Flexible Inference for Cyberbully Incident Detection

Haoti Zhong[1], David J. Miller[1(✉)], and Anna Squicciarini[2]

[1] Electrical Engineering Department, Pennsylvania State University, State College, PA 16802, USA
hzz113@psu.edu, djmiller@engr.psu.edu

[2] College of Information Sciences and Technology, Pennsylvania State University, State College, PA 16802, USA
acs20@psu.edu

Abstract. We study detection of cyberbully incidents in online social networks, focusing on session level analysis. We propose several variants of a customized convolutional neural networks (CNN) approach, which processes users' comments largely independently in the front-end layers, but while also accounting for possible conversational patterns. The front-end layer's outputs are then combined by one of our designed output layers – namely by either a max layer or by a novel sorting layer, proposed here. Our CNN models outperform existing baselines and are able to achieve classification accuracy of up to 84.29% for cyberbullying and 83.08% for cyberaggression.

1 Introduction

Cyberbullying, along with other forms of online harassment such as cyberaggression and trolling [22] are increasingly common in recent years, in light of the growing adoption of social network media by younger demographic groups. Typically, a cyberaggression incident refers to a negative comment with rude, vulgar or aggressive content. Cyberbullying is considered a severe form of cyberaggression, with repeated cyberaggression incidents targeting a person who cannot easily self-defend [5,24].

To date, researchers from various disciplines have addressed cyberbullying (e.g. [10,28]) via detection and warning mechanisms. A growing body of work has proposed approaches to detect instances of bullying by analysis of individual posts, focused on detection of rude, vulgar, and offensive words. Recently, acknowledging that offensive messages are not solely (or always) defined by the presence of a few selected words, studies have also considered lexical features and sentence construction to better identify more subtle forms of offensive content [4,7]. Yet, despite some promising results, previous works typically ignore other characteristics of bullying, such as its repetitive and targeted nature [22]. As such, previous work is typically unable to distinguish between bullying and mere isolated aggressive or offensive messages, oversimplifying the cyberbullying

U. Brefeld et al. (Eds.): ECML PKDD 2018, LNAI 11053, pp. 356–371, 2019.
https://doi.org/10.1007/978-3-030-10997-4_22

detection problem. We believe a better way to detect both cyberbullying *and* cyberaggression is to consider the contextual cues surrounding these incidents, as they are exposed in a conversation. Accordingly, we focus on session-level detection of cyberbullying and cyberaggression, particularly sessions generated in response to posted media, e.g. images.

Our work aims at answering the following questions:

- Can we detect both cyberaggresion and cyberbullying based on a common model structure?
- Can we detect cyberbully incidents at the session level, rather than simply identifying individual aggressive comments?
- Can session-specific elements (e.g. the image, or the caption of the image) help improve inference of session-level bullying episodes?

Note that our research questions are intentionally focused on *session-level* inference. While in some cases bullying may be tied to a person (e.g. a single account) rather than the content of their messages, we here omit observations related to the specific user's history and patterns within a network (i.e. we do not consider social network features or users' features). We investigate the above questions by developing customized Convolutional Neural Network (CNN) models, with a single convolutional layer, that can be trained to attempt to fulfill the above requirements. The model performs a "session-level" analysis and takes all comments in a session as input. Here, a session is an Instagram-like session, with a thread of replies created after an initial post of an image+caption.

Our CNNs process individual users' comments, while also accounting for possible conversational patterns (a comment and its referrant). The CNN outputs, one per comment, are combined by one of our designed output layers – namely by either a max layer or by a novel sorting layer, proposed here. The sorting layer is a generalization of a max layer, which takes all comments into account according to their probabilities of being aggressive.

We test multiple variants of our proposed CNN architecture, training it *both* for bullying and aggression detection. Compared to prior work, our proposed approach is more powerful and comprehensive. Unlike previous works [16], our approach provides flexible inferences – it can distinguish sessions affected by cyberbullying from those affected by cyberaggression and also possibly identify the victim of cyberbullying in a session (be it the poster *or* one of the commenters). Further, it is truly designed to detect cyberbullying as it unfolds during a conversation (it can be applied to make progressive detection decisions, as more and more comments are made), unlike a simple offensive speech detector.

The paper is organized as follows. The next section summarizes related work. Section 3 presents a detailed description of our network. Section 4 describes all the datasets we used for our experiments. Section 5 compares the performance of our models with baselines, then gives insights into the nature of cyberaggression and cyberbullying. Lastly, Sect. 6 discusses our findings and potential extensions of this work.

