WikiDetox Visualization

Visualizing toxicity on Wikipedia

Iris Qu
Jigsaw
xiaoyuq@google.com

Nithum Thain
Jigsaw
nthain@google.com

Yiqing Hua
Cornell Tech
yh663@cornell.edu

ABSTRACT

Discussions on Wikipedia are crucial places for editors to coordinate their work of curating the world's knowledge, unfortunately, it's also a place where toxicity and harassment exists. While there is a growing concern among the community, it's still difficult to discover, track and moderate toxic comments in a timely manner. **WikiDetox Visualization** is a data visualization and a moderation tool that aims to help to address this issue.

The project is made possible by **Conversation AI** team at **Google Jigsaw** -- a team of researchers and engineers working towards ending online harassment. In the early days or our research, the Conversation AI team partnered with **Wikimedia Foundation** on the **WikiDetox** project – an initiative to identify toxicity on Wikipedia and analyse the impact of harassment at scale. From this initiative and a few others, the Conversation AI team was able to create **Perspective API** - a machine learning API that scores toxicity levels for text comments. With Perspective, we are able to close the loop on the WikiDetox project, and build a tool to visualize and moderate discussions on Wikipedia at scale. Our hope is to engage the community in the effort of detoxifying Wikipedia.

CCS CONCEPTS

Human-centered computing → Visualization → Visualization application domains → Information visualization

KEYWORDS

Wikipedia, Wikipedia discussions, content moderation, talk page categorization, conversation reconstruction

1. Background

1.1 The WikiDetox Project

Facing the growing concern over toxicity on Wikipedia [5], Wikimedia Research, in collaboration with Jigsaw's **Conversation AI** team, developed tools for automated detection of toxic comments using machine learning models [4]. These models allow us to analyze the dynamics and impact of comment-level harassment in Wikipedia discussions at scale.

Two distinct types of data were used in the initial research [6]: a corpus of all 95 million user and article page diffs made between 2001–2015 scored by the personal attack model; and a human annotated dataset of 1m crowd-sourced annotations that cover 100k talk page diffs, with 10 judgements per diff. This paper sets a foundation for understanding toxicity on Wikipedia at scale, which led to analysis of the prevalence of personal attacks, characteristics of attackers, and moderation methods.

The Conversation AI [1] team at Jigsaw leveraged this dataset alongside several others to train machine learning models for online conversation released under the **Perspective API** [3]. This API scores comments based on their likelihood of being found toxic by online annotators. Additionally, models are trained to detect subtypes of toxicity including toxicity subtypes including: flirtation, threat, identity attack, insult, sexually explicit, obscene and severe toxicity. The visualization surfaces most toxic subtypes by month, we'll discuss the trends in data insights section.

The goal of this collaboration was to discover how machine-learnt insights can be applied to help the community improve the level of engagement, debate and activity taking place. The WikiDetox visualization aims to assist toward this goal by having toxicity on Wikipedia be more trackable and actionable, as a step towards ameliorating its effect.

2. Data Processing

2.1 Conversation Reconstruction

In order to understand toxicity on Wikipedia, we need to understand toxic trends are moderation rates over time. Wikipedia raw comment data provides basic metadata like page title, timestamp and user id. However, the nature of conversation on Wikimedia is a series of edits to a markup page, so it is often unclear exactly how the conversation unfolded. In order to understand how comments were added, modified, and deleted through time we need an algorithm to reconstruct this information from the raw Wikipedia data dumps.

The visualization uses reconstructed English comments based on the method outlined in the conversation reconstruction method

[3]. The method joins revision data from the Wikipedia data dumps with previous reconstructed page states, and computes new page states in the format of action_id and action_offset. Then we are able to identify action types in five categories: **creation, addition, modification, restoration,** and **deletion.** With the reconstructed attributes, we can then compute the latest actions on a page.

