Automated Content Warnings for Sensitive Posts

Manuka Stratta
Stanford University
Stanford, CA 94305, USA
mstratta@stanford.edu

Cooper deNicola
Stanford University
Stanford, CA 94305, USA
cdenicol@stanford.edu

Julia Park
Stanford University
Stanford, CA 94305, USA
julpark@stanford.edu

CHI '20 Extended Abstracts, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6819-3/20/04.
http://dx.doi.org/10.1145/3334480.3383029

Abstract
The inclusion of content warnings for sensitive topics on webpages contributes to creating a psychologically safe internet for all users. Yet the pervasiveness of these warnings is limited by their reliance on content creators and hosts. Rather than placing the sole responsibility of content moderation on content creators and hosts, our system shows a strong proof-of-concept for automatically generating warnings on the user's side by utilizing keyword identification, sentiment analysis, and online intervention user interface principles. We designed our system as a Chrome extension and evaluated it by testing its accuracy on a dataset of websites with and without sensitive content, and performing a user-interaction lab study. With promising future areas of research such as the ability to personalize thresholds and customize content warnings for specific user needs, this research is a step towards a psychologically safer internet.

Author Keywords
Content Warning; Trigger Warning; Topic Identification; Website Classifier; Online Intervention; Online Safety

CCS Concepts
•Human-centered computing → **Interactive systems and tools;** *Accessibility systems and tools;* **Web-based interaction; •Information systems** → *Browsers;*

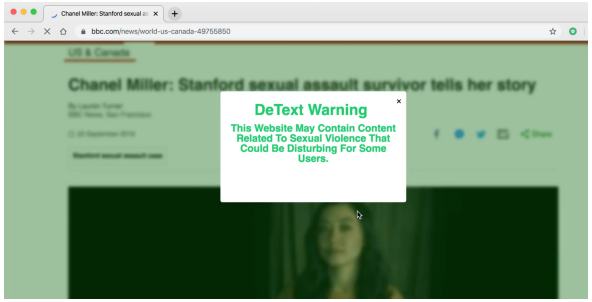

Figure 1: Screenshot of DeText's automatically generated content warning alert running on a website which displays sensitive content relating to sexual violence. The figure shows a blurred out background behind the alert, concealing possible graphic text and imagery.

INTRODUCTION

Online content can include material that may be disturbing to some people [12]. For users with past traumatic experiences or medical conditions such as PTSD, unfiltered content can cause flashbacks and additional trauma [2]. As such, the user is at risk if warnings are not included. Content warnings are important as they provide viewers the opportunity to prepare themselves, decreasing the chances of additional trauma [12].

However, the presence of content warnings depends on the decisions and perspectives of content creators and hosts, which leaves a large portion of online information without warnings [8, 18]. Some content creators have tried to introduce the practice of including content warnings on potentially distressing content that they publish, but this solution is inconsistent due to a lack of established standards for warnings [18]. Content platforms have also attempted to generate content warnings by utilizing third-party moderators who sift through graphic content [15]. However, these moderators can develop PTSD themselves due to exposure to a high volume of distressing content [15].

Given these challenges, we ask: How can we automatically generate online content warnings for distressing content?

Rather than placing the sole responsibility of content moderation on content creators and hosts, this paper shows the potential for automatically generating content warnings from the user's side and then alerting the user of sensitive content through a pop-up notification. Our approach rejects the current paradigm that only content creators or distributors can generate content warnings. Instead, our solution transfers the responsibility of flagging content from content owners to a client-side, automated content warning system. This creates more efficient, wide-reaching content warnings that do not require creators' input and will not overburden a handful of people with the task of tagging content.

To test this idea's feasibility, we designed and created the browser extension, DeText, which automatically generates content warnings and identifies sensitive text relating to sexual violence by using keyword recognition and sentiment analysis. When a webpage loads, the system scans the text and determines whether the user should be warned about distressing content. If so, Detext presents an alert and places a Gaussian blur over the background to conceal the sensitive text (Figure 1). To evaluate DeText's performance, we quantified the extension's accuracy and performed user evaluations.

This work demonstrates that alternative modes of online moderation are possible, namely that moderation can be automatically performed on the user side, shifting responsibility away from platforms and content creators. The promising possibility of automated user-side moderation expands the tools we have at our disposal to ensure that online browsing is safe for everyone.

