Attending the Emotions to Detect Online Abusive Language

Niloofar Safi Samghabadi, Afsheen Hatami, Mahsa Shafaei
Sudipta Kar and Thamar Solorio
Department of Computer Science, University of Houston
{nsafisamghabadi, amhatami, mshafaei, skar3, tsolorio}@uh.edu

Abstract

In recent years, abusive behavior has become a serious issue in online social networks. In this paper, we present a new corpus for the task of abusive language detection that is collected from a semi-anonymous online platform, and unlike the majority of other available resources, is not created based on a specific list of bad words. We also develop computational models to incorporate emotions into textual cues to improve aggression identification. We evaluate our proposed methods on a set of corpora related to the task and show promising results with respect to abusive language detection.

1 Introduction

Nowadays, abusive behavior has become a rising problem in online communities (Jones et al., 2013; Ybarra and Mitchell, 2004). Such adverse behavior can have serious effects on the physical, mental, and social health of online users, among whom teenagers and young adults are the most vulnerable group.[1] To combat this problem at scale, automated Natural Language Processing (NLP) systems can help identify potentially abusive language.

In recent years, there have been several efforts to automate the detection of offensive language across social media platforms. Lexical features have been proven to work quite well for this task (Dinakar et al., 2012; Davidson et al., 2017). However, such features introduce some bias into the systems by heavily relying on profane words, whereas reports show that most profanities are used in a neutral way in today's teen talks (Samghabadi et al., 2017; Vidgen et al., 2019). The following examples signify the need for linguistically more sophisticated techniques beyond profanity dependent models to detect abusive language:

Neutral: *Damn you are such a BEAUTIFUL F*CKING MOMMY!*
Offensive: *u should use ur hands to choke urself.*

In fact, most of the resources available for abusive language detection have been created based on either a list of bad words or seed words related to abusive topics. In this paper, we aim at tackling this limitation by proposing a new method for sampling the data without focusing on a specific bad word list. We are interested to collect this new dataset from a social media website that is specifically popular among youth, since they are the most vulnerable group of users when it comes to online abuse.We scrape our data from Curious Cat,[2] a semi-anonymous question-answering website, that has increased in popularity among teenagers. This platform provides a way to interact anonymously, which opens the door for digital abuse. On this website, users can choose not to reveal any personal information on their account, as well as post comments/questions on other users' timelines anonymously. Additionally, on average, the posts are too short in length. These properties limit both the content of a post, as well as the information about the sender of that post.

To overcome the aforementioned challenges within the data, we propose a new methodology to integrate emotional information into textual cues from the input text to decide whether it is offensive or not. Our main contributions in this paper are as follows:

- We introduce a new corpus for the task of abusive language detection, which is not created based on a specific list of profane words.
- We develop approaches for incorporating emotions into textual information to improve abusive language detection, and create unified

[1] http://enough.org/stats_cyberbullying

[2] https://curiouscat.me

Proceedings of the Fourth Workshop on Online Abuse and Harms, pages 79–88

https://doi.org/10.18653/v1/P17

deep neural models that show promising results across several relevant corpora from various domains.

- We introduce Gated Emotion-Aware Attention (GEA) that dynamically learns the contribution of emotion and textual information to weigh the words inside a sequence. We show that this new attention mechanism significantly outperforms the regular attention, which only utilizes textual hidden representations to learn the word weights when the input text is short and noisy.

2 Related Work

Abusive language identification and hate speech detection have been addressed by many research papers (Mishra et al., 2019c; Schmidt and Wiegand, 2017). Most of the related works have employed feature engineering approaches, and use a combination of different types of lexical, syntactic, semantic, sentiment and lexicon-based features along with classic machine learning algorithms such as Support Vector Machines (SVM), and Logistic Regression (Samghabadi et al., 2018; Davidson et al., 2017; Nobata et al., 2016; Gitari et al., 2015; Van Hee et al., 2015).

