Offence in Dialogues: A Corpus-Based Study

Johannes Schäfer[1] and **Ben Burtenshaw**[2]

[1] Institute for Information Science and Natural Language Processing
University of Hildesheim, Universitätsplatz 1, Hildesheim, Germany
johannes.schaefer@uni-hildesheim.de

[2] Computational Linguistics & Psycholinguistics Research Center
The University of Antwerp, Lange Winkelstraat 40-42, Antwerp, Belgium
benjamin.burtenshaw@uantwerpen.be

Abstract

In recent years an increasing number of analyses of offensive language has been published, however, dealing mainly with the automatic detection and classification of isolated instances. In this paper we aim to understand the impact of offensive messages in online conversations diachronically, and in particular the change in offensiveness of dialogue turns. In turn, we aim to measure the progression of offence level as well as its direction – For example, whether a conversation is escalating or declining in offence. We present our method of extracting linear dialogues from tree-structured conversations in social media data and make our code publicly available.[1] Furthermore, we discuss methods to analyse this dataset through changes in discourse offensiveness. Our paper includes two main contributions; first, using a neural network to measure the level of offensiveness in conversations; and second, the analysis of conversations around offensive comments using decoupling functions.

1 Introduction

Offensive language is a complex problem, especially in social media where operators are required to counter illegal hate speech in user-generated content. However, it is not clear what counts as offensive language since even humans struggle to find objective definitions (Chen et al., 2012). The general approach to this problem is to train systems for the detection of such unwanted content based on human annotations of empirically gathered data instances.

Several shared tasks engaged with the topic of offensive language detection in recent years, for example *OffensEval-2019* for “Identifying and Categorizing Offensive Language in Social Media” (Zampieri et al., 2019b) or *GermEval-2018*, the “Shared Task on the Identification of Offensive Language” (Wiegand et al., 2018). The difficulty for machine learning systems at this task becomes apparent when considering the performance scores of the submitted systems. For example, at *GermEval-2018* the best performing submitted system reached a macro-averaged F_1-score of only $76.77\ \%$. Here, systems struggle to simultaneously detect all various types of offensive language and it remains highly questionable if we can act on these automatic predictions and delete detected offensive language in practical applications.

Following the hint that deletion based on predictions of a machine learning system might not be the most appropriate course of action, we try to approach the problem of offensive language from another direction. In almost the same manner as mentioned above, we rely on machine learning of annotated instances to detect messages which might be offensive; however, we intend to act differently. Rather than deleting supposedly unwanted instances, we suggest to use tactics to counter offence. We aim for an empirical approach to automatically gather such tactics in a first step by a data analysis of instances where humans attempt to defuse offensive situations. In this paper we present our corpus creation using social media data from *Reddit* and discuss methods to analyse offensive dialogues. With our method we intend to classify conversations by offence direction, to facilitate future study on language use in offensive conversations. This research step could

[1] https://github.com/Johannes-Schaefer/oid_ranlp19

Proceedings of Recent Advances in Natural Language Processing, pages 1085–1093,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_125

prove vital to Natural Language Generation researchers in their effort to tackle offensive language by developing similar tactics.

Hate speech detection, which is closely connected to the detection of offensive language, however, focusing on illegal offence, has also been researched extensively in recent years. An overview of seminal work in this field is given by Schmidt and Wiegand (2017).

Context aware models for hate speech detection have been analysed by Gao and Huang (2017). Their dataset also preserves the thread structure of microposts, however, their approach is to use the context to gather additional features for the detection process while they do not focus on analysing the change of offensiveness in the conversation.

In related work on offensive language detection in conversations, Khatri et al. (2018) present an extensive data collection strategy using different sources of social media data. While they also utilise *Reddit* as data source, they only analyse individual utterances regarding offence.

A corpus with a focus on analysing conversations is presented by Walker et al. (2012) as they mine data from a forum and specifically consider the structure of comments in threads. While this corpus has several annotations which might be useful for exploring issues pertaining to online debate, they do not discuss offensive language.

Our research shows similarities to Zhang et al. (2018) who try to identify conversations that are likely to turn into offenses and predict the point in which this is likely to happen. We would also like to refer to the notions of constructive language which is discussed by Kolhatkar and Taboada (2017) and to a certain extent related to cases of defusing offensive conversations.