MOTIVATION AND RELATED WORKS

Content warnings are notices that precede sensitive content to warn readers about information in the upcoming

text [2]. They offer readers a chance to engage with the material in a manner that supports their own personal well being [8] and reduces the risk of shock or trauma [12]. Research shows that knowledge of upcoming stressors reduces stress response and lets people feel in control [6]. Actively coping with trauma through prepared confrontation can also reduce distress over time [14]. Therefore, content warnings are psychologically useful in both the short and the long-term; they let people feel safe before recollecting a traumatic experience and aid in the overall recovery.

Despite the psychological utility of content warnings, in practice content creators rarely provide these warnings due to lack of awareness [8]. In response to the lack of warnings from content creators, some content platforms have attempted to self-regulate. For example, Facebook now hides graphic imagery with a gray screen overlay that users can choose to bypass. However, these systems rely on third-party contract workers to evaluate violent content, who then develop PTSD themselves [15]; furthermore, large portions of the internet are still left unmoderated.

Meanwhile, natural language processing (NLP) technologies like the Python NLTK toolkit can analyze language structure, extract keywords, and discern embedded emotions using sentiment analysis [4, 13]. Sentiment analysis, which determines whether a phrase's sentiment is positive, neutral, or negative [19], has already proven effective in detecting social media disclosures of sexual harassment [5, 7]. Keyword recognition is also often used to classify a text's topic [1, 9].

Furthermore, extensive research has also been conducted on user intervention design principles. Studies show that users are irritated by intrusive interventions [10] and become immune to overused alerts [3]. Thus, the current literature can inform the development of a user intervention that deals with sensitive topics, where alerts should be neither intrusive nor desensitizing.

In light of these existing technologies, our project seeks to automate content warning systems rather than place the burden to include content warnings on content creators and hosts. To design this new system, we combine psychology research on trauma and content warnings with existing research on natural language processing, text moderation technology and user intervention design.

DETEXT: SYSTEM DESIGN AND DESCRIPTION

To test the feasibility of automated content warnings, we created DeText, a Chrome extension. Since DeText is a proof-of-concept, we limited our scope to sensitive content on sexual violence. DeText runs in the background and scans the text of every loaded page for sensitive content. If such content is found, DeText displays an alert and blurs out the background (see Figure 1). The user may then choose to proceed to the page or return to the previous page.

DeText must balance effectiveness and non-intrusiveness. Effectiveness means that DeText's warning should be clearly visible. Non-intrusiveness means that the user should not perceive DeText as a "spam" extension that displays many flashy, frustrating pop-ups that present obstacles to surfing the web [3]. Therefore, in designing DeText, we intended to balance these two concepts to create the most effective and pleasant user experience.

The DeText system consists of the Chrome extension and a web server. The system's first step is to extract the HTML source code from the webpage and send this raw data to the Python server. The server then extracts the visible text from the source code using <p> and tags. The text is passed to the algorithm for analysis.

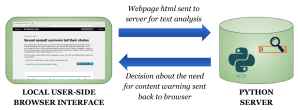

Figure 2: System schematic. The local user-side browser interface extracts HTML from a website and then sends this data to the Python server. The Python server analyzes the HTML using keyword search and the NLTK natural language processing library. This analysis determines whether or not the webpage should have a content warning. If it should, this decision is sent back to the local user-side extension, which alerts the user with a content warning alert.

The first step in the algorithm is a keyword search, which utilizes an implicit and explicit keyword list. All words on the explicit list are by definition connected to sexual violence (e.g. “rape”, “sexual assault”, etc.). In contrast, the implicit keywords may be used in other contexts as well (“pain”, “force”, etc.). The typical flagged page will likely include multiple occurrences of both explicit and implicit keywords. To form our explicit word list, we manually selected words explicitly related to sexual violence from existing lists of lewd and obscene words [11, 17]. To form our implicit word list, we used a training set of 12 articles that were already tagged with a content warning for sexual violence by its author. For each article, we selected the words most relevant to descriptions of sexual violence and categorized them as implicit and explicit. We compiled our findings and asked a dozen third-party people for feedback on which words should not be part of these keyword lists. We used the participants’ feedback to revise the lists and eventually settled on a final list containing 200+ implicit and explicit words.