Due to the popularity of deep neural networks, multiple studies have recently been conducted in order to explore the performance of these models on the task of aggression identification. Most of these studies are focused on hate speech detection within Twitter. Gambäck and Sikdar (2017) use a Convolutional Neural Network (CNN) based model, and investigate different textual and embedding features as the input to the model where word2vec produces the best results. Badjatiya et al. (2017) conduct an extensive evaluation on multiple traditional and deep learning approaches, and report the best results using an ensemble of LSTM and Gradient Boosted Decision Trees. There are also a few works that try to incorporate user information into the model, using approaches such as Graph Neural Networks (Mishra et al., 2019a,b; Ribeiro et al., 2018) to learn the structure of online communities along with the linguistic behaviors of the users within them. The main limitation of these approaches is that they are not applicable to the social media platforms that offer anonymity options to the users such as Curious Cat and ask.fm.

Several research papers have proven that emotion lexicons are helpful features for the tasks of abusive language and hate-speech detection (Koufakou and Scott, 2020; Wiegand et al., 2018; Martins et al., 2018; Corazza et al., 2018; Alorainy et al., 2018; Gao and Huang, 2017). There is also one study that shows jointly modeling of emotion classification and abuse detection, through a multi-task approach, can improve the performance of the latter task (Rajamanickam et al., 2020).

Our methodology has two key differences in contrast to other existing methods: (1) Instead of using an ensemble approach, we create unified deep neural architectures that show very promising results across multiple domains, and (2) We do not use any user-level information in our model. Therefore, the model can be applied to various online platforms, even those that offer anonymity.

3 Dataset

We collected the data from Curious Cat, which is a semi-anonymous, question-answer social media platform. Curious Cat is very popular among the youth and has more than 15 million registered users. On this website, users can choose not to reveal any personal information on their account, as well as post comments/questions on other users' timelines anonymously. The anonymity option available on Curious Cat opens the door for digital abuse. Due to these properties, there are two significant limitations with respect to Curious Cat data: (1) The post content is usually too short making abuse detection harder, and (2) There is very limited information, if any, about the sender of a post.

3.1 Data Collection and Annotation

We crawled around 500K English question-answer pairs from 2K randomly chosen users of Curious Cat. To avoid having bias through some specific swear words in the data, we did not use a particular list of bad words to find potentially offensive messages. Instead, we exploited the state-of-the-art classification method for abusive language detection on ask.fm (Samghabadi et al., 2017)[3] because of two reasons: (1) The format of the data in Curious Cat and ask.fm is very similar,[4] and (2) This method utilizes lexical features that make it capable of learning new words and phrases related to the offensive class. This model combines lexical, domain-specific, and emotion-related features and

[3] We use the code available in https://github.com/NiloofarSafi/Detecting-Nastiness

[4] https://ask.fm

uses an SVM classifier to detect nastiness. We train that classifier on the full ask.fm dataset and apply it to Curious Cat to automatically label all rows of data. While ask.fm and Curious Cat have the same format, we noticed key differences between them, which may substantially affect the quality of automatic labeling. For instance, with Curious Cat, we observe numerous sexual posts that are full of profanities, yet not offensive to the user, e.g, a user may encourage others to post sexual comments to him/her, like the following example:

Question: *I wanna s*ck your d*ck so hard and taste your c*m.*

Answer: *Enter my DMs beautiful.*

Therefore, we randomly selected 2,482 question-answer pairs, where 60% were chosen from the negative/offensive labeled data, and 40% selected from the positive/neutral labeled data (we only considered the label of the questions). Four in-lab annotators[5] annotated the data. Each row was tagged by three different annotators, and the final label assigned to each instance by majority voting. Based on the annotations, the Fleiss's kappa (Fleiss, 1971) score is 0.5 that shows a moderate agreement among the annotators. Figure 1 shows the rate of "complete agreement" among all annotators for positive and negative questions and answers. By complete agreement, we mean the case where all the annotators assigned the same class to an instance (in Curious Cat data, an instance could be a question or an answer). Based on the figure, the complete agreement on the negative/offensive class is much less than the positive/neutral one. This observation demonstrates the fact that the perceived level of aggression is very subjective, so our final agreement score is reasonable Sap et al. (2019). It is also interesting that for negative instances, the annotation results show more complete agreements on top of the questions compared to answers. This indicates that it was more difficult for the annotators to decide whether a reply to a comment is offensive.

