Overview of the HASOC Subtrack at FIRE 2023: Identification of Tokens Contributing to Explicit Hate in English by Span Detection

Sarah Masud[1,*], Mohammad Aflah Khan[1], Md. Shad Akhtar[1] and Tanmoy Chakraborty[2]

[1]*Indraprastha Institute of Information Technology, Delhi*
[2]*Indian Institute of Technology, Delhi*

Abstract

As hate speech continues to proliferate on the web, it is becoming increasingly important to develop computational methods to mitigate it. Reactively, using black-box models to identify hateful content can perplex users as to why their posts were automatically flagged as hateful. On the other hand, proactive mitigation can be achieved by suggesting rephrasing before a post is made public. However, both mitigation techniques require information about which part of a post contains the hateful aspect, i.e., what spans within a text are responsible for conveying hate. Better detection of such spans can significantly reduce explicitly hateful content on the web. To further contribute to this research area, we organized *HateNorm* at HASOC-FIRE 2023, focusing on explicit span detection in English Tweets[1]. A total of 12 teams participated in the competition, with the highest macro-F1 observed at 0.58.

Keywords

Hate Span, Explicit Hate, English Tweet, HASOC'23

1. Introduction

Hate speech is a challenging social issue given its subjective nature: what is hateful changes with time, geography, and cultural context. United Nations defines hate speech[1] as "any form of communication that uses pejorative or discriminatory language with reference to a person or a group based on who they are." Further, hate speech has real-world implications; not only do real-world biases drive up online hate speech, but online hate speech can lead to an increase in hate crimes in the offline world. To reduce the burden on the volume and velocity of hateful content accessed by content moderators, analyzing, mitigating, and countering hateful content via computational methods is binding. While computer-aided can help perform the first level of mitigation, human involvement in subjective matters like hate speech is critical and compulsory for improving social systems in the real world. To aid the systems in better detection of hateful

[1]**Disclaimer:** The paper contains samples of hate speech, which are only included for contextual understanding. We do not support these views.

Forum for Information Retrieval Evaluation, December 15-18, 2023, India

*Corresponding author.
sarahm@iiitd.ac.in (S. Masud); aflah20082@iiitd.ac.in (M. A. Khan); shad.akhtar@iiitd.ac.in (Md. S. Akhtar); tanchak@iiitd.ac.in (T. Chakraborty)

[1]https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech

Table 1
A few examples of hateful posts of varying degrees from the dataset curated by Masud et al. [3] and their corresponding hateful spans marked in red.

Sample #	Text
1	Women ... Can't live with them ... Can't shoot them
2	kathy griffin is the ultimate liberal attack somebody and suffer consequences and 5 & she is the victim c**t needs to have some dick forced up her dy** ass barron trumps

content, one can look into developing systems that can capture and attend to the hateful spans [1] within a sentence. Span detection can help develop a sense of rationale, act as a tool for post hoc analysis, and improve the retrival of critical facts in claim verification [2].

Shared Task Objective. A hate span is a set of continuous tokens that, in tandem, communicate the explicit hatefulness in a sentence. Table 1 provides some examples of harmful social media posts marked for hateful fragments. For instance, in the first sentence of Table 1, *"Women ... Can't live with them ... {Can't shoot them}"*, the portion highlighted in red will be considered as a hateful span. Formally, *given a hate sample, tokenized as* $t = \langle w_1, w_2, \ldots, w_n \rangle$, *the hate span identification task looks for a sequence of hateful tokens,* $\langle w_i, .., w_{i+l} \rangle$ [3].

> **Problem Definition:** *Given a hateful text, identify those specific fragments within the sentence that are hateful. This is a sequence tagging task where the aim is to label each word as either belonging to the hate span or not.*

Share Task Details. Through the *HateNorm* shared task part of HASOC-FIRE 2023[2], we aimed at engaging the broader research community in understanding span detection techniques and contributing towards the extraction of spans inside a hateful text. In this task, we repurpose a part of the publicly available Hate Normalization dataset [3], with each data point containing at least one hate span. The competition ran for a month, from July 13, 2023, to August 16, 2023, PST. Hosted on Kaggle, the task received 72 submissions from 12 teams.