For each paragraph, the system searches for words or phrases contained in the implicit and explicit data set. If the paragraph contains at least 1 phrase from the explicit list and at least 2 phrases from the implicit list, the paragraph is flagged as potentially containing sensitive content.

For each flagged paragraph, the system then performs sentiment analysis to determine the tone of the paragraph containing the flagged keywords. NLTK’s sentiment analysis module returns a “polarity” for the original text of each flagged paragraph [4]. This polarity ranges from -1 (negative sentiment) to 1 (positive sentiment); the magnitude indicates how polarized the sentiment is [19]. We consider sentiment to be “polarized” if its magnitude is greater than 0.05, and “neutral” if otherwise. If the sentiment is neutral, the text is likely factual (such as a Wikipedia article) and won’t contain graphic imagery [7]. Such articles are less likely to be triggering and therefore should not be flagged [7]. Thus the system only marks polarized paragraphs as containing sensitive content. The system then creates a data structure for each paragraph marked as containing sensitive content, storing the number of implicit phrases found, the number of explicit phrases found, and the paragraph’s sentiment polarity value.

After scanning each paragraph as described above, the system uses the data structures to compute the total count of explicit and implicit words found across all flagged paragraphs. If the total count of explicit phrases is greater than 2, and the total count of implicit phrases is greater than 3, and at least one paragraph has polarized sentiment, then the system decides to present a content warning. We arrived at these criteria by trial and error, tweaking the thresholds and testing these criteria against our training set of webpages, optimizing for the lowest false negative rates and highest true positive rates. The server’s decision is then communicated back to the extension.

WEBSITE COMPARISON EVALUATION

Method

Automatically generated content warnings, if shown to operate accurately, are inherently more consistent and efficient than creator-generated content warnings. Therefore, if automatically generated content warnings are as accurate as creator-generated warnings, then they offer an improvement over the previous methodology.

For our first evaluation, we tested DeText’s accuracy against a testing set of webpages whose true classification is known. In this evaluation, our aim was to evaluate the extension’s accuracy as objectively as possible. Here, true classification is determined by the existence of a content warning.

	Had CW	*Had no CW*	*Total*
Flagged	21	1	22
Not flagged	4	24	28
Total	25	25	50

Table 1: Results from website comparison evaluation. Shows confusion matrix of positive and negative actual labels crossed with the predicted labels generated when DeText analyzes the sites in the testing set. These results show that DeText has an extremely high overall accuracy rate for both positive and negative testing cases.

	Had CW	*Had no CW*	*Total*
Flagged	76	4	80
Not flagged	7	144	151
Total	83	148	231

Table 2: Results from user usage evaluation. The false positive and false negative values were directly recorded during the study, while the true positive and true negative values were calculated as the instances where DeText did or did not flag the website but the user did not complain about its decision.

For our testing set, we identified 37 webpages that were known to be about sexual violence. 12 webpages were set aside to train the algorithm. The other 25 webpages were included in the testing set after we stripped them of the content warning sentence (as it would contain easily identifiable keywords). We then arbitrarily gathered 25 additional webpages that did not relate to sexual violence. Thus, our final testing set had 25 webpages that contained sensitive sexual violence content and 25 webpages that did not.

We ran DeText against these webpages and recorded the results. Since DeText tackles a classification problem, we used precision, recall, and F1-score to evaluate our model's performance as these measures take into account false positive/negative rates better than simple accuracy [16].

Results

The F1-score calculated for DeText is 0.8940, demonstrating our classification algorithm's high accuracy. Its precision is 0.9545, and its recall is 0.8400. These results show that our extension does a near perfect job on websites that are not about sexual violence. For DeText, false negatives (not flagging a webpage that should have been flagged) matter more than false positives (flagging a webpage that does not need to be flagged), and our extension shows remarkable accuracy with websites that contain sexual violence content. These results show that content warnings can be automatically generated with high accuracy.

USER USAGE EVALUATION

Method

In our second evaluation, we measured DeText's accuracy, UI design, and usability by asking willing college student participants to use DeText as an extension for 15 minutes. They were then prompted for feedback on their experience through a poll. We collected data from a total of 10 participants in the age range of 18-22, with a mean age of 19.1. 50% of these participants were male and 50% were female.