3.2 Data Statistics

Table 1 shows the final distribution of the proposed Curious Cat corpus. Statistics show that 95% of negative comments were posted on users' timelines anonymously. Looking at the labeled data, we also found that about 100 instances of abusive posts do not include any profanities, and 1327 positive/neutral posts have at least one profane word. It shows that the proposed sampling method could capture the implicit forms of abusive language as well as explicit ones. This technique also samples the posts that include bad words, but are not attacking other users.

[5]Including one graduate and three undergraduate students

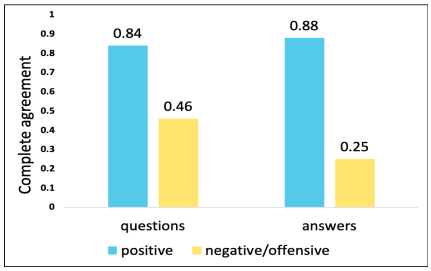

Figure 1: Complete agreement for questions and answers across negative/offensive and positive labeled data.

Class	Question	Answer	Total
Offensive	609	171	780
Neutral	1873	2311	4184
Total	2482	2482	4964

Table 1: Curious Cat data distribution.

3.3 Other Abusive Language Datasets

We also experimented with the following available corpora to better qualify the performance of the proposed models: (1) ask.fm dataset (Samghabadi et al., 2017), (2) Kaggle insult dataset,[6] and (3) Wikipedia personal attacks dataset (Wulczyn et al., 2017). Table 2 compares all resources that we use in this paper. Our Curious Cat data can be accessed through our website.[7]

Data	Size	%Negativity	Avg length
Curious Cat	4964	15.71%	15.30
ask.fm	11194	18.08%	13.92
Kaggle	6597	26.42%	38.35
Wikipedia	~115K	11.70%	81.29

Table 2: Data comparison. The last column shows the average length of the posts with respect to the number of words.

4 Methodology

Emojis help online users to better express their feelings within the text. With this notion, we hypothesize that emojis are effective tools to provide additional context for online comments, resulting in bet-

[6]https://www.kaggle.com/c/detecting-insults-in-social-commentary

[7]https://ritual.uh.edu/curious-cat-corpus/

ter offensive language recognition. For capturing emotions from the text, we use DeepMoji (Felbo et al., 2017) pre-trained on Twitter data. As for the output, this model creates a representation for 64 frequently used online emojis that shows how relevant each emoji is to a given text. Figure 2 illustrates the top 5 emojis that DeepMoji assigned to one neutral and one offensive instances in our Curious Cat data. Both of these comments are very short and include the bad word "die". We can see that DeepMoji correctly recognized the tone of the language in both examples. The colors also show the attention weights assigned by DeepMoji model. The darker colors indicate higher attention weights. Interestingly, the word "die" is attended the most in the offensive instance.

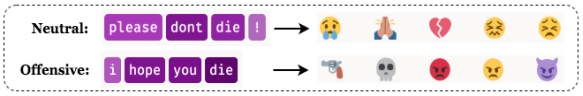

Figure 2: Top 5 emojis that the DeepMoji model assigned to one neutral and one offensive instances from our Curious Cat data. The words are colored based on the attention weights given by the DeepMoji model. Darker colors show higher attention weights.

In this paper, we examine two different approaches to create the model that combines DeepMoji and textual representations to detect whether a given input text is offensive or not. The motivation behind this idea is to exploit emotional representation to better distinguish the use of profanities in an offensive way from a neutral way. Both models include the two following main modules:

1. **Bidirectional Long Short-Term Memory (BiLSTM):** This module has an embedding layer that generates the corresponding embedding matrix for the given input text. Then, we pass the embedding vectors to a Bidirectional LSTM (BiLSTM) layer to extract the contextual information from the sequences of words.

2. **DeepMoji:** This module feeds the input to the DeepMoji model and pass the last hidden representation through a non-linear layer to project it into the same space as the output from the BiLSTM module.