Observations As opposed to a single label per input sentence in a general NLP classification setup, for *HateNorm*, we had a label per token of the sentence [4]. Given the sequential nature of the output, we observed initial hesitation among participants in working with the dataset. However, their engagement improved once a starter kit/codebook was shared. We also observed that among the submissions that submitted a demo paper, the base architecture was more than just a large language model (LLM)-based classifier. There was a mixed usage of both LLM and Bi-directional LSTMs. Further, we noticed that half of the teams did not apply a CRF layer to capture the sequential encoding of the target label but instead relied on LLM's ability to capture context while making predictions for individual tags. The winning team 'FiRC-NLP' with 8 submissions, obtained macro-F1 scores of 0.53 and 0.58 on the public and private leaderboards, respectively. While this beats the start-kit scores of 0.34, it is comparable to the SpanBERT-BiLSTM-CRF model from Masud et al. [3], which also reported a macro-F1 of 0.58. More work is needed to bring mainstream attention beyond a text classification of hatefulness to detecting spans. Shared task venues like HASOC and SemEval are the steps in the right direction.

[2]https://hasocfire.github.io/hasoc/2023/

2. Related Work

Owing to the relevance and need for computational methods to tackle hate speech, we now have a plethora of datasets [5, 6, 7] and techniques [8, 9] exploring the same. Regarding explicit hate speech, hate lexicons [10, 11] have been explored. Auxiliary tasks such as hate normalization [3, 12, 13] and rationale prediction [14] underpinned by the presence or absence of hateful phrases in a sentence led to the foray of hate span detection. In English, the task has been explored from the point of view of detecting toxic and offensive spans [1, 14, 3, 12]. In low-resource settings, the span detection has been explored for Vietnamese [15]. Via the MUDES model, Ranasinghe and Zampieri [16] explored the cross-lingual applicability of hateful span detection when trained on English span datasets. In the multimodal aspect, video frames that conveyed hatefulness were employed as hate spans [17]. Another work detected phrases and sentences in long articles that contribute to hate [18]. In other areas of social computing, span detection has been explored under the English [2] and multilingual [19] factual claim detection settings. Meanwhile, detecting tokens a model pays attention to while labeling a sample as hateful has been employed in posthoc explanations [20].

Table 2
Examples of hateful posts from Masud et al. [3] dataset and their corresponding *BIO* tags depicting harmful spans.

Text	Span
lol what a stupid k*k*	{O, O, O, B, I}
@user text me fa**ot.	{O, O, O, B}
sad to say but I do not trust shit I know how bi****s operate	{O, O, O, O, O, B, I, I, I, O, O, O, B, O}

3. Dataset

This task employed the existing dataset from Masud et al. [3] curated initially for 3 processes – hate intensity prediction, hate span prediction, and hate normalization generation. We employ only the subset of samples labeled for hate span prediction for hosting *HateNorm*. This led to a dataset with 3027 explicitly hateful sentences marked with hate spans. As outlined in Table 2, the spans are tagged via the BIO notation, marking the beginning and inclusion of span tokens as othering, marking the exclusion. Note that a single token can be a span with a corresponding 'B' tag. Meanwhile, an 'I' tag is always preceded by a 'B' tag. The 2421 train samples contained 4695 unique spans with an average of 1.939 spans per training instance. Figure 1 outlines the distribution of the number of spans of a given length, and the majority of spans are ≤ 5 in length. In the train set, each row contained an 'id | space-separated token | list of span indices | space-separated gold span label.' Meanwhile, the 606 test instances were divided into 182 public leader board and 424 privately held instances.

