BullStop: A Mobile App for Cyberbullying Prevention

Semiu Salawu
Aston University
Birmingham, B4 7ET
United Kingdom
salawusd@aston.ac.uk

Yulan He
University of Warwick
Coventry, CV4 7AL
United Kingdom
yulan.he@warwick.ac.uk

Jo Lumsden
Aston University
Birmingham, B4 7ET
United Kingdom
lumsdenj@aston.ac.uk

Abstract

Social media has become the new playground for bullies. Young people are now regularly exposed to a wide range of abuse online. In response to the increasing prevalence of cyberbullying, online social networks have increased efforts to clamp down on online abuse but unfortunately, the nature, complexity and sheer volume of cyberbullying means that many cyberbullying incidents go undetected. BullStop is a mobile app for detecting and preventing cyberbullying and online abuse on social media platforms. It uses deep learning models to identify instances of cyberbullying and can automatically initiate actions such as deleting offensive messages and blocking bullies on behalf of the user. Our system not only achieves impressive prediction results but also demonstrates excellent potential for use in real-world scenarios and is freely available on the Google Play Store.

1 Introduction

Cyberbullying is defined as 'wilful and repeated harm inflicted through the use of computers, cell phones, and other electronic devices (Hinduja and Patchin, 2006, pg. 152). It is estimated that as many as 59% of US teenagers would have experienced some form of cyberbullying by the time they become young adults (Pew Research Center, 2018). Nowhere is cyberbullying more prevalent than on social media, where, as much as 69% of reported incidents of cyberbullying took place (Ofcom Research, 2019). In response to the increased proliferation of online abuse, social media platforms like Twitter, Facebook, and Instagram have, in recent years, introduced policies and features to combat and mitigant cyberbullying and its effects. These include preventing the creation of multiple accounts using similar details and suspending abusive users. Human moderators are also employed by online social networks to review thousands of posts daily. Unfortunately, such is the prevalence of cyberbullying, and online harassment that, despite these efforts, it remains a significant online risk for many young people. Furthermore, as cyberbullying is highly subjective, what is deemed offensive differs amongst people, human moderators can only apply generalised rules in making a judgement.

Existing mobile tools to combat cyberbullying mostly either use wordlists or lack the flexibility to cope with the evolving nature of social media. Lempa *et al.* (2015) developed a "sentence checker" mobile app that allows users to check for offensive content in messages before sending. However, as the app is incapable of sending messages or integrating with other messaging applications, users will have to type or copy the message to other messaging applications to send the message. Vishwamitra *et al.* (2017) developed MCDefender, a mobile app that serves as a cyberbullying detection companion to the Facebook app. The app detects words typed in the Facebook app and analyses each word to determine if the user is engaging in bullying activities. The app's tight integration to the Facebook app, however, exposes some limitations. For example, the app cannot detect messages sent if Facebook is accessed via a web browser or if an unofficial Facebook messaging app is used. A better implementation would be to integrate via the social media platform's API, which is the approach adopted by our system.

Proceedings of the 27th International Conference on Computational Linguistics, pages 70–74
Barcelona, Spain (Online), December 12, 2020.

There are two critical challenges in developing viable tools to combat cyberbullying on social media. These are; the evolution of the mainly colloquial language used in social media and the need for the tool to react in a timely fashion to cope with real-time communication. To overcome these challenges, our system uses a microservices-based architecture that allows the introduction of classifiers in a "plug and play" manner and leverages containerisation and cloud-based technologies to exponentially scale to meet the demands of a modern social media platform. In this paper, we present BullStop; a mobile app developed to help young people combat cyberbullying and online abuse. While Twitter is currently supported, it is designed to work with multiple social media platforms.