For combining the output of the above mentioned modules, we try two following approaches:

Concatenation: One popular way to incorporate information into deep neural models is concatenation. In this approach, we pass the output of BiLSTM to an attention layer, same as Bahdanau et al. (2015), to aggregate the output hidden states of BiLSTM into a single vector. Within this layer, we calculate the weighted sum of $r = \sum_i \alpha_i h_i$, where $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$ is the concatenation of the forward and backward hidden states of BiLSTM. α_i stands for the relative importance of words which is measured as follows:

$$\alpha_i = softmax(v^T tanh(W_h h_i + b_h)) \quad (1)$$

where W_h is the weight matrix, and b_h and v are the parameters of the model. We refer to this attention model as the Regular Attention (RA) in the rest of paper. We concatenate the outputs of the RA and DeepMoji module. The resulting vector is then fed into a hidden dense layer with 100 neurons. To improve generalization of the model, we use batch normalization and dropout with a rate of 0.5 after the hidden layer. Finally, we use a two neuron output layer along with softmax activation to predict whether the input text is offensive or not.

Gated Emotion-Aware Attention (GEA): In this approach, instead of directly concatenating the text and DeepMoji representations, we hypothesize that it is not enough to only focus on the word representations in the attention model because of two reasons: (1) Many bad words may also be used in a neutral way to make jokes and provide compliments among friends, and (2) Some texts do not contain any profanities, but are still offensive to the receiver. Both reasons may confuse the model for final prediction. Therefore, we design the GEA mechanism to consider not only the word representations, but also the emotions behind the text to better determine the most relevant words in a post. We use the idea of Gated Multimodal Unit (Ovalle et al., 2017) to create GEA. The overall architecture of this model is shown in Figure 3.

Let us assume that h_i and e_i are the output representations of BiLSTM and DeepMoji modules, respectively. For each of them, we have a gate neuron (represented by σ nodes in Figure 3) that controls the contribution of each of these features to calculate the attention weights. We calculate the α_i as follows:

$$h'_i = tanh(W_h.h_i) \quad (2)$$

$$e'_i = tanh(W_e.e_i) \quad (3)$$

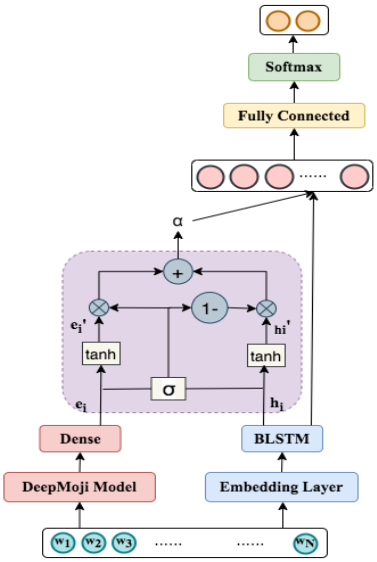

Figure 3: Overall architecture of the Gated Emotion-Aware Attention (GEA) model.

$$z_i = \sigma(W_z.[h'_i, e'_i]) \quad (4)$$

$$hid_i = z_i * h'_i + (1 - z_i) * e'_i \quad (5)$$

$$\alpha_i = softmax(v^T hid_i) \quad (6)$$

where $\{W_h, W_e, W_z\}$ are weight matrices, and v is the parameters of the model. W_e is shared across the words and adds emotion effects to the attention weights. The output of the attention layer is the weighted sum r calculated as follows:

$$r = \sum_i \alpha_i h_i \quad (7)$$

Finally, we pass the output of the attention mechanism to a fully connected layer with the same settings as the Concatenation model, and generate a two-dimensional output.

5 Experiments and Results

We stratified split Curious Cat data into train and test sets with a 70:30 training to test ratio, and use 20% of the train data as the validation set. For the other corpora, we use the same train, validation, and test folds as used by the original papers. As for preprocessing, we truncate the posts to 200 tokens, and right-pad the shorter sequences with zeros. We use Binary Cross Entropy to compute the loss between predicted and actual labels. To smooth the imbalance problem in the datasets, we add information about class weights to the loss function. The network weights were updated using Adam optimizer (Kingma and Ba, 2015) with a learning rate of $1e^{-5}$. We trained the model over 200 epochs, and reported the test results based on the best macro F1 obtained from the validation set.