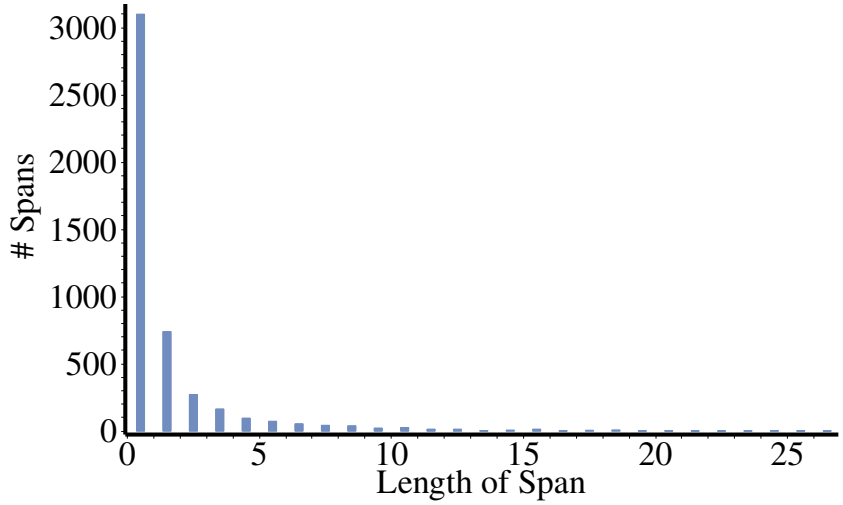

Figure 1: The frequency plot of length of spans in train set. Most samples have only 1 word spans, and majority have spans of length ≤ 5.

Id	Predicted Span List
100	O O O B I
200	O O O O
606	O O O O O B I I I O O O B O

Table 3
Submission format of test.csv with predictions corresponding to inputs enlisted in Table 2.

4. Task Details

Hosting. *HateNorm* was hosted as a Kaggle[3] Competition from 13th July 2023 to 16th August 2023 PST. It received participation from 12 teams, leading to 72 submissions (an average of 6 submissions per team) throughout the competition. As a part of the Kaggle competition, participants were given a sample codebook and a sample 'submission.csv,' as outlined in Table 3. We required the 'id | space-separated predicted label' for the submission file.

Evaluation Metric. Unlike the classification of a single instance that can be adjudged via accuracy or macro-F1, span detection requires evaluating the correct ordering of spans, 'B' following a 'I' and 'O' being the default. To capture this sequential nature of label prediction for individual tokens, we employ the seqeval macro-F1 metric [21]. We hosted the custom metric of seqeval as a script and loaded that to set up the competition so that each incoming submission, by default, gets evaluated via seqeval macro-F1. Further, held 70% of the test samples were private, based on which the final rankings were revealed after the contest. During the contest, the participants saw their rank compared against the public leaderboard 30%. Note that a public leader board does not mean test cases are public.

Baselines. The codebook provided to the participant's finetuned a DistillBERT+FNN setup which reported a extremely low macro-F1 of 0.36. Meanwhile, the baselines provided by Masud

[3]https://www.kaggle.com/competitions/hatenorm23

Rank (Change in Rank)	Team Name	macro-F1	# Submissions
1 (-)	FiRC-NLP	0.57605	8
2 (↑3)	Mohammadmostafa78	0.51382	2
3 (↓1)	CNLP-NITS-PP	0.50888	22
4 (↓1)	IRLab@IITBHU	0.50861	9
5 (↓1)	Niranjan Rao	0.49563	4
6 (↓1)	TextShield	0.45661	5

Table 4
Top-6 teams based on seqeval macro-F1 on the private leader baord. The ranks are the final rank obtained on private leaderboard, and change in rank is how many positions a team moved up or down in the table when the final ranks were computed on the private board. We also enlist the number of submissions made by the team during the competition.

et al. [3] consisted of BiLSTM+CRF with a macro-F1 of 0.44, and the best method[4] being a SpanBERT [22]+BiLSTM+CRF system that reported a macro-F1 of 0.58.