2 Related Work

Studies in psychology and sociology have investigated the dynamics of cyberbullying, bullies' motives and interactions [2,14,18,30,31,37]. In particular, a number of methods have been proposed for detecting instances of cyberbullying, most focused on offensive textual features [4,9,21,36] (e.g., URLs, part-of-speech, n-grams, Bag of Words as well as sentiment) [6,10,29]. Several recent studies have also begun to take user features into consideration to help detect bullies themselves [3,6,12]. [17] conducted a simple study hinting at the importance of social network features for bully detection. They considered a corpus of Twitter messages and associated local *ego-networks* to formalize the local neighborhood around each user. Here, however, we focus on inference in the absence of such information, *i.e.*, purely based on the snapshot offered by one session.

[16] provides an initial effort on the detection of bullying and aggressive comments. The authors tested several text and social features and found that the counts of 10 predefined words offered the best results. Our work not only significantly increases the overall performance compared to this prior work, but also naturally gives the capability to identify the aggressive comments within a cyberbullying incident.

Our methodology is related to the recent body of work on CNN and deep neural net (DNN) models for textual analysis. CNNs were first designed for image-based classification [25], and have more recently been applied to various domains in NLP including document classification and sentence modeling [19,23]. Many efforts have focused on learning word or sentence vector representations [1,8,27], converting a word or sentence into a low dimensional vector space in order to overcome the curse of dimensionality and to allow calculation of similarities between words. Also many works use DNNs to classify a sentence or document, e.g. [13,33]. Specifically, Kim et al. [20] use a CNN-based model for sentence classification tasks. CNNs use a filtering (convolutional) layer to extract spatially invariant low-level feature "primitives". Subsequent layers perform spatial pooling of primitive information, and also pooling across primitive types. This is followed by several fully connected layers, leading to an output layer that predicts the class for the sentence/word (e.g., its sentiment).

Here, we propose a CNN approach (without a great number of layers), rather than a DNN approach. This is because CNNs (without many layers) are naturally *parsimonious* in that there is weight sharing and therefore they are more suitable for applications like ours, where there is a limited amount of available training data. The scarcity of labeled cyberbullying data for training does not support use of DNNs with many layers and a huge number of free parameters to learn.

3 Approach

3.1 CNN for Session-Level Bully Incident Detection

In designing our approach, we rely on commonly accepted definitions of cyberbullying and cyberaggression, consistent with recent literature in the field [5,16].

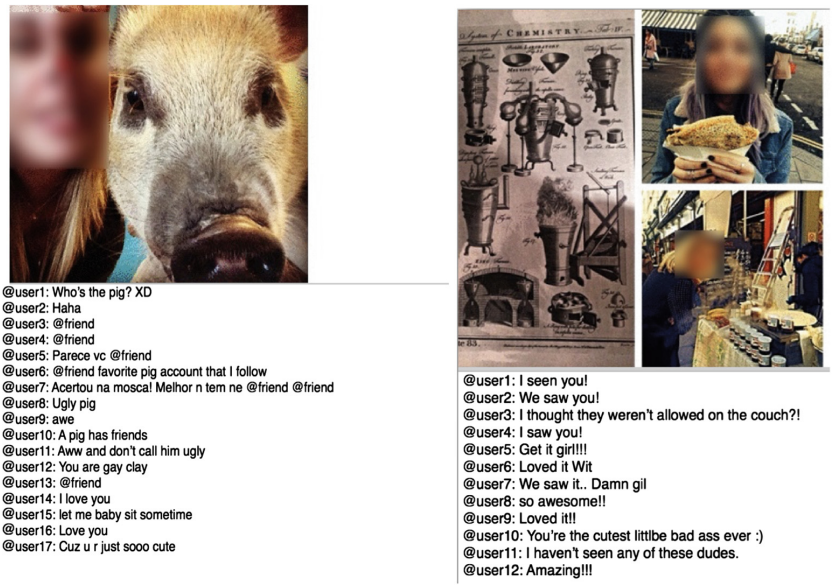

Fig. 1. Examples of sessions with cyberbullying (left) and without (right).

We view cyberbullying as repeated acts of explicit targeted aggression, whereas cyberaggression refers to occasional vulgar or rude attacks in isolation. Accordingly, we exploit the fact that cyberbullying events are considered a subset of cyberaggression.[1] In Fig. 1, we report examples of two sessions, one with instances of bullying, and one that is instead *not* affected by bullying.

Therefore, in order to identify a cyberbullying incident, we should observe *multiple* negative comments within a conversation, all related to the same victim (i.e. a single account denoting an individual or group). We consider this in the context of a *session*, a collection of temporally sorted comments related to the same topic/discussion. Since there are multiple comments in a session, if we simply train a detector of offensive or aggressive comments, the false positive rate may be close to 100% since in this case the probability of a session being bullied is the probability of at least one offensive comment, i.e. $P(bully) = 1 - \Pi_i(1 - p_i(bully))$ – even a small overestimated probability for comment detection may lead to a significant error rate at the session level. Also, this approach is inconsistent with the very definition of cyberbullying, as it ignores the targeted and repeated nature of these incidents.