After giving each participant a brief overview of what a content warning is and a tutorial of DeText, we asked them to test the extension by browsing the web normally for 5 minutes and then specifically looking for content relating to sexual violence in the last 10 minutes. We asked them to report each time they felt the extension 1) did not flag something that should have been flagged and 2) flagged something that should not have been flagged. We manually recorded the false positive and false negative rates, the total number of webpages visited, and how many pages were flagged by DeText.

After the 15-minute period was over, we asked the participants to subjectively evaluate the accuracy of DeText by responding to 4 questions about their perceived accuracy, trust, and user experience.

Results

On average, 23.1 websites were visited by each participant, for a total of 231 websites visited. We aggregated the number of flagged pages, pages visited, false positives, and false negatives to calculate the precision, recall, and F1-score of the extension.

The F1-score calculated for DeText is 0.9325. The extension's precision is 0.9500, and its recall is 0.9157. These results confirm the data from our previous evaluation method: our extension performs extremely well. The F1-score increased with more articles tested (50 vs. 231). This finding confirms that DeText's flagging algorithm works as real users expect it to work.

Q1 (*"How often did you feel that DeText correctly presented or did not present content warnings?"*) tested the participants' perceived accuracy of DeText; common responses

User Survey Questions:

Question 1: How often did you feel that DeText correctly presented or did not present content warnings?

Question 2: How much do you trust DeText's accuracy?

Question 3: Please rate the design and usability of DeText.

Question 4: Please provide any general comments and feedback on the extension for further investigation and improvement.

	Mean	Median
Q2	5.75	6.0
Q3	5.3	5.75

Table 3: The mean and median scores for quantitative user survey questions 2 and 3, on a scale of 1-7. Questions 1 and 4 are addressed in the *Results* section as they follow a free response format.

included phrases such as "pretty well", "almost always", "in the 90-95% range", "Oversafe rather than undersafe [*sic*]". The mean and median scores for Q2 (*"How much do you trust DeText's accuracy?"*) were 5.75 and 6 respectively, on a scale of 1-7 where 1 means "Not at all" and 7 "Completely". This positive rating in general confirmed the participants' qualitative responses to Q1.

For Q3 (*"Please rate the design and usability of DeText."*), the average and mean scores were 5.3 and 5.75 respectively, on the same scale. Although slightly lower than those for Q2, the responses reflected an overall satisfaction in the extension's design. Responses to Q4 (*"Please provide any general comments and feedback on the extension for further investigation and improvement."*) included suggestions such as "Add customization for specific keywords, topics, and thresholds" or "Block out individual page sections, not entire articles", which can direct future research.

Thus, our two evaluation methods have shown that content warnings can indeed be automatically generated with an accuracy that meets human expectations, validating our original claim and thesis. The positive responses on the user poll questions on usability and design as well as the general feedback questions also confirm that the extension in general provides a positive experience to the user. Overall, our research shows that user-side automated content warnings can be accurately generated without negatively impacting user experience.

DISCUSSION

DeText proves that automatically generating content warnings from the user-side is feasible. The discussion will acknowledge the limitations of our evaluation, look at future applications of DeText's analysis model, and address potential future research vectors.

Study Limitations

One confounding variable in our first evaluation is that articles written by individuals who willingly add content warnings to their writing may use different word patterns than creators who do not add content warnings. We attempted to overcome this limitation by evaluating our extension against general human expectations in our second evaluation.

A drawback from our second evaluation, however, is the limited testing time. There is evidence that online intervention methods become frustrating over time, causing users to become impatient with the extension and uninstall [10]. While our users approved of DeText's non-intrusiveness, their opinion may change when they use the extension for a longer period of time. Further evaluation over a longer period of time would help us better evaluate DeText's design.

Future Work

As DeText focused on acting as a proof of concept, there are many aspects that can be improved upon in future research. For example, DeText can be improved to provide warnings for a wider range of topics such as suicide and self-harm, or analyze a wider range of media including emails, pictures, and videos. Allowing users to customize their personal keyword list would give users more control over their online experience.

CONCLUSION

This research demonstrates that it is possible to automatically detect potentially upsetting content online. Such systems can supplement or replace the current inconsistent and labor-intensive methods for very little, if any, trade-off in accuracy. The findings of this research can be used to create systems where the user's psychological safety on the internet is in their own hands.