For our analysis of contexts of offensive language in dialogues, we decided to acquire our own corpus material which we process using methods tailored for this task. In summary, the research questions which drive our data analysis are as follows: How do people in dialogues react to offensive language? – Especially in terms of: what tactics do they try to counter offence? From a methodological view, we are particularly interested in investigating how to measure the change of offensiveness in turns of a dialogue.

The further sections of this paper are structured as follows: Section 2 presents our data sources and outlines the corpus construction process. In Section 3 we list our methods for data processing and analysis which are then applied to our dataset in experiments as described in Section 4. Finally, we conclude by discussing the results and the efficiency of our methods in Section 5.

2 Corpus Data

The analysis of offensive comments in dialogues requires a dataset of user posted messages which are referencing other messages. Such data can mainly be found on online social media where multiple users discuss a certain topic while interacting with each other. In this section we discuss our corpus construction process and motivate our selection of dialogues.

2.1 Data Sources

The typical sources of online social media data, *Twitter* and *Facebook*, have been mined extensively for text data to be researched, also with regard to containing offensive language or hate speech. The microposts from *Twitter* often lead to flat conversation structures as users mostly initiate a discussion or directly reply to an initial post. Hence, we disregarded this data source for our research on the change of offensiveness in a dialogue. We also decided not to use data from *Facebook* as we could not locate a restricted domain. We were afraid that when only including messages from a few selected *Facebook*-sites, we would not be able to get enough data for a statistical analysis.

To acquire a corpus of deeply structured dialogues about constrained topics, we decided to sample comments from *Reddit*[2] which is a social news aggregation, web content rating, and discussion website. In our corpus we only include comments from the Europe-*Subreddit*[3] where users post news or discussions which are geographically or politically related to Europe. We reason this decision on the basis that we – as Europeans – feel eligible to assess the content of these posts and we assume this topic to include lively (possibly heated) discussions containing offensive language. While *Reddit* is an American organisation, authors of posts in the Europe-*Subreddit* are mostly Europeans, but this is not restricted. Non-European users are also allowed to participate in the discus-

[2] `https://www.reddit.com/`

[3] A *Subreddit* is a forum dedicated to a specific topic as part of the website *Reddit*. The Europe-*Subreddit* can be found at `https://reddit.com/r/europe`.

sions, however, the content has to be related to topics from Europe.

2.2 Source Data Structure

Comments on *Reddit* are structured in threads, each one being a directed graph form of a rooted tree – similar to forums. An initial post, called submission, is directly posted in a *Subreddit* where users share content by posting stories, links, images, and videos. It can also just consist of a headline. Users can then reply to the initial post or to previously posted replies in a recursive manner. Thus, a thread can comprise several comments structured in a tree with the submission as a main or top-level (level 0) post, which corresponds to the root of the tree. Direct replies to the top-level post, direct successors of the root, we consider as being on level 1, replies to those in turn as being on level 2, etc. Leaf nodes are posts which have no further direct replies. Thus, there is always exactly one directed path from the root to any comment in a thread. A comment can technically never be a direct reply to multiple comments, i. e. cannot have multiple direct predecessors.

2.3 Data Acquisition

The first step to construct our corpus is to download *Reddit* posts using the *Python 3 psaw*[4] module, which is a minimalist wrapper for searching public *Reddit* comments and submissions via the *pushshift.io Reddit API*[5]. Our script selects all comments in the time frame from 2009-12-31 23:00:00 (first posts in the Europe-*Subreddit*) until 2019-04-04 22:00:00 (date of our corpus initialisation). The downloaded submissions and comments are stored as individual dictionary objects, however, contain metadata (ID for itself and a link to the ID of the predecessor) which allows to reconstruct the abovementioned tree structure. We store them in a *pickle* object and save it into a file.

2.4 Corpus Format

To be able to efficiently process these threads automatically as well as manually by annotation, we convert the downloaded data into a specifically tailored *Extensible Markup Language (XML)* corpus format with the following general structure: A single root element `subredditcorpus` is defined as containing all the other elements. It consists of multiple `submission` elements which correspond to the threads in our Europe-*Subreddit* data. A `submission` element contains an optional `main_post` element (which is not present in a few cases when the submission consists only of a headline) and an arbitrary number of `comment` elements. A `main_post` element can either consist of a `link` element, in cases where no text comment is submitted and only a link to an external site or another *Reddit* post is given, or of a `comment` element itself. A `comment` consists of a text string and can recursively nest further comments – besides the `comment` element of the `main_post` element, which never has a successor; however, we do not ensure this in our *XML Document Type Definition (DTD)*.