Our model aims at detecting cyberaggresion first, treating this as a multi-instance learning problem. The assumption is that a bag of instances (i.e. a session) can be labeled as positive (i.e. inclusive of an aggressive comment) if there is at least one positive instance. Specifically, to detect aggressive incidents,

[1] As we discuss in Sect. 4, our experimental evaluation is based on a dataset labeled consistently with these definitions.

we maximize over all (probability) outputs of the aggressive comment detection CNN: $P_{incident}(agg) = Max_i(P_{sentence_i}(agg))$. By suitably generalizing this decision rule, our model can also be used to detect cyberbullying incidents, as will be explained in the following sections.

Our model includes two main components. The front-end component is a shared CNN model which learns how to classify sentences as "aggressive" or "non-aggressive". Here, "shared" refers to the fact that all sentences go through the same (a common) network. The back-end part of the model is the output layer, learning to detect cyberbullying incidents based on the outputs from the front-end model. These two components are not learned separately and simply combined together – all layers of our customized CNN are jointly learned, so as to maximize a training set measure of cyberbullying detection accuracy. Our CNN architecture with a sorting-based decision layer is shown in Fig. 2.

Next, we describe in more depth the main layers and configuration of our network. Note that in the design of our CNN, several hyper-parameters were chosen, including the top number of words, the dropout rate, the word embedding length, convolutional layer filters' number and size, pooling layer size, hidden layer's neuron number and activation function. Because of the extremely large set of possible combinations, we split the hyperparameters into two groups. For each group, we chose the optimized hyper-parameter combination by means of nested cross-validation (CV). We selected these hyperparameters once, and used the same configuration across all extensions and variants of our model.

3.2 Input Layer

The input to the comment-level network is a vector of indices where each index uniquely identifies one of the top 20,000 words (in terms of frequency of occurrence in the training set). Each such index is mapped to a vector via word2vec. We first preprocessed all the words by removing all non ASCII characters, then applied tokenizing and stemming. Since for any comment, we also know to which comment it replied (or if it is a direct reply to the post itself), this contextual information should also be exploited, in modelling each comment. To achieve this, we concatenate to the "responding" comment's index vector the index vector of the "referrant" comment. If there is no such comment, we simply concatenate a vector of 0s; if it is a response to the original posting, the referrant is the representation of the image's caption.

3.3 Shared CNN Model for Sentence Classification

Inspired by prior work [20], we first define a shared comment level aggression detection model. The output of this shared CNN will be a comment-level probability of cyberaggression, for each comment in a session.

The CNN is constructed as follows:

– *Dropout layer* [32]: Downsamples some of the features during training to increase the robustness of the model; the dropout rate is a hyper-parameter (not shown in Fig. 2 for simplicity).

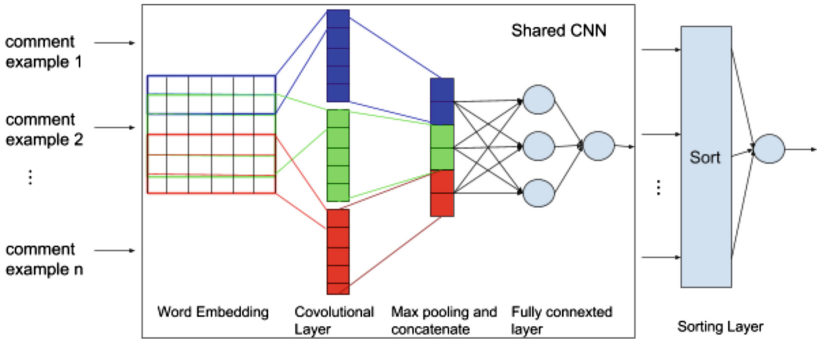

Fig. 2. Our CNN architecture, consisting of a shared CNN model applied to each comment, and an output (fusion) layer. In this case the fusion is based on sorting shared CNN outputs that represent the aggression probabilities for all sentences and feeding these sorted probabilities to a sigmoidal output activation layer. The decision layer is the sorting layer.

- *Embedding layer:* Applies a pre-trained word2vec [27] model to embed each word into a vector of length 100, so now each comment is a (25, 100) matrix, since we use a fixed number (25) of words for each comment – longer comments are truncated. Shorter comments are padded with repetition of a "null" word to fill out to length 25.
- *Convolutional layer*: Uses several filters (1, 3, 5, 8) to get a convolution response from the original matrix; ReLU activation functions (linear response if positive; zero if negative) are used to process these responses. A filter with length one encompasses only individual words, whereas if the length is increased to encompass multiple words what is learned is word-order dependent.
- *Max pool layer*: Downsamples the responses of the convolutional layer, and concatenates them into a single vector. We use a large pooling size (e.g. each sentence will pool to 1 output). This allows us to capture cases where a single offensive word or an aggressive phrase expresses a strong negative meaning, and also may reduce potential overfitting (since smaller pooling sizes require more model parameters).
- *Concatenation layer*: Concatenates all channel's outputs into a single vector.
- *Fully connected layer*: A hidden layer like a traditional MLP hidden layer with 100 neurons using ReLU activations.
- *Output layer*: 1 unit output using a sigmoid activation function, which evaluates the probability that a comment is aggressive.

