Early Detection of Cyberbullying Events in Online Social Media

NAME

Email address

NAME

Email address

1. INTRODUCTION

Nowadays, the number of active users on online social media is dramatically growing. Unfortunately, some people misuse these online environments to harass, threaten, humiliate and bully others. Cyberbullying is a threat and a nuisance. It mostly affects teens and pre-teens. According to a High School Youth Risk Behavior Survey, 14.8% of students in the United States reported being bullied electronically[1]. Also, the research done by the Cyberbullying Research Center[2] from 2007 to 2015 shows that on average, 26.3% of middle and high school students from across the United States have been victims of cyberbullying, and about 16% of the students have admitted that they have cyberbullied others at some point in their lives. Cyberbullying victims face social, emotional, physiological and psychological disorders that lead them to harm themselves and even commit suicide. Therefore, it is extremely important to detect cyberbullying cases before they cause irreparable damages to the victims.

Several works have been done towards the finding cyberbullying traces by detecting online hateful and aggressive comments, but most of these efforts are focused on forensic scenarios (i.e., after the cyberbullying took place), and cannot be used for prevention.

In this research, we present our initial approach to early detect cyberbullying cases using as few text as possible with as much anticipation as possible. In line with this goal, we first build a new corpus especially suited for early cyberbullying detection. Then we use various type of NLP features to build our predictive model.

2. RELATED WORK

Previous works in this field mostly focus on abusive language or hate speech detection in social media, and can be categorized to the following tasks:

- Presenting a new corpus created over different social media platforms like Twitter, ask.fm, YouTube, Wikipedia [1], [2], [3].
- Exploiting linguistic features (e.g. lexical, semantic, stylistic) for abusive language or hate speech detection [3], [4].
- Applying different deep learning models to detect online nasty messages [5].

To the best of our knowledge, none of the previous proposed models is able to detect cyberbullying events. The biggest limitation of those systems is that they are not designed to work as a dynamic mechanism to monitor streams of users' conversation. For this reason, early text classification strategies could be a solution to predict considering the risks of waiting for more evidence, and the cost and confidence of taking an action.

The early text categorization problem is an emerging research topic with scant works. Recently, the relevance of the problem has motivated specialized forums such as eRisk-CLEF17 [6]. One of the first attempts is based on processing documents at sentence level [7]. At every time t, the model reads the sentence and tries to determine the class of the whole document. More recently, [8] proposed a straightforward solution for early detection scenarios by using Naive Bayes classifier. The idea is to train the model over the full documents, and classify partial information.

3. APPROACH

3.1 Dataset

The initial step towards finding the cyberbullying cases is to detect abusive language. Cyberbullying happens when a user repeatedly receives lots of negativity. Therefore, to detect this event, we need to monitor full history of online users. We collect our data from ask.fm. This is a semi-

[1] http://nobullying.com/the-complicated-web-of-teen-lives-2015-bullying-report/

[2] http://cyberbullying.org/summary-of-our-cyberbullying-research

anonymous social network, where anyone can post a question to any other user and may choose to do so anonymously. The anonymity option in ask.fm allows attackers the power to freely harass users by flooding their pages with profanity-laden questions and comments.

To create the data, we collect a large data of 3K users from ask.fm. We use the dataset proposed in [3] as training set and apply the best system presented in the same work to automatically label each row of the data. For making the positive examples (cyberbullying cases), for each user, we create a fixed-size sliding window and move it through the whole history of the user. For each window sample, we calculate the ratio of negativity. If it is greater than a threshold, we consider the window as a potential cyberbullying case and check if we can expand it keeping the inside negativity rate the same. We do it for two reasons. First, larger windows are more likely to be the instances of cyberbullying, since they contain more negativity in a specific period of time. Second, having more context would provide more information for the model. We then look at the resulting windows manually to make sure that they represent the cyberbullying events. We empirically fixed the minimum window size and threshold to 20 and 45% respectively. For the negative examples (non-cyberbullying cases), we apply the same method inversely. In this case, we look for the windows that have the negativity ratio less than chosen threshold. We also make sure that we cover the cases which have high negativity, but are not cyberbullying (e.g. when two users fight with each other in another user's timeline) as the samples for negative class.

Table 1 shows the distribution of the data. Since cyberbullying is very rare, we keep the ratio of positive to negative examples 1:10 to be closer to the real scenarios. Finally, we divide all training and test examples to 10 different chunks to specialize the corpus for early text classification task. Each chunk contains 10% of the whole information for ta single user window.

Table 1: Data distribution

Class	Training	Test	Total
positive	19	8	27
negative	190	80	270
Total	209	88	297

3.2 Methodology

We use the following features to extract the information from the texts in each iteration:

Lexical: Words are powerful tools to convey a feeling, describe or express an idea. With this notion, we use word n-grams, char n-grams, k-skip n-grams (to capture long-distance context) as features. For word n-gram features, we build a vocabulary that only considers top 10K features ordered by term frequency across the corpus. We weigh each term with its term frequency-inverse document frequency (TF-IDF).