5.1 Baselines and SOTA Approaches

We compared our proposed model against the state-of-the-art and several strong baselines listed below:

DeepMoji Baseline: We directly passed the output of the DeepMoji module to the dense and output layers. The motivation behind this baseline was to estimate the power of the DeepMoji model to detect abusive language on its own.

BiLSTM + RA: In this baseline, we do the classification, only using the textual information. This model uses the RA on top of BiLSTM module and directly passes the output representation to the fully connected and output layers. The motivation behind this model is to compare the performance of RA with GEA.

BERT Baseline: We directly passed the hidden representation of the BERT last layer for [CLS] token to the dense and output layer. With this model, we aim at testing the power of BERT as a feature extractor for the task of abusive language detection.

Sam'17 (Samghabadi et al., 2017): This is the state-of-the-art for the ask.fm corpus and applies an SVM classifier on top of a combination of various features.

Kaggle Winner: It shows the results of the winner of Kaggle competition on detecting insults in social commentary. This model includes an ensemble of several machine learning classifier with word n-grams and character n-grams lexical features.[8]

Bodapati'19 (Bodapati et al., 2019): This work reported the state-of-the-art results on the Wikipedia dataset. The authors added a single dense layer on top of BERT to fine-tune it for the task of abusive language detection. We implemented this model ourselves since the code was not released.

5.2 Classification Results

For the evaluation, we use the F1 score for the negative/offensive class, since this is the class of

[8]The code for this model is available through the competition discussion page: https://www.kaggle.com/c/detecting-insults-in-social-commentary/leaderboard

interest. We also report the weighted F1 score, which calculates the average performance over both classes. This is to ensure that the model does not sacrifice the positive/neutral class to increase the performance of the negative class.

The nature of the data could be different across various domains. For example, in Curious Cat and ask.fm data, informal language is used more often than Kaggle and Wikipedia. Therefore, the type of embeddings we use in our experiments could be an important factor for the final performance. We plan to use BERT language model in our experiments as the embeddings; however, we prefer not to fine-tune BERT weights because of the computational cost. Therefore, we run our BiLSTM + RA baseline with the two following embedding models to see which one works best across all corpora:

1. 200-dimensional $Glove$[9] embeddings trained on Twitter
2. $BERT_{base}$ (uncased) contextualized embeddings trained on the BookCorpus and English Wikipedia corpus (Devlin et al., 2019).[10]

Based on the results shown in Table 3, it seems that BERT performs better than Glove embeddings across all datasets, despite the fact that we do not fine-tune its weights. Therefore, we use BERT as the embeddings in the rest of the experiments.

	Glove		BERT	
	F1	W F1	F1	W F1
Curious Cat	60.16	87.1	**65.29**	**88.2**
ask.fm	51.69	83.9	**52.44**	**84.0**
Kaggle	69.73	85.1	**75.12**	**87.0**
Wikipedia	75.60	95.3	**79.21**	**95.9**

Table 3: Comparison between Glove and BERT embeddings using BiLSTM + RA baseline. We do not fine-tune BERT in our experiments and only use it as a feature extractor.

Table 4 compares the performance of GEA and RA attention mechanisms. For Curious Cat and ask.fm corpora, BiLSTM + GEA model performs significantly[11] better than BiLSTM + RA, which demonstrates the effectiveness of our proposed attention to detect offensive language in short and noisy texts. BiLSTM + RA shows slightly better performance on Kaggle, as well as significant improvement on Wikipedia datasets in comparison with BiLSTM + GEA. This observation could be explained by the following reason: the length of documents are longer in Kaggle and Wikipedia compared to Curious Cat and ask.fm. Therefore, the DeepMoji module which is trained on short tweets has probably some difficulties to generate the emotion representation for Kaggle and Wikipedia data.

