionary *Wiktionary* already includes an entry. Unfortunately, the data dump of Wiktionary we used in our experiments was created a few months before the Brexit result (June 2016) when the term was not yet invented. As a consequence, this word is not part of the abusive lexicon we induced.

Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL/IJCNLP), pages 1074–1084, Beijing, China, 2015.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetti. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, Montréal, Canada, 2017.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 595–605, Austin, TX, USA, 2016.

Nitin Jindal and Bing Liu. Opinion Spam and Analysis. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 219–230, Palo Alto, CA, USA, 2008.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of Tricks for Efficient Text Classification. In *Proceedings of the Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pages 427–431, Valencia, Spain, 2017.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. A Dependency Parser for Tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012, Doha, Qatar, 2014.

Jey Han Lau and Timothy Baldwin. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany, 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at the International Conference on Learning Representations (ICLR)*, Scottsdale, AZ, USA, 2013.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive Language Detection in Online User Content. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 145–153, Republic and Canton of Geneva, Switzerland, 2016.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deeper Attention to Abusive User Content Moderation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark, 2017.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. Ultradense Word Embeddings by Orthogonal Transformation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL)*, pages 767–777, San Diego, CA, USA, 2016.

Partha Pratim Talukdar, Joseph Reisinger, Marius Pasca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 582–590, Honolulu, HI, USA, 2008.

William Warner and Julia Hirschberg. Detecting Hate Speech on the World Wide Web. In *Proceedings of the Workshop on Language in Social Media (LSM)*, pages 19–26, Montréal, Canada, 2012.

Zeerak Waseem and Dirk Hovy. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL – Student Research Workshop)*, pages 88–93, San Diego, CA, USA, 2016.

Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 1646–1652, Marrakech, Morocco, 2008.