Word Embeddings: The idea behind this approach is to use a vector space model to improve lexical semantic modeling**.** We use pre-trained Google News model including embeddings for about 3 million words. We create our feature vector by averaging the word embeddings of all the words in each post.

3.3 Experimental Setups

In an early text classification task, at every time t, we have access to all 10 chunks of the training data, but only t chunks of the test set. In all the experiments, we trained a LinearSVC using full-length documents in the training dataset. In the testing phase, the classifier uses all the available information in each of the 10 chunks iteratively. Specifically, we generate document representations starting with the first chunk, and then incrementally adding one more chunk of data at a time. The models will then make predictions incrementally as well.

3.4 Evaluation

For the evaluation of our early predictive model, we report the performance of the different methods when using increasing amounts of textual evidence (chunk by chunk evaluation). This evaluation allows to quantify prediction performance when using partial information in documents, and it is a strategy that has been used to evaluate early classification [6], [8]. We use the F1 measure as the evaluation metrics due to the fact that our dataset is highly imbalanced towards the negative class.

4. RESULTS AND CONTRIBUTIONS

4.1 Results

Table 2 shows the preliminary classification results of positive class for some features. The results for the other features like skip-gram are not included in the table due

Table 2: F1-score for chunk by chunk evaluation for positive class

Feature	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10
Unigram	0.46	0.54	0.66	**0.76**	0.61	0.71	0.71	0.61	0.61	0.67
Trigram	0.00	0.20	0.22	0.22	0.40	0.54	0.54	**0.61**	0.50	0.33
Character 3-gram	0.40	0.22	0.40	0.40	**0.54**	0.36	0.40	0.54	0.54	0.54
Word2vec	0.43	**0.59**	0.47	0.53	0.53	0.57	0.50	0.50	0.50	0.36
Unigram + Word2vec	0.67	0.61	0.67	0.67	**0.71**	0.71	0.71	0.71	0.61	0.66

to the very low performance; however, based on previous research they have reasonable performance in case of abusive language detection [3]. It shows that even though these two tasks look very similar, but in practice, they are completely different. Based on results, the best F1 measure obtained from unigram features using first 4 chunks of the test data. It shows that the vocabulary used in cyberbullying episodes are pretty different from non-cyberbullying cases especially in the early scopes of the event occurrence. Another interesting observation is that word embeddings features obtain the best performance in earlier chunks. Also, the results show that by combining embeddings and unigram features, the system shows a good performance even in the first chunk considering the fact that the corpus we are using is highly imbalanced towards the negative class. It seems that adding more information to the test data decreases the performance of the system in all cases. We can explain it in two different ways: 1) Adding more evidence increases the complexity of the system and make the decision harder for the classifier 2) Early text categorization scenario can be successfully mapped to cyberbullying detection task, since it seems that in most cases, the system can detect the changes in early chunks.

4.2 Contributions

The main contributions of this work are:

- Building a new system to automatically find the potential victims of Cyberbullying in social media data.
- Presenting the first corpus suited for early cyberbullying prediction in which we have the history of each user.
- Designing and developing a system to detect cyberbullying events as early as possible. In this system, we use lexical, semantic, and stylistic features to capture explicit abusive contents.

5. REFERENCES

[1] Wulczyn, E., Thain, N. and Dixon, L. 2016. Ex Machina: Personal Attacks Seen at Scale. *CoRR*.

[2] Dinakar, K., Jones, B., Havasi, C., Lieberman, H. and Picard, R. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*.

[3] Samghabadi, N. S., Maharjan, S., Sprague, A., Diaz-Sprague, R and Solorio, T. 2017. Detecting Nastiness in Social Medi. In *Proceedings of the First Workshop on Abusive Language Online*.

[4] Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W. and Hoste, V. 2015. Detection and Fine-Grained Classification of Cyberbullying Events. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*.

[5] Badjatiya, P., Gupta, S., Gupta, M. and Varma, V. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*.

[6] Losada, D. E., Crestani, F. and Parapar, Javier. 2017. eRISK 2017: CLEF Lab on Early Risk Prediction on the Internet: Experimental foundations. In *Proceedings of International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer.

[7] Dulac-Arnold, G., Denoyer, L. and Gallinari, P. 2011. Text classification: a sequential reading approach. In *European Conference on Information Retrieval*. Springer.

[8] Escalante, H. J., Montes-y-Gomez, M., Villasenor-Pineda, L. and Errecalde, M. L. 2016. Early text classification: A Naive solution. In *Proceedings of NACCL-HLT*.