2 Architecture

The app utilises a novel approach comprised of two key strategies. Firstly, the detection of cyberbullying and online abuse is modelled on how email applications treat spam. It uses a generalised deep learning model to identify the majority of cyberbullying instances and then improves the base model via online training utilising the user's input as ground truth. In this way, the app becomes a personalised cyberbullying detector for each user with a deep understanding of how the user communicates. Secondly, the use of a loosely coupled microservices architecture disassociates the deep learning model from the rest of the system, thus allowing models to be "plug and play". Thus, any model can be easily incorporated into the system; all that is required is a model capable of multi-label classification that accepts textual data and provides its output in the required JSON format. This dramatically improves the system's flexibility, allowing newer and more advanced classifiers to be introduced into the system as required.

The system architecture is illustrated in Figure 1. Messages and contact information from the online social network are introduced into the system via the Synchroniser or Webhook. The Synchroniser is a mobile component that performs regular data harmonisation with online social networks. It connects to the social media platform via an API and extracts new messages, posts and contacts from the user's account. The Webhook provides a similar function, but unlike the Synchroniser which retrieves information on a schedule via a pull mechanism, the Webhook receives new data via a push from the online social network. The Webhook is an optional component only used when the social media platform provides a mechanism to send such notifications. It is an event-triggered feature that allows the social network to send new data to an external interface when monitored events occur, for example, a new friend request or message received. Data received from the online social network is placed on the Message Queue where it is processed by the Abuse Detection Module (ADM).

The ADM is comprised of one or more machine learning models that predict the labels for the messages retrieved from the Message Queue. The models have been trained to predict the labels, *Cyberbullying*, *Insult*, *Profanity*, *Sarcasm*, *Threat*, *Exclusion*, *Spa*, *Porn* for each message. BERT, DistilBERT, RoBERTA and XLNET are some of the models available in the ADM, but other models can be easily added. The labels predicted for a message can be corrected by the user, and these are stored and used to retrain the model. The output from the ADM determines actions taken by the Marshaller. The Marshaller communicates with the social networks via the API and initiates appropriate actions on behalf of the user.

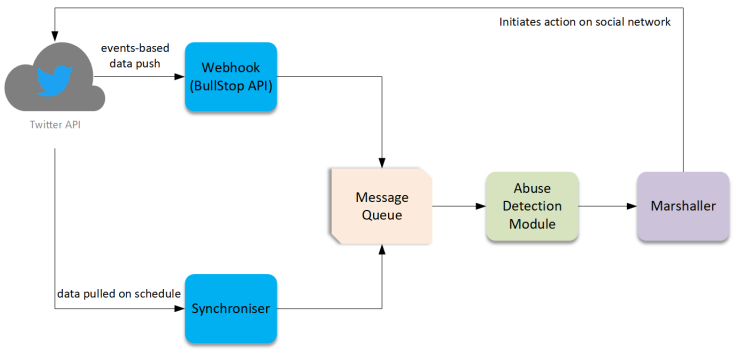

Figure 1: High Level Architecture.

3 Use Case

UI design is a critical component for mobile apps, more so for the young target audience of our system. Therefore, from the onset, the app was designed to appeal to this audience, and many of the design decisions taken were in furtherance of this goal. Feedback from a cross-section of stakeholders, including young people, parents, educators, law enforcement and child mental health professionals was actively sought and influenced the user interface of the application. For the adult stakeholders, we conducted six focus group sessions to gain insight into their views on cyberbullying and its prevention. Individual interviews were then conducted with twenty-five adolescents representing the target audience which was followed by a participatory design phase where six young people worked with us to collaboratively design the mobile application. The functionalities and UI of the mobile app is based on the outcome of these sessions.

After installing the app, users are prompted to create a profile which is used to store their settings and preferences. Industry-standard cryptographic and encryption technologies are used to create and manage accounts securely. Users can then associate their profiles with social network accounts (see Figure 2) and configure personal settings (see Figure 3). The app will continuously monitor the associated social media accounts for new messages/posts to analyse. Relevant labels are assigned to the message based on the model's prediction (see Figure 4). Each label is associated with a score, and the cumulative score for the message is computed and compared against the user's preferences which determine if the message should be deleted and the sender blocked. Users can review received messages (including deleted ones) and update the assigned labels. Any such message is used by the app as data for online training. We acknowledged that providing users with the ability to read deleted offensive messages in order to re-classify them may seem counterproductive. The stakeholders were however in favour of this feature, and as a mitigating feature, the app permanently removes the most offensive messages automatically after a configurable period.