Several types of metadata are maintained in our *XML* corpus, such as post IDs which allow us to find the original source on the website. Additionally, we store for each comment the `date` of the post, the `author` ID (*Reddit* user name), the `author_flair` (which is a customisable string appearing next to the user name and specific to each *Subreddit*; in the Europe-*Subreddit* it is possible to choose a country name as a flair and users usually select their country of origin) and the `score` of the post which was assigned by other users (via down- or upvoting the post).

3 Methods

In this section we describe our methods for corpus annotation and processing. First, we present our offensive language detection system in Section 3.1 which is based on a neural network and predicts offensiveness probabilities for each comment. Then we give our methods for automatically extracting uniformly structured linear dialogues containing offensive language from the corpus in Section 3.2. Finally, in Section 3.3, we show how we intend to further analyse these linear dialogues by applying decoupling functions to model the change of offensive probability.

3.1 Offensive Language Detection

To detect the level of offensiveness of comments we use a supervised machine learning method, which is a typical approach for this task. We train a model on manually labelled messages which have been classified whether they contain offensive language or not. Our system operates solely on the linguistic text of an individual comment

[4] https://github.com/dmarx/psaw

[5] https://github.com/pushshift/api

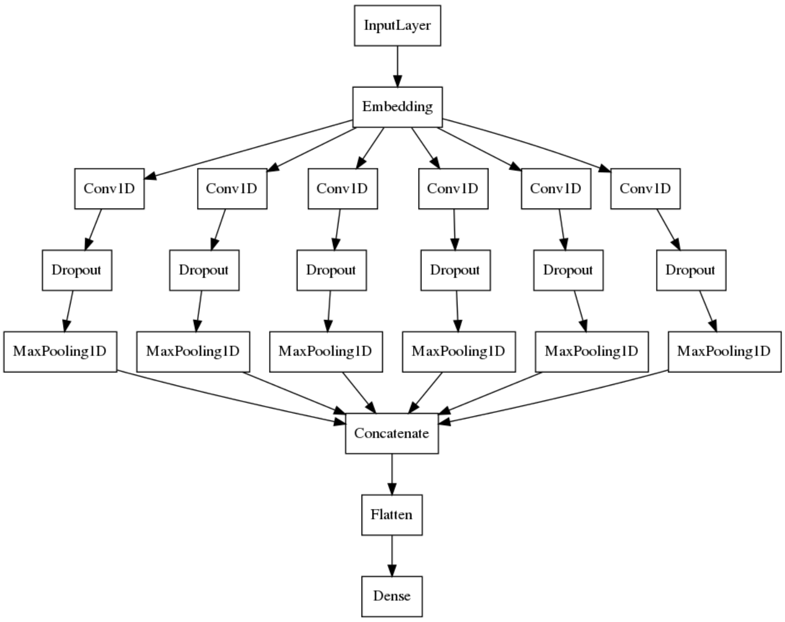

Figure 1: Neural network model architecture of our offensive language detection system.

as input since there is no overlap in the types of metadata between the training data and our corpus data. For each comment, given its text and the trained model, the system computes an offensiveness probability which is annotated in our corpus. In the remainder of this section we elaborate on a few details of our system – for the full configuration refer to our provided code.

We use a neural network to train a model of offensiveness of short text posts. A neural network is able to learn highly complex functions given enough labelled training data instances. Convolutional Neural Networks (CNNs) are structures commonly applied to natural language which can automatically identify sequences of words in a text that are significant features for a specific classification task. By design, CNNs contain regularisation which is capable to abstract from a limited set of training instances to unseen – ideally similar – test data instances.