The reconstruction method outputs the following schema:

- **id**: a unique string of three integers - the id of the revision that the action was recorded, the offset of the starting position where the action took place on the new page, and on the previous page.
- **content**: the cleaned comment text associated with the action.
- **type**: the type of the action from the six categories listed above.
- **parent_id**: the id of the direct parent action that contributed to the original content of the comment. Thus, section creation and comment additions don't have a parent. For modification, removal and restoration, the parent action represents the original content that was modified, removed, or restored.
- **ancestor_id**: the id of the original action from which all the derivations come from.
- **indentation**: number of indentation symbols at the beginning of the content.
- **replyTo_id**: the id of the action that the current action is replying to, reply relation inferred using indentations.
- **conversation_id**: id of the conversation that the action belongs to.
- **user_id**: id of user that took the action. Returns null if user is not registered.
- **user_text**: username or IP address of user that took the action.
- **authors**: a list of pairs of id and username information from the Wikipedia editors who changed the content of the action.
- **timestamp**: the timestamp of the action.
- **page_id**: the id of the page where the action took place
- **page_title**: title of the page where the action took place.
- **rev_id**: id of the revision when the action took place.

To prepare reconstructed data for the visualization, we filter the comments to the most recent revisions (for each group of comments that shares an **ancestor_id**, we take the most recent **revision_id**). That way we have a table of unique comments reflecting the latest page status. We then visualize every comment with toxicity levels greater than 0.8. The **type** attribute determines "**toxic**" and "**detoxed**" property - when the type equals "deletion", we consider the comment detoxed.

2.2 Talk page categorization

We want to maximize user engagement for detoxification, and users often feel more inclined to act on topics that are relevant to them. One proposed solution is to categorize talk pages where toxic comments live into categories, so users can track and filter comments by "trending topics".

Based on the data schemas listed in the conversation reconstruction method, the best candidates for categorization are: (1) the comment content or (2) the page where the action took place (**page_title**). After initial rounds of data analysis, we saw that the toxic comments are often off-topic, thus using the first method did not lead to concise categories. By using the page titles in combination with Google's Natural Language API, we are able to recover an appropriate categorization for the page on which this content takes place.

Wikipedia does provide a categorization system for some of its articles, but it is difficult to extract a single root category from all direct subcategories to a given page. One possible solution is to get the most significant root category by computing the shortest distance to any direct page category -- but it's extremely expensive to compute when we account for the number of categories we are dealing with. With this approach, the computation can end up in a loop, or have multiple root categories that are equally significant.

With that in mind, we developed a method to categorize pages into meaningful categories with the **Google Cloud natural language API**[1], in the following steps:

1. Get direct subcategories for each unique talk page with **Wikimedia API**:

   ```
   https://en.wikipedia.org/w/api.php?format=json&acti
   on=query&prop=categories&cllimit=max&titles=[P
   ```

2. Clean result categories to delete entries irrelevant to page content. These entries usually contain keywords:

   ```
   ['Wikipedia', 'AC with', 'CS1', 'Good articles',
   'Articles', 'All articles', 'Pages', 'Use mdy dates from',
   'Use dmy dates from', 'English from', 'Webarchive
   template', 'births', 'deaths', 'Redirects']
   ```

   Example cleaned results:
   Bonsai Kitten:

   ```
   ["2000 hoaxes", "Cats in popular culture",
   "Entertainment websites", "Fiction about animal
   cruelty", "Fictional cats", "Fictional companies",
   "Humorous hoaxes in science"]
   ```

Watergate:

```
["Watergate scandal", "Nixon administration controversies", "20th-century scandals", "Political controversies", "Political scandals in the United States", "Political terminology of the United States", "1970s in the United States", "News leaks"]
```

3. Run cleaned categories and page title as one string through the Google Cloud Natural Language API. The API returns up to 3 relevant category/subcategory/sub-sub category combos with confidence level greater than 0.5. If the page categories are not sufficient for categorization, this step will return null.
Example returns:
Bonsai Kitten:

```
/Arts & Entertainment - Confidence:  0.73
/People & Society - Confidence:  0.68
/Hobbies & Leisure - Confidence:  0.65
```

Watergate:

```
/News/ Gossip & Tabloid News / Scandals & Investigations - Confidence:  0.97
/Sensitive Subjects - Confidence:  0.93
/News/Politics - Confidence:  0.63
```