3.4 Decision Layers for Bully Incident Detection

Max Layer: The main idea of the max layer is to identify the comment with the maximum aggression probability in each session; on the training set, we essentially learn a good "threshold" for discriminating between the maximum prob-

ability for a cyberaggression (cyberbullying) session, and for a non-aggression session. Strict gradient descent-based learning for cost functions that depend on a max() operator are problematic due to the discontinuous nature of the max function. However, using the *pseudo-gradient* for the max() function introduced in [34], we can update the model at each iteration based on the comments in each session with the max probability (there may be more than one) as the *effective* training set for this (gradient-based) iteration.

Training to minimize a sum of (Kullback-Leibler) KL divergences, the model is effectively learning to *minimize* the maximum probability of aggression over the comments in a non-bully session. Also, since there are possibly multiple aggressive comments in a given session, we further find the comment with second highest probability if its value is higher than a certain ratio compared with the maximum probability and we take these two top comments into account in the training process. This additional information is expected to boost the performance of the CNN. The learning procedure is shown in Algorithm 1. The final model was chosen by maximizing the average validation accuracy. In the experiment we found the model tends to overfit after a few iterations; thus we set the maximum iteration number as 10.

During the inference phase, the model's output for each session can be obtained by calculating the max probability over all comments in the session, and this probability is treated as the probability of a cyberaggressive incident in a session. Moreover, our trained cyberaggresion model can be used as a cyberbully incident detector by changing the classification rule to adhere more strictly to the definition of cyberbullying: i.e. at least two comments with a probability greater than 0.5 must target the same user (either the original poster *or* a commenter).

Algorithm 1: Train the cyberaggression incident detector CNN

initialize parameter θ of CNN f(x);
iter = 0;
while $iter \leq MaxIter$ **do**
  $iter$ += 1;
  $training\_data$ = [];
  $training\_label$ = [];
  for *each session s* **do**
    $idx1 = \arg\max_{c \in s} f(x_c|\theta)$
    $idx2 = \arg\max_{c \in s, c \neq idx1} f(x_c|\theta)$
    $training\_data.append(comment\ idx1)$;
    $training\_label.append(label_s)$
    if $f(x_{idx2}|\theta) \leq 0.9 * f(x_{idx1}|\theta)$ **then**
      $training\_data.append(comment\ idx2)$;
      $training\_label.append(label_s)$
    end
  end
  save model if it achieves the best validation accuracy;
  update θ *with ADAGRAD[11];*
end

Sorting Layer: Generalizing beyond the "top two" comments in a session, we propose a novel *sorting layer* which exploits the *top* K comments (those with the K largest probabilities of aggression) in learning a multi-instance CNN. Each comment goes through a shared (front-end) model (the same as the model we described before), which produces the comment's probability of aggression. These probabilities are then sorted, with only the top K retained (K a hyper-parameter) and fed into a final sigmoid output layer. The learned sorting layer will tend to give the comment more "weight" if it is an aggressive comment, and zero weight if it is *uninformative*– e.g., if $K = 10$, the learning for this layer may determine that only the top four weights to the output layer are non-zero. In such case, only the top 4 comments (in terms of aggression) are found useful for discriminating bullying from non-bullying sessions. Likewise, if e.g. the 7-th ranked comment has a low probability of being labeled as aggressive, this information is likely neutral as to whether the session involves bullying or not, and the weight on this comment may thus be learned to be zero. The learning phase for this model (i.e. the CNN with sorting layer) is a natural generalization of the learning approach described above involving max() functions, just as a sorting function itself is a natural generalization of a max() function. Similar to the previous (max-based) model, this sorting-based model can be trained to predict either cyberbullying (versus negation) or cyberaggression (versus negation), depending on the supervising labels that are chosen. The pseudo-code for the algorithm is reported in Algorithm 2. Note that we set the maximum iteration number as 50, larger than for Algorithm 1, in order to allow all the parameters (more than for Algorithm 1) to be reasonably learned.