Neural Network Architecture: Our overall neural network model is designed as follows (see also Figure 1). In a first step the tokenised[6] input text is encoded as a sequence in an embedding layer. We use (*Tweet-*) word embeddings by Deriu et al. (2017) to represent the meaning of our input text based on the principle of distributional semantics. To model offensive language based on the word embeddings of the input text, we apply six parallel CNNs with different window sizes (from one to six). These CNNs process the given sequence by moving a filter over it from left to right, in each step selecting a number (depending on its fixed window size) of consecutive words and computing a weighted combination of the dimensions of their word embeddings. Thereby, we aim to select word n-grams which are likely to be relevant to assess if a given text instance contains offensive language. Each CNN consists of a single convolutional layer followed by a dropout layer and a max pooling layer. With the inherent regularisation of a CNN by convoluting and max-pooling plus the additional dropout layer (here we use a dropout probability of 0.25), this step includes a considerabe amount of regularisation. We justify this design for our encoder by considering that our test data (*Reddit* comments) is vastly different from the training data (*Twitter* microposts) in terms of text type and website. By using a high amount of regularisation we intend to be able to generalise well when predicting on out-of-domain text. In the final steps of our neural network, the encoded output of the parallel CNNs is concatenated, reformatted using a flatten layer, and finally we compute a score which can be interpreted as offensiveness probability using a densely-connected layer with a sigmoid activation function. The exact parametrisation of the model is given in our provided code.

[6] For tokenisation we use the *NLTK TweetTokenizer* (`www.nltk.org/api/nltk.tokenize.html`) which includes custom-built methods to deal with social media text.

Training Procedure: We train our model using the *Offensive Language Identification Dataset (OLID)* (Zampieri et al., 2019a) which has been used for the *OffensEval 2019*[7] shared task of *"Identifying and Categorizing Offensive Language in Social Media"*. The training dataset consists of 13,240 *Tweets* as individual messages labelled for containing offensive language or not. The distribution of non-offensive to offensive messages in this dataset is approximately 2:1, i. e. there are 4,400 messages labelled as being offensive. Our training algorithm optimises the weights of the model based on 11,916 samples (90%) of these annotated instances and validates on the remaining 1,324 samples (10%) to avoid overfitting. Early stopping is executed when the performance on the validation set did not improve in the last few training epochs and we load the weights from after the epoch which lead to the maximum validation set performance.

Prediction: During testing we apply the abovementioned trained model to each comment of our corpus and annotate the predicted offence as follows. We add new metadata to our *XML* corpus by including the attributes `p_off` and `off` for the `comment` elements. While the predicted probability score is directly stored as value of `p_off`, the value of `off` expresses if the predicted probability of a comment to contain offensive language is higher than 0.5, which we annotate as one of the possible binary values `"True"` or `"False"`.

3.2 Extraction of Linear Dialogues

The abovementioned corpus creation process (cf. Section 2) provides us with a dataset of comments structured in a forum-like manner as a tree – a top post with direct replies which can in turn have direct replies themselves and so on. After we processed this corpus using our offensive language detection system, the comments in this corpus include annotations expressing their offensive probabilities. However, as we aim to analyse conversations as turn-based dialogues around offensive comments, we now have to filter the corpus and extract such linear dialogues. In this section we describe our method for this extraction process.

Our corpus can be formally described as a set of comments C, a set of submissions $S \subset C$ and a set of relations R, where each comment $c_i \in C$ is either a submission or top post $c_i \in S$ or a direct reply to exactly one other comment $c_j \in C$, i. e. $\forall_{c_i \in C} \exists!_{c_j \in C} : (c_j, c_i) \in R$.

#comments = n	#submissions
$n \leq 10$	265,068
$10 < n \leq 100$	83,295
$100 < n \leq 1000$	8,539
$n > 1000$	80
average	14
average$_{(n>1)}$	54

Table 1: Number of comments per submissions.

For our target of linear conversations we now need to extract a simplified dialogue out of this dataset, i. e. a linear structure of turns. In general, we consider a linear conversation an ordered list of comments from our corpus as $\{c_1, c_2, ..., c_n\}$ where $\forall_{i \in \{1,...,n-1\}} : (c_i, c_{i+1}) \in R$.

We assert the following requirements to refine this structure. As we intend to compare dialogues around offensive comments with each other and find patterns amongst them, we have to analyse a uniform structure of contexts. Thus, we require all linear conversations to have a fixed length (number of comments) and the context before and after an offensive comment to be equal in size. Additionally, we included the the top-level comment for each linear conversation to give information about the general topic in each case.

Thus, we define a linear conversation from our corpus as $l = \{c_0, c_1, c_2, ..., c_n\}$, where $\forall_{i \in \{0,...,n\}} c_i \in C$, $\forall_{i \in \{1,...,n-1\}} : (c_i, c_{i+1}) \in R$, $c_0 \in S$, $\texttt{p\_off}(c_{\frac{n}{2}}) > 0.5$ with $n = 2 * k + 2$, where $k \in \mathbb{N}$ corresponds to the number of comments of the context to the left and right of the offensive comment (or window size) and there has to be a path from c_0 to c_1.