[1] https://cloud.google.com/natural-language/

page_title	category1	sub_category1	subsub_category1
Horary astrology	People & Society	Religion & Belief	
Whakapohai River	Shopping	Consumer Resources	Coupons & Discount Offers
Promote Mandarin Council	Reference	Language Resources	Foreign Language Resources
Glossary of French expressions in English	Reference	Language Resources	Foreign Language Resources
Hook flash	Internet & Telecom	Communications Equipment	
Harry Craddock	Food & Drink	Beverages	Alcoholic Beverages
Stanislav Batishchev	Beauty & Fitness	Fitness	
Elbs reaction	Food & Drink	Food	
Lago Petroleum Corporation	Business & Industrial	Energy & Utilities	Oil & Gas
George Joannides	Law & Government	Public Safety	
Analytical Feminism	People & Society	Social Issues & Advocacy	
Piaçabuçu Environmental Protection Area	People & Society	Social Issues & Advocacy	
Vinica, Tomislavgrad	Shopping	Consumer Resources	Coupons & Discount Offers
First Australian Building Society	Business & Industrial	Construction & Maintenance	
Dogwood Alliance	People & Society	Social Issues & Advocacy	Green Living & Environmental Issues
Valentines, Virginia	Hobbies & Leisure	Special Occasions	
Catherine Gardner	Arts & Entertainment	Celebrities & Entertainment News	
Smbat Shahaziz	Books & Literature	Poetry	
Prodromos Meravidis	Arts & Entertainment	Entertainment Industry	Film & TV Industry

Figure 1: Example talk page category results, showing category/ subcategory/ sub-sub category combo with highest confidence

The visualization takes **category - subcategory** combos with the highest numbers of toxic comments for a given month, and show them as "top trends" for that month. Figure 1 shows some example results using this method.

2.3 Data Streaming

On top of building a visualization for data analysis, we want to add real time moderation features so toxic comments can be taken down in a timely manner. To accomplish this task, we need to stream real time data from Wikipedia, and perform the reconstruction as follows:

1. Stream new comment action data from Wikipedia API.
2. Compare new actions with stored page states, and perform conversation reconstruction (discussed above).
3. Score comments with the Perspective API.
4. If the comment is made on a talk page, check if talk page is categorized. If not, perform the talk page categorization process (discussed above).
5. If the action is not new (**ancestor_id != id**), find and delete previous revision with same ancestor_id in the destination table.
6. Add scored, reconstructed and categorized comment to the destination data table.

For the current version of the visualization, all data is hosted on Google Cloud BigQuery. We use a cron job to interact with BigQuery and streams data to destination tables. One potential improvement on this model for better scalability is to move cron job scripts to cloud functions such that:

1. Data is streamed from Wikipedia via a cron job.
2. Inform Google Cloud Pub/Sub of the data change.
3. Achieve conversation reconstruction and talk page categorization with cloud functions.

Figure 2 maps out the backend infrastructure we are currently using:

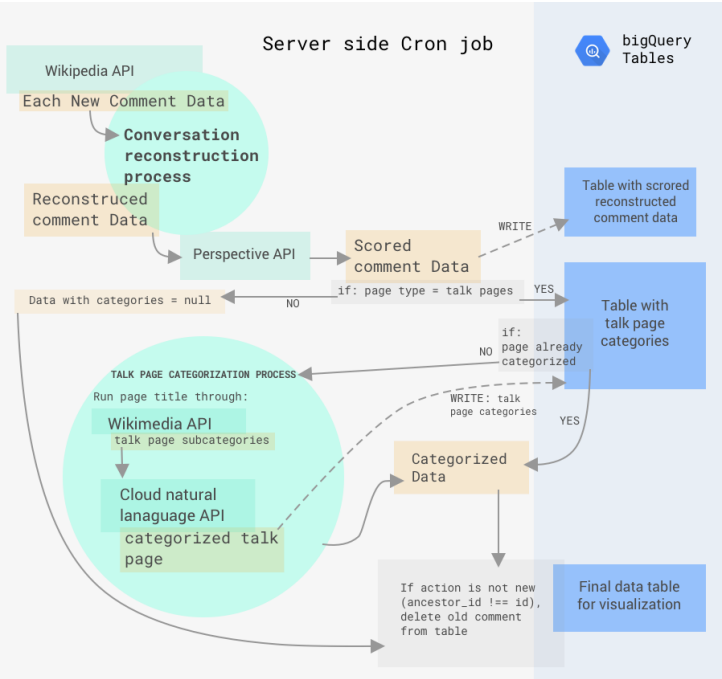

Figure 2: Server side cron job component map

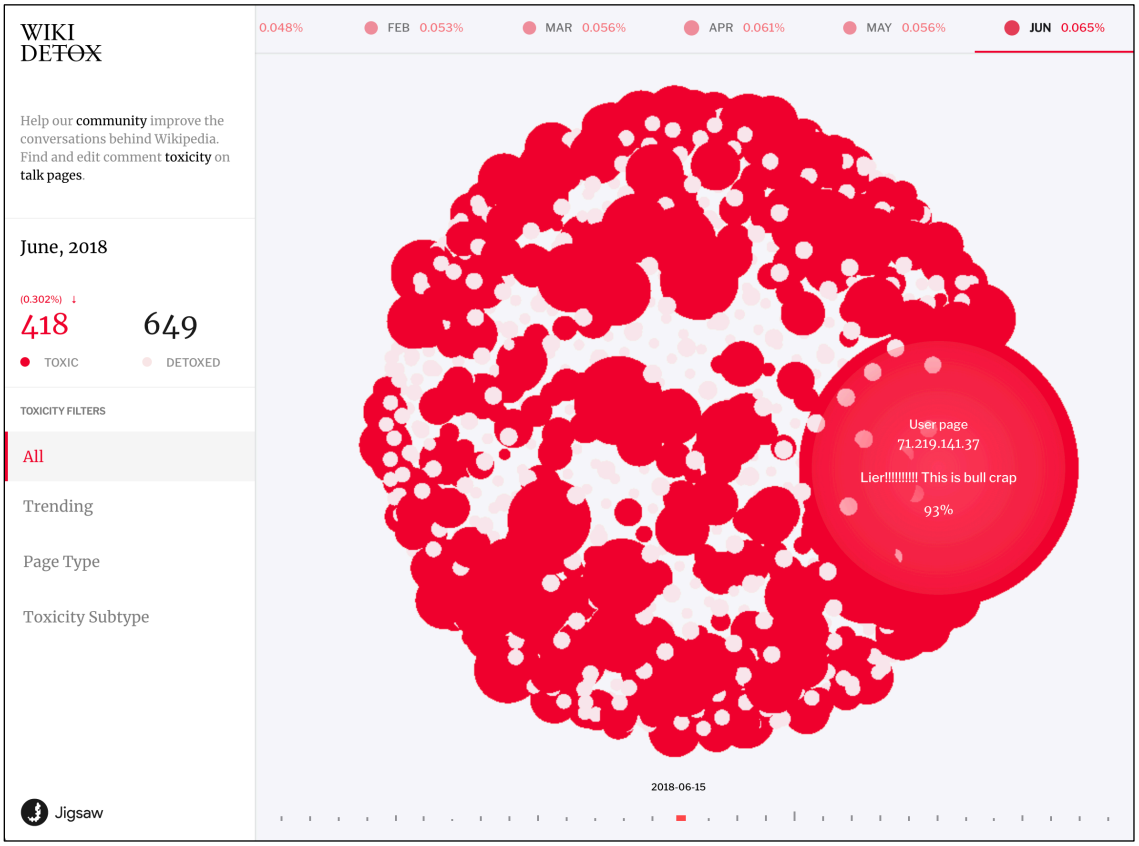

Figure 3: Data visualization default value, showing data for June 2018

3. User interactions

3.1 View, filter and sort data

On initial load of the visualization (Figure 3), users see toxic comments of the most recent month clustered with two different colors – dark red representing toxic comments, and light red representing detoxed comments. The initial globe sorts toxic comments by day, and users can navigate to the daily trend chart located at the bottom of the screen to filter comments by date (figure 4).

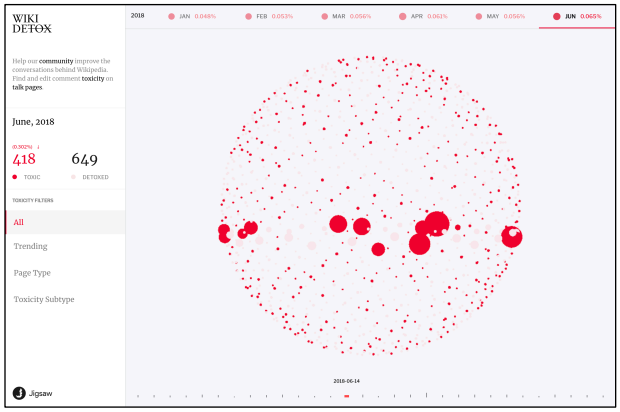

Figure 4: Data filtered by date

Additionally, users can view data from another month by selecting from the monthly trend tabs located at the top of the visualization (see figure 3 and 4). The trend tabs give a glimpse into the toxic volumes as well as toxic trends for the months, accompanied by percentage values indicating monthly toxic ratio. On month selection, information such as volume of toxic and detoxed comments are immediately available on the left-hand panel, with an arrow indicating percentage of toxicity trend comparing to the previous month (Figure 5).