5. Submitted System

Table 4 enlists the top 6 submissions. Among the participating teams that shared the overview notes, we observed that 'FiRC-NLP' employed an ensemble of SpanBERT + CRF with teacher enforcing. When run under lowercase preprocessing, the setup led to the highest macro-F1 of 0.58. The SpanBERT-based method, 'FiRC-NLP,' is also at par with the SpanBERT system of the baseline solution [3]. Note that owing to a one-to-one mapping of tokens to span tags, we discouraged the users from performing additional preprocessing. Meanwhile, the second-best team 'Mohammadmostafa78' with a macro-F1 of 0.52, overcame the skewness in BIO notations by converting the label space to only BO and employing an XLM-RoBERTa [23]+FNN setup. The third highest scoring teams, 'CNLP-NITS-PP' and 'IRLab@IITBHU,' have a macro-F1 of 51, differing only fourth decimal place. However, both employ distinct methods. While the former employs a BERT+BiLSTM+FNN setup, the latter employs contextual embedding (Glove) based BiLSTM+CRF setup akin to the existing baseline. Similar to the observations in our baseline solutions, we observe that BiLSTM and contextual embedding-based solutions perform considerably well. Overall, while Transformer systems either in the form of BERT or SpanBERT help improve the performance, a BiLSTM system trained via CRF is equally viable. We also observe that the proposed systems submissions more or less follow the performance trends of the existing baseline solutions, further corroborating that combining transformer-based systems with CRF and BiLSTM attention mechanisms is the optimal way to detect hateful spans.

6. Conclusion

Despite engaging with malicious content, some online users are adaptable and can be persuaded to change their beliefs through empathy and corrective conduct. Through this task, we aimed

[4]Note: We excluded the Elmo based system from baseline due to reproducibility issues with Elmo on both Tensorflow and Pytorch.

to help these users whose social interactions can eventually be nudged to becoming non-hateful. We believe that the proposed systems can be effectively utilized to assist the moderators.

References

[1] J. Pavlopoulos, J. Sorensen, L. Laugier, I. Androutsopoulos, SemEval-2021 task 5: Toxic spans detection, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Association for Computational Linguistics, Online, 2021, pp. 59–69. URL: https://aclanthology.org/2021.semeval-1.6. doi:10.18653/v1/2021.semeval-1.6.

[2] M. Sundriyal, A. Kulkarni, V. Pulastya, M. S. Akhtar, T. Chakraborty, Empowering the fact-checkers! automatic identification of claim spans on Twitter, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7701–7715. URL: https://aclanthology.org/2022.emnlp-main.525. doi:10.18653/v1/2022.emnlp-main.525.

[3] S. Masud, M. Bedi, M. A. Khan, M. S. Akhtar, T. Chakraborty, Proactively reducing the hate intensity of online posts via hate speech normalization, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 3524–3534. URL: https://doi.org/10.1145/3534678.3539161. doi:10.1145/3534678.3539161.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[5] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on Twitter, in: Proceedings of the NAACL Student Research Workshop, Association for Computational Linguistics, San Diego, California, 2016, pp. 88–93. URL: https://aclanthology.org/N16-2013. doi:10.18653/v1/N16-2013.

[6] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, Proceedings of the International AAAI Conference on Web and Social Media 11 (2017) 512–515. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14955.

[7] A. Kulkarni, S. Masud, V. Goyal, T. Chakraborty, Revisiting hate speech benchmarks: From data curation to system deployment, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 4333–4345. URL: https://doi.org/10.1145/3580305.3599896. doi:10.1145/3580305.3599896.

[8] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: L.-W. Ku, C.-T. Li (Eds.), Proc. of the Fifth International Workshop on Natural Language Processing for Social Media, ACL, Valencia, Spain, 2017, pp. 1–10. URL: https://aclanthology.org/W17-1101. doi:10.18653/v1/W17-1101.

[9] A. Founta, L. Specia, A survey of online hate speech through the causal lens, in: A. Feder, K. Keith, E. Manzoor, R. Pryzant, D. Sridhar, Z. Wood-Doughty, J. Eisenstein, J. Grimmer,

R. Reichart, M. Roberts, U. Shalit, B. Stewart, V. Veitch, D. Yang (Eds.), Proceedings of the First Workshop on Causal Inference and NLP, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 74–82. URL: https://aclanthology.org/2021.cinlp-1.6. doi:10.18653/v1/2021.cinlp-1.6.