Algorithm 2: Train the bully incident detector sorted-CNN

```
initialize parameter θ of shared CNN f(x);
iter = 0;
while iter ≤ MaxIter do
    training_data = [];
    training_label = [];
    for each session s do
        keys=[];
        for each comment c do
            keys.append(f(x_c|θ)
        end
        indexs=argsort(keys);
        new_session=[];
        for i in indexs and i ≤ 30 do
            new_session.append(c_i)
        end
        training_data.append(new_session);
        training_label.append(label_s)
    end
    save model if it achieves the best validation accuracy;
    update full model with ADAGRAD;
end
```

4 Datasets

The dataset we used for our experiments is taken from [16], which was collected from Instagram posts. The authors collected Instagram sessions with more than 15 comments so that labelers could adequately assess the frequency or repetition of aggression. Labelers were asked to label the whole post, whether or not there is (1) a cyberaggression incident and/or (2) a cyberbullying incident. For each Instagram post, the dataset includes images, captions, poster's id, the number of followers, all comments and all commenters' ids and post times. In this paper, we call each post a session. Most comments are short, with few words. The dataset has 1540 non-bully sessions and 678 bully sessions. It likewise has 929 aggression sessions and 1289 non-aggression sessions (consistent with cyberbullying being a subset of cyberaggression).

To validate our claim that a comment level aggressive detector is not suitable for detecting cyberbullying incidents, we trained a CNN baseline with a comment level aggression dataset, which was taken from [38]. (We will report results for this method in the next section). The dataset was crawled from publicly visible accounts on the popular Instagram social platform through the site's official API, and had a subset of the comments labeled for aggression. Labelers had access to the image, the image's commentary, and indicated whether or not each comment represented aggression. Overall, the training dataset for this model includes 1483 non aggressive comments and 666 aggressive comments.

5 Experiments

In order to validate our model, we carried out a number of experiments. We validate the accuracy of our CNN frameworks, and investigate whether additional inference power can be gained by exploiting the session's uploaded image (which triggered the session). We compare accuracy of our approach with two baselines, discussed next.

5.1 Baselines

We considered two main baselines for comparative purposes.

First, we replicated [16]. In this work, several text and image features were tested to find out the best features for detecting evidence of cyberbullying within a session. The authors eventually claimed that using the count for 10 specific (mostly vulgar) words as the features input to a logistic regression model provided the best performance. The model is directly trained with cyberbully labels applied at the session level.

As a second baseline, we used the same CNN architecture described in Sect. 3.3, but trained it with an Instagram dataset labeled at the comment level for aggression [38] (see Sect. 4 for a description). We call this the COMMENT CNN. In this case, given a comment label {aggressive,non-aggressive}, we assess

the accuracy of detection for a *cyberbully* incident. After training as a comment-level detector, we refine the classification rule during testing so that, only if there is more than one aggressive comment targeting the same user, the session is declared "bullied". Testing is performed on [16]'s dataset.

5.2 Variants of Our Model

In addition to comparing with state-of-the art baseline models, we experiment with several variants of our proposed model. For all the model variants (besides the CNN) each comment is concatenated with the comment to which it replied, as input to the shared CNN structure.

- *CNN:* The shared CNN is coupled with a max function in the decision layer and takes all comments as input. Each comment is separately input to the shared CNN structure. The model is trained with session level bully labels.
- *CNN-R:* The shared CNN (using comments and their referrant comments) is coupled with a max function in the decision layer. The model is trained with session level bully labels.
- *SORT CNN-R:* The shared CNN is coupled with a sorting function in the decision layer followed by a sigmoidal nonlinearity. The front-end model parameters are initialized using the trained CNN-R model. This model is trained with session level bully labels.
- *ACNN-R:* The shared CNN is coupled with a max function in the decision layer. The model is trained with session level *aggression* labels. We can use this model directly to detect cyberaggression. Alternatively, for cyberbully detection, during testing, we can replace the max layer by a layer that checks whether there are at least two comments with a front-end output greater than 0.5 which responded to the same user.
- *SORT ACNN-R:* The shared CNN is coupled with a sorting function followed by a sigmoidal nonlinearity in the decision layer. Front-end model parameters are initialized using the trained ACNN-R model. This model is trained with session level aggression labels.

5.3 Results for Session-Level Cyberbullying Detection

Results for cyberbully detection of our models are reported in Table 1. Cross validation was used; the dataset was divided into 10 (outer) folds, and 8 of these folds were used for training, one for validation, and the last fold for testing. As shown, the Hosseinmardi et al. baseline approach achieves a True Positive Rate (TPR) of 67.98% and True Negative Rate (TNR) of 75.01%. The COMMENT CNN gives a TNR of 27.39% and TPR of 97.63% – it performs as expected, since even a small overestimated probability of comment-level aggression will lead to a large false positive rate (low false negative rate). All variants of our CNN models consistently outperform Hosseinmardi et al.'s method, as indicated next.

CNN achieves a TPR of 73.06% and a TNR of 86.44%. The biggest advantage over the baseline is that our model captures not only sensitive words indicative

Table 1. Performance for cyberbully detection CNNs on [15] dataset.