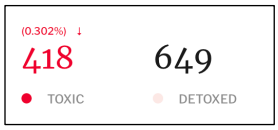

Figure 5: toxicity metrics for June 2018

The left-hand panel also contains toxicity filters for the comment data. When user toggle a filter type, the numeric breakdowns of filter items expand into an ordered list (Figure 6 and 7). When a filter is clicked on, the comments will morph down in volume to meet the filter criteria, resulting in a smaller globe.

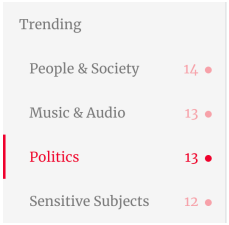

Figure 6: Trending topics for June 2018

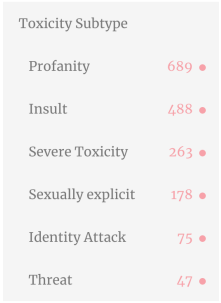

Figure 7: Toxicity subtypes for May 2019

Users can interact with the globe by hovering on the particles. Each particle will expand into a comment on hover, showing information like the page title, comment details and toxicity scores (Figure 3 shows a comment on hover). All UI features are also available on mobile devices with a responsive UI (figure 8).

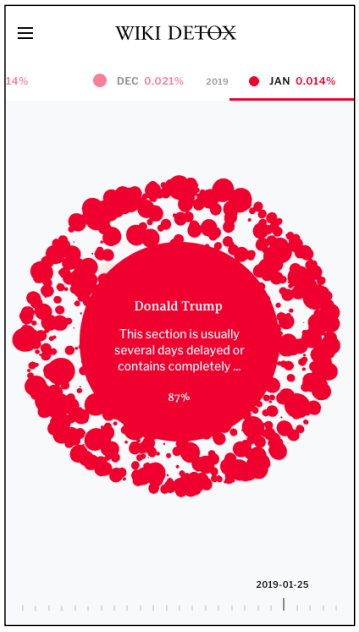

Figure 8: Mobile UI

3.2 Content moderation

One of the goals for this visualization is to engage Wikipedia users to detoxify Wikipedia, so we looked for ways to simplify the editing process for non-editors to contribute. For the visualization,

we created the DetoxAgent[2] bot to make revisions for toxic comments.

From the hover state discussed in previous section, users can click on the comment to expand them into a comment details view (Figure 9). If the user wants to detoxify the selected comment, they can simply click "Detox" button located in the lower right corner. The scripts will then submit a new revision on behalf of the user to delete the toxic comment. Once the comment is "detoxed", the color of the particle will change to light red (Figure 10).

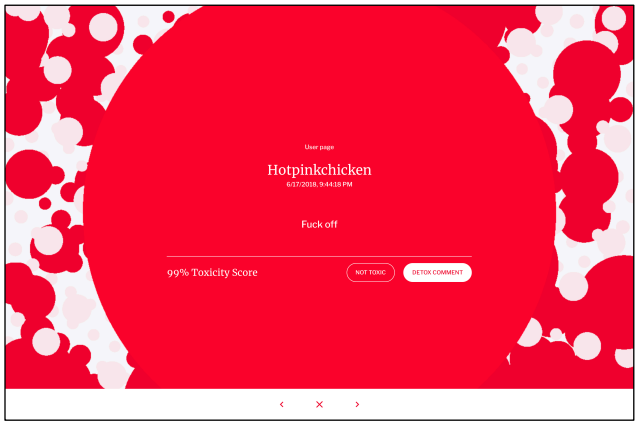

Figure 9: Comment details view - toxic

Figure 10: Comment details view - detoxed

3.3 Feedback mechanism

Perspective API is here to help human moderators to improve online discussions. The alpha version of our perspective model scores comments based on robust training data – but we are constantly looking for ways to improve it. Experiments like WikiDetox visualization present opportunities for us to test our grounds and open a two way stream for feedbacks.

[2] https://en.wikipedia.org/wiki/User:DetoxAgent

The engineers on our team wrote a server to register user feedbacks, and have human readers to evaluate them. If users find comments that are not toxic marked as so in the visualization, they can simply click the "Not toxic" button in the comment details view, and the scripts will send a feedback with the comment text to our server.