Figure 2: Authorisation Screen

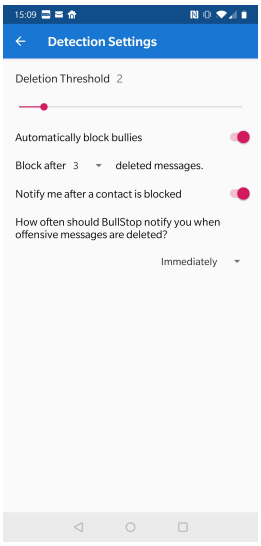

Figure 3: Settings Screen

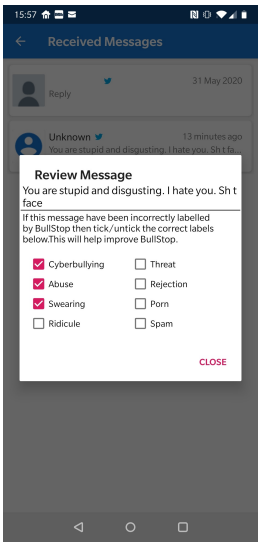

Figure 4: Message Screen

4 Experiments

Dataset. The models used in the ADM were trained on 62,587 tweets extracted from Twitter using query terms designed to return offensive tweets. Query terms used included the 15 most frequently used profane words on Twitter (Wang *et al.*, 2014) as well as hashtags such as #sarcasm, #stayinyourlane, #maga, #blacklivesmatter to capture different forms of online abuse and bullying. Our collection strategy was aimed at creating a dataset with high concentration of bullying content. This approach differs from standard practice, which seeks to emulate natural distribution and typically results in datasets with

minimal cyberbullying content thus requiring oversampling techniques to improve cyberbullying distribution within the dataset. We used a pool of 17 annotators to label the dataset, and each annotator was provided with 6,000 – 10,000 tweets. Each tweet was assigned to 3 different annotators and majority agreement required for each label. Krippendorff's Alpha (rater agreement) was calculated to be 0.67. Standard preprocessing steps including removal of punctuation, symbols, non-ASCII characters, user mentions, URL and lower casing were performed. The number of tweets associated with each class is as shown in Table 1.

Label	Profanity	Porn	Insult	Spam	Bullying	Sarcasm	Threat	Exclusion	None
Count	51,014	16,690	15,201	14,827	3,254	117	79	10	10,768

Table 1: Total number of tweets each label was assigned to.

Evaluation. We trained a set of traditional classifiers (Multinomial Naive Bayes, Linear SVC, Logistic Regression) and deep learning-based models (BERT, RoBERTa, XLNet, DistilBERT) on the dataset. BERT (Bidirectional Encoder Representations from Transformers) is a language representation model used to pre-train deep bi-directional representations from unlabeled text (Devlin *et al.*, 2018). RoBERTa (Robustly Optimized BERT Pretraining Approach) is an optimised BERT-based model (Liu *et al.*, 2019) trained using ten times more data than BERT to improve performance. DistilBERT (Distilled BERT) is a compacted BERT-based model (Sanh *et al.*, 2019) that requires fewer computing resources and training time than BERT but preserves most of BERT performance gains with little performance degradation. XLNet (Yang *et al.*, 2019) is an autoregressive BERT-like model designed to overcome some of the limitations of BERT. We utilised both pre-trained versions of the deep learning models as well as fine-tuning the models on our dataset. The deep learning models outperformed the baseline classifiers with Multinomial Naive Bayes emerging as the worst classifier across the experiments (see Table 2). An interesting discovery was also that the pre-trained models performed better than the fine-tuned models, which is in agreement with the findings of Radiya-Dixit and Wang (2020). A probable reason for this is because our dataset is much smaller than the corpus used for pre-training and as such, the models were unable to learn additional context during fine-tuning. The overall best classifier was RoBERTa using pre-trained weights. Consequently, this was the model used in our system.