	BiLSTM + RA		BiLSTM + GEA	
	F1	W F1	F1	W F1
Curious Cat	65.29	88.2	**72.22***	**90.9***
ask.fm	52.44	84.0	**60.70***	**85.2***
Kaggle	**75.12**	**87.0**	74.98	86.7
Wikipedia	**79.21***	**95.9***	77.15	95.5

Table 4: Comparison between RA and GEA attention models. The starred results show significant improvement compared to the opposite model.

Table 5 shows the classification results, including the performance of our proposed models, Baselines, and state-of-the-art approaches across all four different corpora. For the Curious Cat data, DeepMoji Baseline shows very promising results. This model performs significantly better than fine-tuned BERT (Bodapati'19), which shows the power of DeepMoji representations. Combining the text and emotion information through either BiLSTM + RA + DeepMoji, or BiLSTM + GEA models produces results that are slightly better than DeepMoji Baseline.

For the ask.fm corpus, BiLSTM + RA + DeepMoji and BiLSTM + GEA + DeepMoji indicate almost similar performance. The former performs slightly better on the negative/offensive class (showing a higher F1), while the latter works better on the positive/neutral class (having a higher weighted F1, as well as a very promising F1). The reported results for both models are significantly better than the state-of-the-art results on ask.fm (Sam'17), DeepMoji baseline, and fine-tuned BERT (Bodapati'19) that prove the effectiveness of our proposed approaches to integrate emotion information into the textual representation.

For Kaggle, Bodapati'19 reports best results. However, the performance of that model compared to our best model, BERT Baseline + DeepMoji, is not significantly better under the Mcnemar test. Although none of our main models (BiLSTM + EA + DeepMoji and BiLSTM + GEA) is the winner for Kaggle, still, the best performing model across our proposed approaches and baselines (i.e., BERT Baseline + DeepMoji) has DeepMoji as part of its

[9] https://nlp.stanford.edu/projects/glove

[10] We only use BERT as a feature extractor.

[11] All the significant testing are done using Mcnemar test.

Model		Curious Cat		ask.fm		Kaggle		Wikipedia	
		F1	W F1	F1	W F1	F1	W F1	F1	W F1
DeepMoji Baseline		71.90	91.0	59.21	85.1	73.45	86.0	72.20	94.5
BERT Baseline	-	40.86	81.6	37.29	80.1	64.72	81.4	50.84	89.6
	+ DeepMoji	70.17	89.9	60.79	85.6	76.50	87.6	73.24	94.9
BiLSTM + RA	-	65.29	88.2	52.44	84.0	75.12	87.0	79.21	**95.9**
	+ DeepMoji	72.05	**91.1**	**62.40**	85.7	76.06	87.7	78.35	95.7
BiLSTM + GEA	-	**72.22**	90.9	60.70	85.2	74.98	86.7	77.15	95.5
	+ DeepMoji	71.09	90.1	62.12	**86.0**	75.47	87.4	77.86	95.7
Sam'17		65.54	88.3	58.47	84.1	72.85	86.0	74.48	94.7
Kaggle Winner		65.86	90.0	51.49	84.4	72.03	86.5	74.45	95.2
Bodapati'19		68.19	89.9	56.38	85.0	**76.86**	**88.5**	**80.13**	**95.9**

Table 5: Classification results in terms of F1-score for the negative/offensive class and weighted F1. +DeepMoji refers to the experiments in which we directly concatenated DeepMoji vectors with the last hidden representation generated by the model.

architectures. This model significantly outperforms the Kaggle Winner results as well.

For Wikipedia, Bodapati et al. (2019) report the weighted F1 of 95.7 as the state-of-the-art results. However, when we re-implement their model, we achieve a slightly better weighted F1 of 95.9 as what we report in Table 5. Although we achieve the same weighted F1 of 95.9 with BiLSTM + RA model, we can see that the F1 for the offensive class is around 1% worse than Bodapati'19, indicating that our model probably works better for the neutral class. For this corpus, it seems that integrating the emotion information into the model decreases the performance, which is inline with what we observe in Table 4. A possible reason for this is that the Wikipedia corpus, in nature, is very similar to the data used for pre-training BERT, and is very different from the Twitter data used for pre-training DeepMoji. Therefore, in this case, the text representation generated by BERT is more powerful than the DeepMoji representation. Then, combining these two representations does not improve the results.