[10] M. Polignano, G. Colavito, C. Musto, M. de Gemmis, G. Semeraro, Lexicon enriched hybrid hate speech detection with human-centered explanations, in: Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, UMAP '22 Adjunct, Association for Computing Machinery, New York, NY, USA, 2022, p. 184–191. URL: https://doi.org/10.1145/3511047.3537688. doi:10.1145/3511047.3537688.

[11] V. Stamou, I. Alexiou, A. Klimi, E. Molou, A. Saivanidou, S. Markantonatou, Cleansing & expanding the HURTLEX(el) with a multidimensional categorization of offensive words, in: K. Narang, A. Mostafazadeh Davani, L. Mathias, B. Vidgen, Z. Talat (Eds.), Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH), Association for Computational Linguistics, Seattle, Washington (Hybrid), 2022, pp. 102–108. URL: https://aclanthology.org/2022.woah-1.10. doi:10.18653/v1/2022.woah-1.10.

[12] J. Pavlopoulos, L. Laugier, A. Xenos, J. Sorensen, I. Androutsopoulos, From the detection of toxic spans in online discussions to the analysis of toxic-to-civil transfer, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3721–3734. URL: https://aclanthology.org/2022.acl-long.259. doi:10.18653/v1/2022.acl-long.259.

[13] V. Agarwal, Y. Chen, N. Sastry, Haterephrase: Zero- and few-shot reduction of hate intensity in online posts using large language models, 2023. arXiv:2310.13985.

[14] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, A. Mukherjee, Hatexplain: A benchmark dataset for explainable hate speech detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 14867–14875.

[15] P. G. Hoang, C. D. Luu, K. Q. Tran, K. V. Nguyen, N. L.-T. Nguyen, ViHOS: Hate speech spans detection for Vietnamese, in: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 652–669. URL: https://aclanthology.org/2023.eacl-main.47. doi:10.18653/v1/2023.eacl-main.47.

[16] T. Ranasinghe, M. Zampieri, MUDES: Multilingual detection of offensive spans, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, Association for Computational Linguistics, Online, 2021, pp. 144–152. URL: https://aclanthology.org/2021.naacl-demos.17. doi:10.18653/v1/2021.naacl-demos.17.

[17] M. Das, R. Raj, P. Saha, B. Mathew, M. Gupta, A. Mukherjee, Hatemm: A multi-modal dataset for hate video classification, Proceedings of the International AAAI Conference on Web and Social Media 17 (2023) 1014–1023. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/22209. doi:10.1609/icwsm.v17i1.22209.

[18] L. Zhou, A. Caines, I. Pete, A. Hutchings, Automated hate speech detection and span extraction in underground hacking and extremist forums, Natural Language Engineering 29 (2023) 1247–1274. doi:10.1017/S1351324922000262.

[19] S. Mittal, M. Sundriyal, P. Nakov, Lost in translation, found in spans: Identifying claims in multilingual social media, arXiv:2310.18205 (2023).

[20] B. Kennedy, X. Jin, A. Mostafazadeh Davani, M. Dehghani, X. Ren, Contextualizing hate speech classifiers with post-hoc explanation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5435–5442. URL: https://aclanthology.org/2020.acl-main.483. doi:10.18653/v1/2020.acl-main.483.

[21] H. Nakayama, seqeval: A python framework for sequence labeling evaluation, 2018. URL: https://github.com/chakki-works/seqeval, software available from https://github.com/chakki-works/seqeval.

[22] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, SpanBERT: Improving pre-training by representing and predicting spans, Transactions of the Association for Computational Linguistics 8 (2020) 64–77. URL: https://aclanthology.org/2020.tacl-1.5. doi:10.1162/tacl_a_00300.

[23] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747. doi:10.18653/v1/2020.acl-main.747.