3.3 Decoupling Functions

In order to determine the progression of offensive probability we have preliminarily tested two gradient based approaches on linear dialogues of two users: firstly, using the gradient of all dialogue turns $(c_1, \ldots, c_n)$; secondly, comparing the gradient before and after the offensive comment (comparison of $(c_1, \ldots, c_{\frac{n}{2}-1})$ to $(c_{\frac{n}{2}+1}, \ldots, c_n)$). We visualise these approaches in our experiments in Section 4.4. In future work we intend to compare the gradient of the two different users. This will theoretically show the most meaningful forms of

[7] https://competitions.codalab.org/competitions/20011

c_i	author	off_p	comment
0	user_1	0	Do you find your government to be trustworthy?
1	user_2	0.11	So why exactly they don't feel safe? Because Russia planned Babchenko assasination? Or because SBU managed to prevent that and safe him and 30+ others?
2	user_1	0.20	The way I understood it, they were not feeling safe in the first place, and now they feel that the authorities undermined what little trust they had.
3	user_2	0.97	If your understanding is correct than I must state that those journalists are dumb as fuck and infantile people.
4	user_1	0.85	Why are they dumb? Because they are sceptical of this whole circus?
5	user_2	0.30	Because they can't set priorities between national security, life and death of people and their personal feelings.
6	user_1	0.52	The issue is that they don't trust the government. They don't believe this thing was needed, or even worse, they don't believe that there was an actual threat in the first place. Do you find your government to be trustworthy? Why do you believe that this was real?
7	user_2	0.64	"The issue is that they don't trust the government." No one trust gov-t in Ukraine. Any gov-t. It's a national feature. Sometimes it's good, sometimes it's bad. And up to 100% of journalists only criticizing the gov-t all the time - it's just how things work here. People don't trust the officials and journalists wat to be in trend. I doubt that any of those journalists actually lost some trust to gov-t. They just never had it and now using this situation to state it.

Table 2: Example of a linear dialogue containing offence from our corpus.

dialogue progression. We plan to publish a detailed analysis of these examples, but for illustrative purposes and to articulate the motivation in data collection, below are brief overviews of instances where offence probability declines.

4 Experiments

In this section we describe our observations when we applied the abovementioned methods to our corpus data.

4.1 Corpus Analysis

The download of *Reddit* submissions and comments using the mentioned API led to the total number of 11,217,768 posts (356,982 being submissions). In Table 1 we show the distribution of comments per submissions for certain ranges, to gain an understanding of how comments are structured in threads in this dataset. While the vast majority of submissions (approximately 74%) have very few replies ($\leq$ 10 comments), only 80 submission have more than 1000 comments. However, there is a substantial amount of threads with more than 10 comments and the average number of comments per thread – if we exclude threads with none or only one comment – is 54. Thus, we consider this dataset to be rich enough to study conversations and to be representative for different phenomena.

4.2 Offensive Language Detection

Our training algorithm for offensive language detection calls early stopping on training epoch 10, where a maximum binary accuracy is reached on the validation dataset. The evaluation after this epoch shows a binary accuracy of 0.82 on the training data and a binary accuracy of 0.73 on the validation data.

4.3 Extraction of Linear Conversations

To extract linear conversations as we defined them above, we tested different window sizes for the context. We decided to at least have 50k instances of conversations around offensive comments as we expect this to be a representative set of different types of phenomena. We set the window size $k = 3$ for the contexts before and after offensive comments, which leads to 67,456 instances of linear dialogues of length 7 (plus one for the top post), i. e. here $l = \{c_0, c_1, \ldots, c_7\}$ while $\texttt{p\_off}(c_4) > 0.5$. In Table 2 we provide an individual example of a linear conversation from our