Overall, we can conclude that:

1. For short and noisy text data like Curious Cat and ask.fm, integrating the emotion information (by DeepMoji representation) into the textual representation produces the best results in comparison with all other baselines. It demonstrates the advantages of using DeepMoji representation to extract contextual information from online content. The reason is that DeepMoji considers fine-grained emoji categories, which capture different levels of emotional feelings (e.g., 😡, 😤, and 😠show different levels of anger). Such information helps the model to determine the tone of language more precisely. In Section 5.3, we provide a more detailed analysis of the DeepMoji model.

2. For Kaggle and Wikipedia data that are longer and more structured, fine-tuned BERT (Bodapati'19) is the winner. However, the results reported by this model are not significantly better than our best performing approaches (i.e., BERT Baseline + DeepMoji for Kaggle, and BiLSTM + RA for Wikipedia). It should be noted that unlike Bodapati'19, we do not fine-tune BERT (fine-tuning BERT is computationally expensive, especially on large corpora like Wikipedia), which is a good achievement.

3. There are major differences between the performances of different models across the various datasets that we use. This observation shows that it is very challenging to build a model that works well in different domains. It also confirms the need to collect more data from a variety of social media platforms.

5.3 Why Does DeepMoji Work?

To show why emoji representations are helpful to detect the abusive language in social media, we plot the emoji distribution over the neutral and offensive classes for the Curious Cat training data (Figure 4). For creating this plot, we use the average DeepMoji vector extracted for each instance. This vector shows the relevance of each emoji to a specific comment. We create the overall emoji vector per class by averaging the emoji vectors extracted for all of the instances of the same class. Finally, we select 19 out of the 64 emojis used in the DeepMoji project to create the final plot. As it is shown in Figure 4, there are different patterns visible for the neutral and offensive classes. This

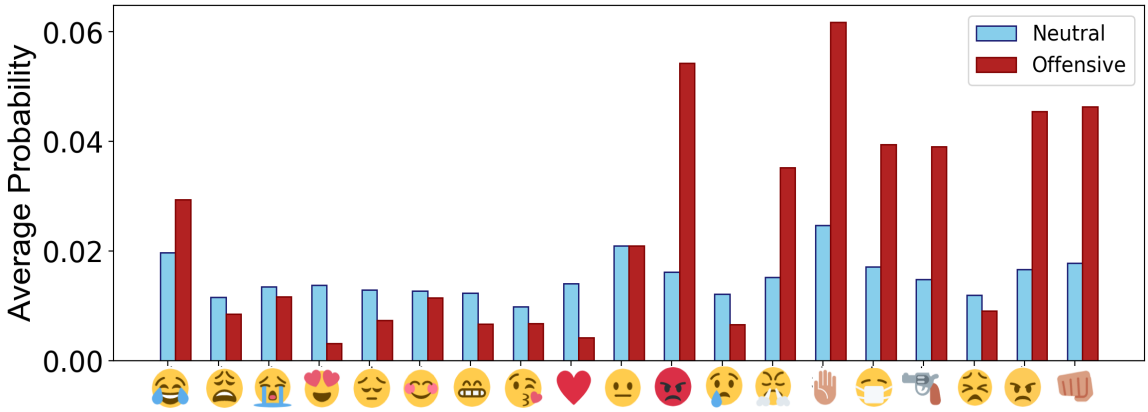

Figure 4: Emoji distribution over Curious Cat data.

observation validates our hypothesis on why it is useful to incorporate emoji information into the model.

Based on Figure 4, angry emojis (😡, 😤, 😠) are highly correlated with the offensive class, inversely happy and love faces (😁, 😍, ❤️) appeared more frequently in the neutral class. For the happy and love faces, 😊and 😘, the differences between offensive and neutral classes are much less. We believe that this represents the scenarios where a defender (a user who defends the victim of online attacks) tries to support an attacked user by complimenting him/her, while expressing their hatred towards the attackers. Sad faces (😩, 😭, 😔, 😢, 😣) are more frequent in neutral instances than offensive ones. It possibly shows the cases where a user expresses his/her unhappiness in response to an attack. Interestingly, the laughing face, 😂, shows a higher probability for the negative class. This can be linked to the scenario where someone attempts to bully a user by mocking him/her. Additionally, the plot shows exactly the same probabilities for the poker face (😐) over the offensive and neutral classes. So, we can conclude that this emoji does not convey any additional information related to offensive language. Other emojis (✋🏽, 😇, 👊🏽, and 🔫) that indicate the violent and threatening behavior towards the receiver also seem to appear in the offensive class frequently.