REFERENCES

[1] Saleem Abuleil. 2007. Using nlp techniques for tagging events in arabic text. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, Vol. 2. IEEE, 440–443.

[2] Payton J. Jones Bellet, Benjamin W. and Richard J. McNally. 2018. "Trigger warning: Empirical evidence ahead". *Journal of behavior therapy and experimental psychiatry* 61 (2018), 134–141. DOI: http://dx.doi.org/10.1016/j.jbtep.2018.07.002

[3] Jan Panero Benway. 1998. Banner Blindness: The Irony of Attention Grabbing on the World Wide Web. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 42, 5 (1998), 463–467. DOI: http://dx.doi.org/10.1177/154193129804200504

[4] Steven Bird, Ewan Klein, Edward Loper, and Jason Baldridge. 2008. Multidisciplinary Instruction with the Natural Language Toolkit. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*. Association for Computational Linguistics, Columbus, Ohio, 62–70. https://www.aclweb.org/anthology/W08-0208

[5] Dasha Bogdanova, Paolo Rosso, and Thamar Solorio. 2012. On the impact of sentiment and emotion based features in detecting online sexual predators. In *Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis*. Association for Computational Linguistics, 110–118.

[6] Guy A. Boysen. 2017. "Evidence-based answers to questions about trigger warnings for clinically-based distress: A review for teachers". *Scholarship of Teaching and Learning in Psychology* 30.2 (2017), 163–177. DOI: http://dx.doi.org/10.1037/stl0000084

[7] Arijit Ghosh Chowdhury, Ramit Sawhney, Puneet Mathur, Debanjan Mahata, and Rajiv Ratn Shah. 2019. Speak up, Fight Back! Detection of Social Media Disclosures of Sexual Harassment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. 136–146.

[8] Science College of Literature and the Arts. 2017. An Introduction to Content Warnings and Trigger Warnings. (2017). sites.lsa.umich.edu/inclusive-teaching/2017/12/12/an-introduction-to-content-warnings-and-trigger-warnings/.

[9] Marco Guerini, Carlo Strapparava, and Oliviero Stock. 2010. Evaluation Metrics for Persuasive NLP with Google AdWords.. In *LREC*.

[10] Geza Kovacs, Zhengxuan Wu, and Michael S. Bernstein. 2018. "Rotating Online Behavior Change Interventions Increases Effectiveness But Also Increases Attrition". *PACMHCI* 2 (2018), 95:1–95:25. DOI:http://dx.doi.org/10.1145/3274364

[11] LDNOOBW. List of Dirty Naughty Obscene and Otherwise Bad Words. (????). https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words/blob/master/en.

[12] Eleanor Amaranth Lockhart. 2016. Why trigger warnings are beneficial, perhaps even necessary. *First Amendment Studies* 50, 2 (2016), 59–69. DOI: http://dx.doi.org/10.1080/21689725.2016.1232623

[13] Nitin Madnani. 2007. Getting started on natural language processing with Python. *ACM Crossroads* 13, 4 (2007), 5.

[14] Willie Langeland Miranda Olff and Berthold PR Gersons. 2005. "The psychobiology of PTSD: coping with trauma". *Psychoneuroendocrinology* 30.10 (2005), 974–982. DOI:http://dx.doi.org/10.1016/j.psyneuen.2005.04.009

[15] Casey Newton. 2019. “The Trauma Floor: The Secret Lives of Facebook Moderators in America”. (25 February 2019). www.theverge.com/2019/2/25/18229714/cognizant-facebook-content-moderator-interviews-\trauma-working-conditions-arizona.

[16] Chris Nicholson. Accessed 2019. Evaluation Metrics for Machine Learning - Accuracy, Precision, Recall, and F1 Defined. (Accessed 2019). https://pathmind.com/wiki/accuracy-precision-recall-f1.

[17] Luis von Ahn’s Research Group. 2006. Offensive/Profane Word List. (2006). https://www.cs.cmu.edu/~biglou/resources/bad-words.txt.

[18] Katy Waldman. 2016. The Campus Debate Over Trigger Warnings Is at an Impasse. Science Can Help. (5 September 2016). www.slate.com/articles/double_x/cover_story/2016/09/what_science_can_tell_us_about_trigger_warnings.single.html.

[19] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.