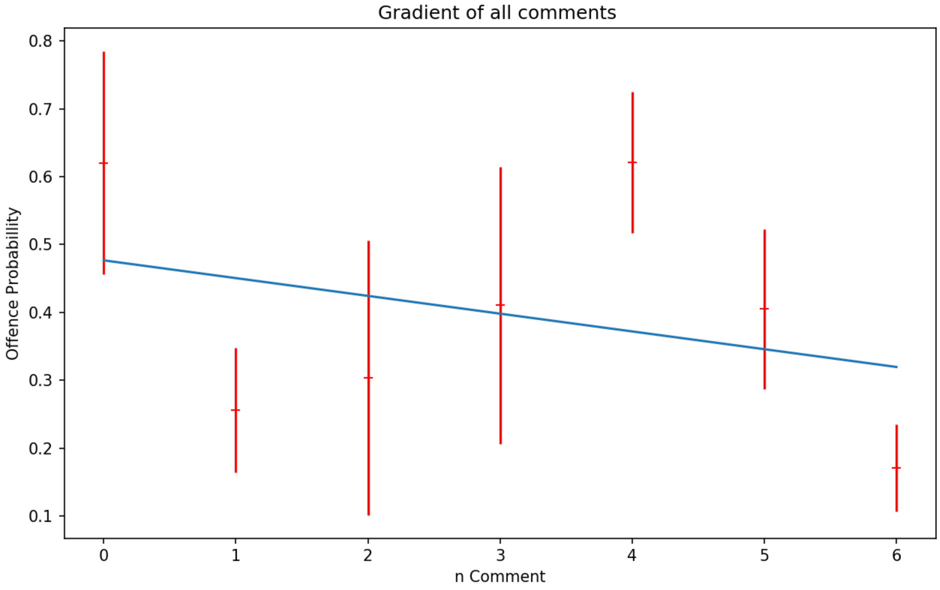

Figure 2: Gradient of all comments in extracted linear conversations.

dataset including offensive probabilities.

We have to consider that linear conversations from our dataset can have a certain partial overlap. From the data structure it only follows that the comments in the context before the offensive comment are given, here $\{c_0, c_1, c_2, c_3\}$. For the comments after the offensive comment (here $\{c_5, c_6, c_7\}$), there is the possibility to produce multiple linear conversations when there are multiple direct replies to one comment. We call this case branching, i. e. when we have multiple linear conversations which differ only in some of the comments $\{c_5, c_6, c_7\}$ – when following a different branch in the tree.

To analyse the frequency of this phenomenon, we counted how many unique first parts exist in our set of 67,456 linear conversations. If we only consider $\{c_0, \ldots, c_4\}$, there are 54,286 unique instances. If we add c_5, there are 57,077 unique instances and if we add c_6 there are 60,647 unique instances. Considering these values, we assume that branching is rather infrequent and we have mostly entirely different linear dialogues of this fixed length.

4.4 Progression of Offensive Probability

We now want to investigate the change of the level of offensiveness in turns of our linear dialogues statistically.

Complete Linear Dialogue Progression: We calculated the linear regression gradient of all comments in the collected linear dialogue, and searched for instances where gradient decline was the steepest. In simple terms, these reflect conversation where replies have been less offensive than the stimulus. The graph in Figure 2 shows the mean placement of each dialogue turn for the 50 steepest gradient dialogues, as well as standard deviation in red; a line of best fit is shown in blue. A clear downward vector in the overall gradient is shown, as well as a steep decline in the latter half of the dialogue, standard deviation accounted for.

Pre/Post Offence Comparison: We used the initiating offensive instance (comment 4) to split the dialogue into two halves (comment 1-4 and 4-7), and then we compared their gradients. We searched for instances with opposing directions. Unlike the above approach, comparing two halves allows us to see the impact on dialogue of the offensive comment, and measure how it was reacted to. The graph in Figure 3 shows a mean downward trajectory for the fifty most clear examples of this gradient.

In the example instance given in Table 2 the initiating comment probability is 0.85 in a reply to a clearly offensive comment towards a group of journalists (probability of 0.97). The latter half of the dialogue passively expands on that point, without returning to profanity or offence. We can reasonably say that user_1 is consistently responding to user_2 without offending, and therefore of interest to a study on inoffensive approaches to offensive dialogue.