6 Conclusion and Future Work

In this paper, we create a new resource for the task of abusive language detection that does not focus on specific list of bad words. We also propose two different approaches for incorporating emotion information into textual representation by presenting end-to-end deep neural models that show very promising results across three existing corpora, and our new corpus for abusive language detection. Based on the results, adding emotion information to the model can improve the performance, especially for short and noisy textual data. As for the future work, due to the fact that perceived level of aggression is very subjective to the user, we plan to jointly model the question and answer within a pair for the Curious Cat and ask.fm data. We believe that the reply that the user provides in response to a received question/comment is a strong indicator whether it was offensive or neutral towards the user. Another possible path in order to move the research forward, is to expand this task to the detection of cyberbullying incidents which has also become a growing concern in online communities.

References

Wafa Alorainy, Pete Burnap, Han Liu, Amir Javed, and Matthew L Williams. 2018. Suspended accounts: A source of tweets with disgust and anger emotions for augmenting hate speech data sample. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, pages 581–586. IEEE.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Sravan Bodapati, Spandana Gella, Kasturi Bhattacharjee, and Yaser Al-Onaizan. 2019. Neural word decomposition models for abusive language detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 135–145, Florence, Italy. Association for Computational Linguistics.

Michele Corazza, Stefano Menini, Pinar Arslan, Rachele Sprugnoli, Elena Cabrio, Sara Tonelli, Serena Villata, and Fondazione Bruno Kessler. 2018. Comparing different supervised approaches to hate speech detection. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:230.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, pages 512–515. AAAI Press.

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

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):18.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1615–1625. Association for Computational Linguistics.

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

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

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*, pages 260–266.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Lisa M Jones, Kimberly J Mitchell, and David Finkelhor. 2013. Online harassment in context: Trends from three youth internet safety surveys (2000, 2005, 2010). *Psychology of violence*, 3(1):53.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Anna Koufakou and Jason Scott. 2020. Lexicon-enhancement of embedding-based approaches towards the detection of abusive language. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 150–157.

Ricardo Martins, Marco Gomes, José João Almeida, Paulo Novais, and Pedro Henriques. 2018. Hate speech classification in social media using emotional analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 61–66. IEEE.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019a. Abusive language detection with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2145–2150. Association for Computational Linguistics.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019b. Author profiling for hate speech detection. *CoRR*, abs/1902.06734.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019c. Tackling online abuse: A survey of automated abuse detection methods. *CoRR*, abs/1908.06024.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

John Edison Arevalo Ovalle, Thamar Solorio, Manuel Montes-y-Gómez, and Fabio A. González. 2017. Gated multimodal units for information fusion. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Santhosh Rajamanickam, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Joint

modelling of emotion and abusive language detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4270–4279, Online. Association for Computational Linguistics.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth International AAAI Conference on Web and Social Media*.

Niloofar Safi Samghabadi, Suraj Maharjan, Alan Sprague, Raquel Diaz-Sprague, and Thamar Solorio. 2017. Detecting nastiness in social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 63–72.

Niloofar Safi Samghabadi, Deepthi Mave, Sudipta Kar, and Thamar Solorio. 2018. Ritual-uh at TRAC 2018 shared task: Aggression identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 12–18. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678.

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.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 672–680.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy. Association for Computational Linguistics.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1046–1056, New Orleans, Louisiana. Association for Computational Linguistics.

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

Michele L Ybarra and Kimberly J Mitchell. 2004. Youth engaging in online harassment: Associations with caregiver–child relationships, internet use, and personal characteristics. *Journal of adolescence*, 27(3):319–336.