4. Tech-stack

4.1 Data Queries

The destination BigQuery table shows Wikipedia comment data from January 2017 onward that are reconstructed and categorized with the latest revisions. The visualization interacts with BigQuery data table via an express server, and performs the following queries to render visualization for an initial load:

- Get all comment data for the most recent month.
- Get the numbers of toxic and detoxed comments per month
- Compute the daily trend of toxic + detoxed comments per day of the most recent month.

When users change the month selection, the server does additional queries to compute toxic data for daily trends and trending topics for the selected month.

4.2 Client-side rendering

This project is set up with Vue CLI 3.0.5, and uses Vue 2.5.17 as the main frontend javascript framework for the benefit of its small bundle size, easy lifecycle management, ES6 support and robust state management. VueJS is a progressive framework that comes with the bare minimum features to build an app, and is extremely flexible and customizable with add-on features. That said, it's very easy to use Vue with other javascript libraries like ThreeJS and D3js that also performs dom manipulation.

In order to efficiently paint large number of particles without pixilation and delays, the point cloud is rendered with ThreeJS buffer geometries written in shaders. The shader scripts buffer attributes inputs to compute particle size, position and color. The positions of the particles are sorted by time on a fibonacci sphere. All particle animations rely on manipulation of the shadow position, color and size attributes.

5. Data insights

5.1 Talk page vs user page

With talk page categorization and toxicity types measured across time, interesting patterns start to emerge. First, we see that user pages are always more toxic than article talk pages, with the toxic

ratio of roughly 2:1 (Figure 11 shows user-to-talk page ratio by month for 2017). One hypothesis is that toxic actions on article talk pages can carry over to user pages, and morph into personal attacks.

month	talkpages count	userpages count	Userpage / talkPage RATIO
2017.01	380	779	2.05
2017.02	349	612	1.75
2017.03	332	734	2.21
2017.04	341	755	2.21
2017.05	370	712	1.92
2017.06	318	637	2
2017.07	290	877	3.02
2017.08	351	605	1.72
2017.09	297	662	2.23
2017.1O	388	734	1.89
2017.11	364	665	1.83
2017.12	350	753	2.15

Figure 11: Toxic talk page to user page ratio broken down by month for 2017

5.2 Talk page categories

We used the talk page categorization method to surface some of the most toxicity topic areas by month. We take all three category levels for every talk page comment, if available, to calculate the most toxic categories and subcategories. When sorting by most toxic categories, the top topics per month usually include: **People & Society, Arts & Entertainment, News and Law and Government**. While these do demonstrate a toxic bias towards certain issues, these categories are too broad to draw many conclusions from. In order to get more specific category data, we are prioritizing subcategories in the final dataset. If a talk page has a category as well as a subcategory, we count it towards the subcategory while excluding it from the parent category. With this method, the top five most toxic categories in 2017 are: "**Music & Audio", "People & Society", "Movies" and "Politics"**. If visualized with undeleted comments only, the most toxic categories get more specific by month. The trending topics for May 2018 are:

- Religion & Belief, 12 items
- Biological Sciences, 10 items
- Movies, 10 items
- Politics, 9 items

CONCLUSION

We hope this tool can help the Wikipedia community to better understand online toxicity, and provide an engaging method for first time users as well as experienced editors to detoxify Wikipedia. Instead of having the visualization as an end product, we hope to continue experimenting with detoxification methods. We also believe our work on conversation reconstruction and talk page categorization can benefit the community for future research and experiments.

REFERENCES

[1] Conversation AI. https://conversationai.github.io/.
[2] E Wulczyn, N Thain, and L Dixon, 2017, April. Ex machina: Personal attacks seen at scale. In Proceedings of the 26th International Conference on World Wide Web (pp. 1391-1399). International World Wide Web Conferences Steering Committee.
[3] Y Hua, C Danescu-Niculescu-Mizil, D Taraborelli, N Thain, J Sorensen and L Dixon, 2018. WikiConv: A Corpus of the Complete Conversational History of a Large Online Collaborative Community. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 2818-2823).
[4] Perspective API Reference. https://github.com/conversationai/perspectiveapi/blob/master/api_reference.md
[5] Research: Detox. https://meta.wikimedia.org/wiki/Research:Detox.
[6] Support and Safety Team. Harassment Survey. Wikimedia Foundation, 2015. https://upload.wikimedia.org/wikipedia/commons/5/52/Harassment_Survey_2015_-_Results_Report.pdf.