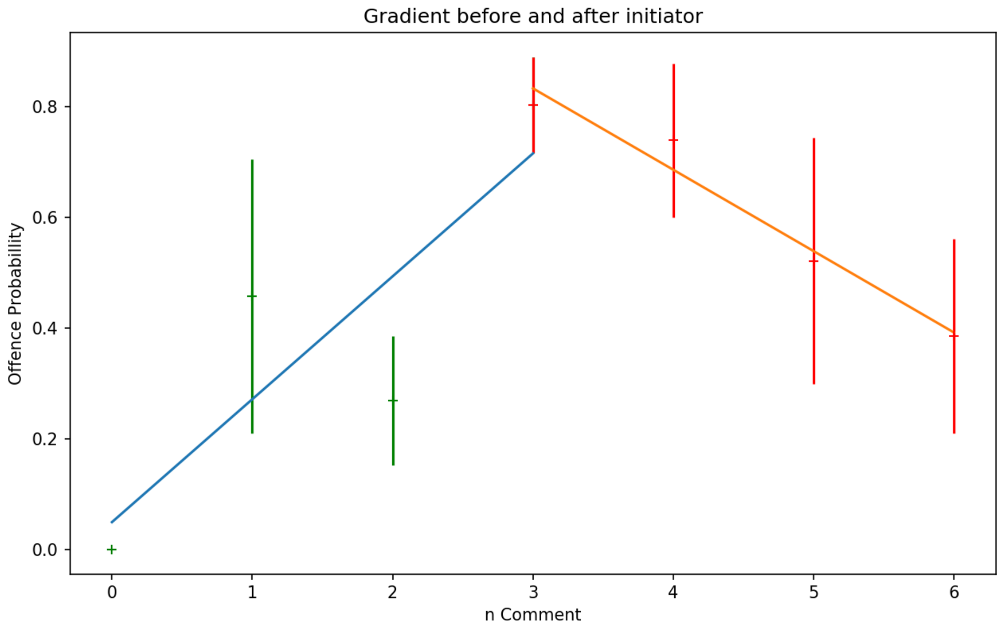

Figure 3: Gradient in comments before and after offensive comment trigger.

5 Conclusion

In this paper we presented our data collection strategy for a corpus containing conversations and discussed methods to analyse the corpus for linear dialogues around offensive comments. With a substantial dataset of over 11 million posts we are able to analyse over 50k unique linear dialogues, each consisting of seven turns predicted to contain offensive language. We now discuss our two main contributions.

Using offensive-tagged individual messages as training data for assessing the offensiveness of dialogue turns: Our method to detect offence is based on a machine learning model which predicts whether individual messages contain offensive language or not. In our model architecture, the computed probability expresses the confidence of the system that words and n-grams of the post are typical expressions of offensive language. We understand the offensiveness of a dialogue turn to be approximately in line with this assessment. A post can be seen as highly offensive when it contains several word sequences which are usually used to express offence. A low level of offensiveness can be expressed by using word sequences which are only incidentally used in offensive comments, i. e. they might for example not be offensive to everyone.

We also want to note that our offensive language detection model is not fully optimised: we intend in future work to experiment with different training data. The rather low scores given for the performance on the training/validation set can also be justified by the high amount of regularisation in our model which was implemented to make it generalise more when applied to our dataset (which is different from the training data). The training algorithm also does not optimise on accuracy as we use class weights, giving the more infrequent class of offensive comments a higher weight. If we further go into the direction to base methods for automatic analyses on the computed scores, it might be worth to investigate further how to optimise the detection system.

Analysing conversations around offensive comments using decoupling functions to find tactics to counter offence: We have shown that the analysis of gradients should be considered when we want to measure the change in offensiveness of a conversation. With the use of decoupling functions it especially seems suitable to split dialogues around offensive comments into two halves to find tactics to counter offence, i. e. especially instances where the gradient declines after the offensive comment trigger.

In future work we aim to focus on researching manual and statistical methods to find tactics to counter offence. However, the analyses of the given dataset and the provided code show promising results by pointing into the right direction.

References

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

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, pages 1045–1052.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. INCOMA Ltd., Varna, Bulgaria, pages 260–266.

Chandra Khatri, Behnam Hedayatnia, Rahul Goel, Anushree Venkatesh, Raefer Gabriel, and Arindam Mandal. 2018. Detecting offensive content in open-domain conversations using two stage semi-supervision. *arXiv preprint arXiv:1811.12900* .

Varada Kolhatkar and Maite Taboada. 2017. Constructive language in news comments. In *Proceedings of the First Workshop on Abusive Language Online*. Association for Computational Linguistics, Vancouver, BC, Canada, pages 11–17.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. pages 1–10.

Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *LREC*. Istanbul, pages 812–817.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria September 21, 2018. - Vienna, Austria: Austrian Academy of Sciences. Pp. 1-10* .

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). *arXiv preprint arXiv:1903.08983* .

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 1350–1361.