Classifier	Overall accuracy	TNR	TPR	F1-measure
Baseline [16]	71.5%	75.01%	67.98%	0.7132
COMMENT CNN	62.51%	27.39%	97.63%	0.4278
CNN	79.75%	86.44%	73.06%	0.7919
CNN-R	81.25%	89.83%	72.67%	0.8034
SORT CNN-R	82.05%	88.89%	75.2%	0.8147
ACNN-R	82.92%	82.24%	83.59%	0.829
SORT ACNN-R	84.29%	83.24%	85.33%	0.8427

of a cyberaggression attack, such as the 10 words used by the baseline, but also information from the whole sentence.

As mentioned in Sect. 3.3, in order to capture contextual cues about the ongoing discussion, we concatenated the target comment with the comment to which the target comment replied and trained CNN-R. This simple change increases overall accuracy by about 1.5%, showing the importance of leveraging insights from users' conversational patterns. SORT CNN-R is more effective in detecting bully incidents than directly choosing the max aggressive comment probability as the session level output. SORT CNN-R increases accuracy by about 1% compared with CNN-R (81.25% vs. 82.05% for SORT CNN-R). However, likely due to an insufficient number of training posts, SORT CNN-R tends to overfit if we use all the comments in a session. Thus, we only use the top 30 comments, with the highest aggressive probabilities, and also stop model training once training accuracy exceeds validation accuracy. We also note that if we train a CNN with aggression labels rather than cyberbullying labels, using either max layer (ACNN-R) or sorting layer (SORT ACNN-R), we consistently obtain an overall accuracy gain of about 2% for *cyberbullying* detection. One of the reasons is that the aggression domain is more class-balanced than the bullying domain. This is likely helping the trained model to better learn the characteristics of negative comments since there are more negative incidents from which the model can learn, compared with the case of cyberbullying labels.

5.4 Applying a CNN Model for Cyberaggression Detection

We also explored whether our CNN model could help detect cyberaggression. We compare the performance of two variants of our model: ACNN-R with a modified logistic classification rule (three classes: no detection, cyberaggression, cyberbullying), and SORT ACNN-R.

Note that ACNN-R is able to detect *both* cyberaggression and cyberbully sessions, simultaneously. We achieve a TPR of 85.13% and a TNR of 80.64% for cyberaggression detection with ACNN-R, and the overall accuracy for cyberbullying detection is 82.92%. With SORT ACNN-R for cyberaggression detection, we achieve a TPR of 82.44% and a TNR of 83.71%. This result shows our model's

ability to detect incidents of cyberaggression. Since cyberaggression occurs when there is even just one aggressive comment, the sorting layer, as expected, is not very necessary for this task (Sort ACNN-R does not outperform ACNN-R here).

5.5 Image-Related Information for Cyberbully Detection

We also investigated several approaches for integrating image (i.e. visual content) information within the CNN model. We tested both a model that included the image features concatenated in our model's hidden layer, and a model version wherein we let the model's convolutional layers predict the image content as an additional task. However, we did not find image information to be helpful in improving our detection accuracy. We validated these two approaches over 1142 available full Instagram sessions (i.e. images and comments were available), which include 763 non-bully and 379 bully sessions. We applied these two ideas into our ACNN-R model. The image features concatenated in the hidden layer yields to a decrease of TNR and TPR to 73.68% and 58.83%, respectively. A multi-task approach instead achieves a TNR of 74.11% and TPR of 88.09% and still shows no significant gain or harm compared to the original inference power of the convolutional layers for the text-only CNN.

5.6 Insights on Cyberbullied Sessions and Their Context

In order to gather additional insights on the correlation between images and comments, we carried out two additional experiments. First, we generated an image caption according to Xu's recent approach [35]: images' visual features are extracted from a CNN model, then a Recurrent Neural Network (RNN) converts these features into word sequences. We then calculated sentence similarity between the detected aggressive comments and the image's generated caption[2]. We analyze similarity distribution using sentence similarity based on semantic nets and corpus statistics [26].

Second, we calculated the similarity distribution between the aggressive comment and the caption written by the image poster. As shown in Fig. 3, left, both similarity distributions have a low average (0.2595 and 0.1900, respectively), indicating that there is no strong relationship between aggressive comments and the posted content in general. This may be due to the (skewed) nature of the dataset – as discussed in Sect. 4, only sessions with a sufficiently large number of comments were crawled. This may have generated a dataset of sessions mainly from popular users and celebrities. In these cases, it is possible that users posting aggressive comments simply target the poster, not the posted content - or that responders are intrinsically aggressive.

To further gain insights on the conversations triggering bullying, we estimated how cohesive the posted comments are within a session (in terms of the content being discussed). To do so, we first applied the Twitter-to-vector model [8] to generate a fixed length vector for every comment. The Twitter-to-vector

[2] Two comments are considered similar if the model's output is greater than 0.5.