Model	**Macro ROC-AUC(↑)**	**Accuracy (↑)**	**Hamming Loss (↓)**	**Macro F_1(↑)**	**Micro F_1(↑)**
Multinomial Naive Bayes	0.8030	0.4568	0.1014	0.2618	0.7200
Linear SVC	0.8353	0.5702	0.0866	0.3811	0.7674
Logistic Regression	0.8354	0.5743	0.0836	0.3587	0.7725
BERT (pre-trained)	0.9657	0.5817	0.0736	**0.6318**	0.7998
DistilBERT (pre-trained)	0.9675	0.5802	0.0764	0.5202	0.7855
RoBERTa (pre-trained)	**0.9695**	0.5785	**0.0722**	0.5437	**0.8081**
XLNet(pre-trained)	0.9679	0.5806	0.0738	0.5441	0.8029
BERT (fine-trained)	0.9651	0.5822	0.0725	0.5300	0.8022
DistilBERT (fine-trained)	0.9633	**0.5834**	0.0753	0.5040	0.7872
RoBERTa (fine-trained)	0.9670	0.5794	0.0724	0.5329	0.8044
XLNet(fine-trained)	0.9654	0.5819	0.0741	0.5308	0.8037

Table 2: Results of classification. (↑: higher the better; ↓: lower the better)

5 Conclusion

We have presented our system for fine-grained detection and prevention of cyberbullying and online abuse on social media. Besides its flexible architecture which allows the use of any classifier, it incorporates online training using ground truth provided by the user for retraining. The app is available

on the Google Play Store (https://play.google.com/store/apps/details?id=mobile.bullstop.io), and future work planned includes the use of multiple classifiers and the expansion of the original dataset to include more tweets and content from other online social networks.

References

Evani, Radiya-Dixit and Xin Wang. 2020. How fine can fine-tuning be? Learning efficient language models. *Computing Research Repository*, arXiv:2004.14129. Version 1.

Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Computing Research Repository*, arXiv:1810.04805. Version 2.

Justin W. Patchin and Sameer Hinduja. 2006. Bullies Move Beyond the Schoolyard. *Youth Violence and Juvenile Justice*, 4(2):148–169

Nishant Vishwamitra, Xiang Zhang, Jonathan Tong, Hongxin Hu, Feng Luo, Robin Kowalski, and Joseph Mazer. 2017. MCDefender: Toward effective cyberbullying defense in mobile online social networks. In Proceedings of the 3rd ACM International Workshop on Security and Privacy Analytics 2017, pages 37-42, Arizona, USA.

Ofcom Research. 2019. Online Nation. [online] ofcom.org.uk. Available at: `https://www.ofcom.org.uk/__data/assets/pdf_file/0025/149146/online-nation-report.pdf`

Pawel Lempa, Michal Ptaszynski and Fumito Masui. 2015. Cyberbullying Blocker Application for Android. In Proceedings of the 7th Language & Technology Conference 2015, pages 408 - 412, Poznan, Poland.

Pew Research Center. 2018. A Majority of Teens Have Experienced Some Form of Cyberbullying. [online] pewresearch.org. Available at: `https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2018/09/PI_2018.09.27_teens-and-cyberbullying_FINAL.pdf`

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *Computing Research Repository*, arXiv:1910.01108. Version 4.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. 2014. Cursing in English on Twitter. In Proceedings of the 17th ACM conference on Computer supported cooperative work social computing, pages 415-425, Baltimore, USA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Computing Research Repository*, arXiv:1907.11692 Version 1.

Yu D. Weider, Maithili Gole, Nishanth Prabhuswamy, Sowmya Prakash, and Vidya Gowdru Shankaramurthy. 2016. An approach to design and analyze the framework for preventing cyberbullying. In Proceedings of IEEE International Conference on Services Computing, 2016, pages 864-867, San Francisco, USA.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Computing Research Repository*, arXiv:1906.08237 Version 2.