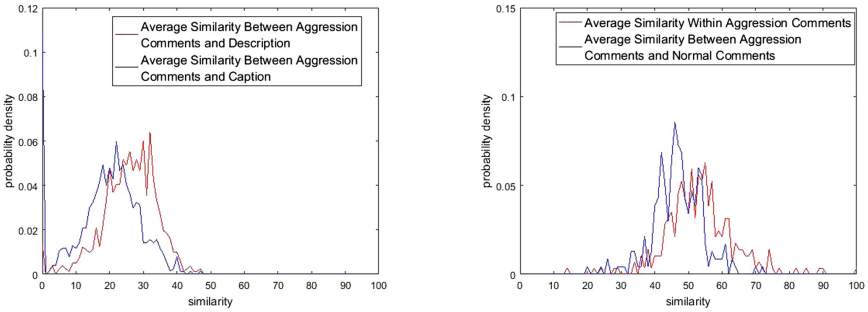

Fig. 3. Left: Similarity between aggressive comment and image captions. Right: Comment similarity comparison

model finds vector space representations of whole tweets by learning complex, non-local dependencies in character sequences. We calculated the average cosine similarity among all aggressive comments in a session (similarity is calculated pair-wise between all the aggressive comments), and compared with the average cosine similarity between aggressive comments and normal comments in the same session. As shown in Fig. 3, right, pair-wise comment similarity among aggressive comments is larger than pair-wise similarity between normal and aggressive comments. This partially supports the hypothesis that aggressive comments share similar content for a given post or bullies simply attack users in a similar and repeated fashion.

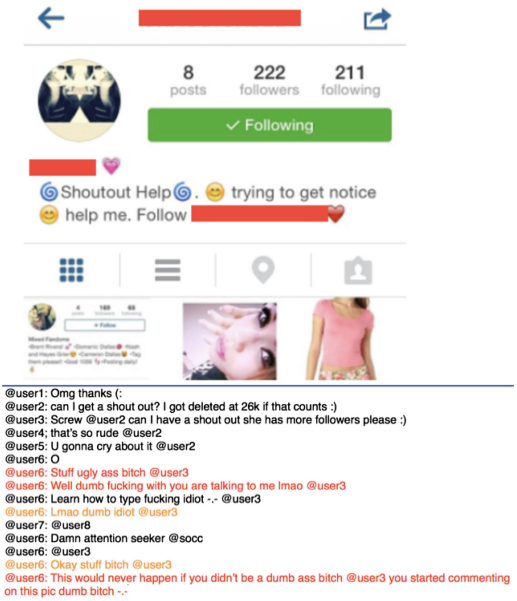

Fig. 4. Example of highlighted aggressive comments in a session

6 Conclusions and Future Work

In this work, we presented a session-level approach for detection of sessions affected by bullying or aggressive comments. We present several CNN-based models and demonstrate that our proposed models increase the average accuracy by about 13% compared with baselines. All of our models achieve similar performance in detecting cyberbullying and cyberaggression.

Our model lends itself to several interesting applications and extensions. One potential application is to explicitly infer which comment(s) are the likely triggers of bullying incidents. This is a natural extension of the current approach, as we already determine the probabilities of aggression for each of the top K comments. Moreover, each such comment has an earlier *referrant* comment. Thus, likely candidates for the trigger include: (1) the comment with greatest probability; (2) the referrant comment for the comment with greatest probability, if the referrant comment's probability itself is above a threshold; (3) the first comment with probability exceeding a given threshold.

We carried out an initial experiment with our CNN with max layer output. Figure 4 shows the outcome for a sample session. We highlighted the detected aggressive comments in red, and the comments with a probability greater than 0.25 as orange. As shown, our model correctly identifies most of the bully comments. We will continue along this direction and provide a systematic way to rank comments within conversations, not only with respect to their aggressive or bullying nature but also with respect to their specific "role" in the conversation.

An additional possible refinement includes further studying how to effectively leverage information contained in the images included in the analyzed sessions. Finally, future work also includes extending our model to support a more thorough distinction among aggressive comments (e.g. trolling vs harrassment) in a semi-automated manner.

Acknowledgements. Work from Dr. Squicciarini and Haoti Zhong was partly supported by the National Science Foundation under Grant 1453080 and Grant 1421776.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. JMLR **3**, 1137–1155 (2003)
2. Berson, I.R., Berson, M.J., Berson, M.J.: Emerging risks of violence in the digital age: lessons for educators from an online study of adolescent girls in the united states. J. Sch. Violence **1**(2), 51–71 (2002)
3. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., Vakali, A.: Mean birds: detecting aggression and bullying on twitter. arXiv preprint arXiv:1702.06877 (2017)
4. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: PASSAT, pp. 71–80. IEEE (2012)

5. Corcoran, L., Guckin, C.M., Prentice, G.: Cyberbullying or cyber aggression?: A review of existing definitions of cyber-based peer-to-peer aggression. Societies **5**(2), 245–255 (2015)
6. Dadvar, M., Trieschnigg, D., de Jong, F.: Experts and machines against bullies: a hybrid approach to detect cyberbullies. In: Sokolova, M., van Beek, P. (eds.) AI 2014. LNCS (LNAI), vol. 8436, pp. 275–281. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06483-3_25
7. Dadvar, M., Trieschnigg, D., Ordelman, R., de Jong, F.: Improving cyberbullying detection with user context. In: Serdyukov, P., et al. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 693–696. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36973-5_62
8. Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., Cohen, W.W.: Tweet2Vec: character-based distributed representations for social media. arXiv preprint arXiv:1605.03481 (2016)
9. Dinakar, K., Reichart, R., Lieberman, H.: Modeling the detection of textual cyberbullying. In: The Social Mobile Web (2011)
10. Djuric, N., Zhou, J., Grbovic, M., et al.: Hate speech detection with comment embeddings. In: WWW, pp. 29–30. ACM (2015)
11. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. JMLR **12**, 2121–2159 (2011)
12. Galán-García, P., de la Puerta, J.G., Gómez, C.L., Santos, I., Bringas, P.G.: Supervised machine learning for the detection of troll profiles in Twitter social network: application to a real case of cyberbullying. In: Herrero, Á., et al. (eds.) SOCO 2013-CISIS 2013-ICEUTE 2013, vol. 239, pp. 419–428. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-01854-6_43
13. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: a deep learning approach. In: ICML, pp. 513–520 (2011)
14. Hinduja, S., Patchin, J.W.: Social influences on cyberbullying behaviors among middle and high school students. J. Youth Adolesc. **42**(5), 711–722 (2013)
15. Hosseinmardi, H., Mattson, S.A., Rafiq, R.I., Han, R., Lv, Q., Mishra, S.: Detection of cyberbullying incidents on the Instagram social network. CoRR abs/1503.03909 (2015)
16. Hosseinmardi, H., Rafiq, R.I., Han, R., Lv, Q., Mishra, S.: Prediction of cyberbullying incidents in a media-based social network. In: ASONAM, pp. 186–192. IEEE (2016)
17. Huang, Q., Singh, V.K., Atrey, P.K.: Cyber bullying detection using social and textual analysis. In: SAM, pp. 3–6. ACM (2014)
18. Juvonen, J., Graham, S.: Bullying in schools: the power of bullies and the plight of victims. Annu. Rev. Psychol. **65**, 159–185 (2014)
19. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
20. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
21. Kontostathis, A., Reynolds, K., Garron, A., Edwards, L.: Detecting cyberbullying: query terms and techniques. In: WebSci, pp. 195–204. ACM (2013)
22. Kowalski, R.M., Limber, S.P., Limber, S., Agatston, P.W.: Cyberbullying: Bullying in the Digital Age. Wiley, Hoboken (2012)
23. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: AAAI, vol. 333, pp. 2267–2273 (2015)
24. Langos, C.: Cyberbullying: the challenge to define. Cyberpsychol. Behav. Soc. Netw. **15**(6), 285–289 (2012)

25. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
26. Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. **18**(8), 1138–1150 (2006)
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
28. Nahar, V., Unankard, S., Li, X., Pang, C.: Sentiment analysis for effective detection of cyber bullying. In: Sheng, Q.Z., Wang, G., Jensen, C.S., Xu, G. (eds.) APWeb 2012. LNCS, vol. 7235, pp. 767–774. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29253-8_75
29. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: WWW, Republic and Canton of Geneva, Switzerland, pp. 145–153. International World Wide Web Conferences Steering Committee (2016)
30. Rodkin, P.C., Farmer, T.W., Pearl, R., Acker, R.V.: They're cool: social status and peer group supports for aggressive boys and girls. Soc. Dev. **15**(2), 175–204 (2006)
31. Salmivalli, C., Isaacs, J.: Prospective relations among victimization, rejection, friendlessness, and children's self-and peer-perceptions. Child Dev. **76**(6), 1161–1171 (2005)
32. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. JMLR **15**(1), 1929–1958 (2014)
33. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: EMNLP, pp. 1422–1432 (2015)
34. Teow, L.-N., Loe, K.-F.: An effective learning method for max-min neural networks. In: IJCAI, pp. 1134–1139 (1997)
35. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: ICML, pp. 2048–2057 (2015)
36. Yin, D., Xue, Z., Hong, L., Davison, B.D., Kontostathis, A., Edwards, L.: Detection of harassment on web 2.0. In: Proceedings of the Content Analysis in the WEB, vol. 2, pp. 1–7 (2009)
37. Zalaquett, C.P., Chatters, S.J.: Cyberbullying in college. Sage Open **4**(1), 1–8 (2014). https://doi.org/10.1177/2158244014526721
38. Zhong, H., et al.: Content-driven detection of cyberbullying on the Instagram social network. In: IJCAI, pp. 3952–3958 (2016)