TECHNISCHE UNIVERSITÄT DRESDEN

Dominik Pataky

Faculty of Computer Science, Institute of Systems Architecture, Chair of Privacy and Data Security

CrowdFilter

Client-side Filtering of Web Content

Großer Beleg // Dresden, 19th June, 2018

The Problem

- Websites and platforms host user-generated content
- Content needs to be regulated
  - Regional laws, illegal content, copyright claims
  - Community rules, discussion moderation
- Users might leave if community is unmoderated
- States pressure companies to act:
  fines (NetzDG), hosting illegal content is legal risk
- Content regulation gets outsourced to third-parties and algorithms

Current consequence

No regulation-control mechanism. Premature deletion of content.

The Problem

- Websites and platforms host user-generated content
- Content needs to be regulated
  - Regional laws, illegal content, copyright claims
  - Community rules, discussion moderation

- Users might leave if community is unmoderated
- States pressure companies to act:
  fines (NetzDG), hosting illegal content is legal risk
- Content regulation gets outsourced to third-parties and algorithms

Current consequence

No regulation-control mechanism. Premature deletion of content.

The Problem

- Websites and platforms host user-generated content
- Content needs to be regulated
  - Regional laws, illegal content, copyright claims
  - Community rules, discussion moderation

- Users might leave if community is unmoderated
- States pressure companies to act:
  fines (NetzDG), hosting illegal content is legal risk
- Content regulation gets outsourced to third-parties and algorithms

Current consequence

No regulation-control mechanism. Premature deletion of content.

The Problem

- Websites and platforms host user-generated content
- Content needs to be regulated
  - Regional laws, illegal content, copyright claims
  - Community rules, discussion moderation

- Users might leave if community is unmoderated
- States pressure companies to act:
  fines (NetzDG), hosting illegal content is legal risk
- Content regulation gets outsourced to third-parties and algorithms

Current consequence

No regulation-control mechanism. Premature deletion of content.

The Problem

- Websites and platforms host user-generated content
- Content needs to be regulated
  - Regional laws, illegal content, copyright claims
  - Community rules, discussion moderation

- Users might leave if community is unmoderated
- States pressure companies to act:
  fines (NetzDG), hosting illegal content is legal risk
- Content regulation gets outsourced to third-parties and algorithms

Current consequence

No regulation-control mechanism. Premature deletion of content.

The Problem

- Websites and platforms host user-generated content
- Content needs to be regulated
  - Regional laws, illegal content, copyright claims
  - Community rules, discussion moderation

- Users might leave if community is unmoderated
- States pressure companies to act:
  fines (NetzDG), hosting illegal content is legal risk
- Content regulation gets outsourced to third-parties and algorithms

Current consequence

No regulation-control mechanism. Premature deletion of content.

The Questions

- How do we know regulation is not censorship?
- Should we not discuss all generated content?
- How can small platforms without the man power for moderation be supported?

The Questions

- How do we know regulation is not censorship?
- Should we not discuss all generated content?
- How can small platforms without the man power for moderation be supported?

The Questions

- How do we know regulation is not censorship?
- Should we not discuss all generated content?
- How can small platforms without the man power for moderation be supported?

The Approach

- Content should not be deleted by the platforms → client-side „overlay“
- Needs tools to classify, detect and handle content inside the browser

CrowdFilter

Creating classifications for text.

1. Platform independent, no implementations on web servers needed
2. Balanced data flows, crowdsourced input, no false positives
3. No central authority, client-side options empower users

The Approach

- Content should not be deleted by the platforms $\rightarrow$ client-side „overlay“
- Needs tools to classify, detect and handle content inside the browser

CrowdFilter

Creating classifications for text.

1. Platform independent: no implementations on website needed
2. Balanced data base: crowd-sourced input of classifications
3. No central authority: client-side options empower users

The Approach

- Content should not be deleted by the platforms → client-side „overlay“
- Needs tools to classify, detect and handle content inside the browser

CrowdFilter

Creating classifications for text.

1. Platform independent: no implementations on website needed
2. Balanced data base: crowd-sourced input of classifications
3. No central authority: client-side options empower users

The Approach

- Content should not be deleted by the platforms → client-side „overlay“
- Needs tools to classify, detect and handle content inside the browser

CrowdFilter

Creating classifications for text.

1. Platform independent: no implementations on website needed
2. Balanced data base: crowd-sourced input of classifications
3. No central authority: client-side options empower users

The Approach

- Content should not be deleted by the platforms → client-side „overlay“
- Needs tools to classify, detect and handle content inside the browser

CrowdFilter

Creating classifications for text.

1. Platform independent: no implementations on website needed
2. Balanced data base: crowd-sourced input of classifications
3. No central authority: client-side options empower users

The Approach

- Content should not be deleted by the platforms → client-side „overlay“
- Needs tools to classify, detect and handle content inside the browser

CrowdFilter

Creating classifications for text.

1. Platform independent: no implementations on website needed
2. Balanced data base: crowd-sourced input of classifications
3. No central authority: client-side options empower users

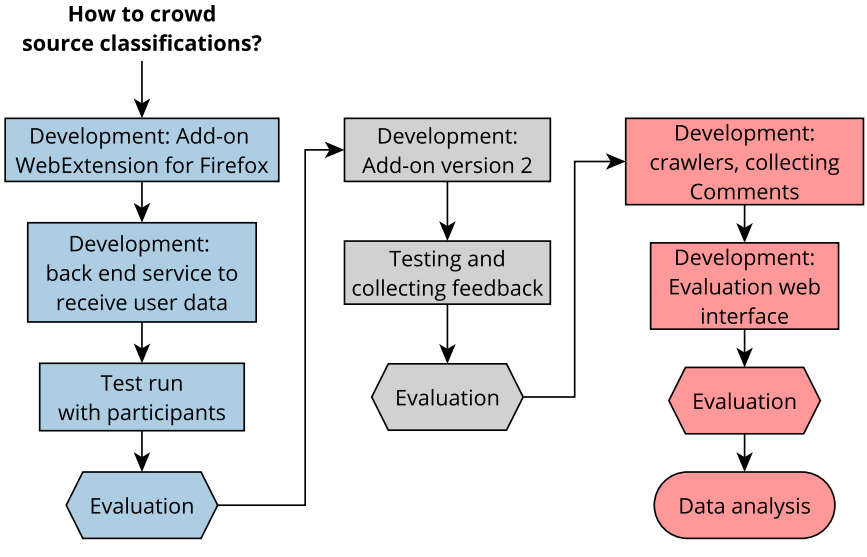


Figure 1: Project development iterations

Version 1: Button for specific sites

- Websites use DOM elements for content: paragraphs, forum board posts, Tweets
- Idea: inject a button into these elements with a list of classifications → clicking on a classification sends it to the back end server
- Three platforms for testing:
  Github, Twitter and Heise Forums

Version 1: Button for specific sites

- Websites use DOM elements for content: paragraphs, forum board posts, Tweets
- Idea: inject a button into these elements with a list of classifications → clicking on a classification sends it to the back end server
- Three platforms for testing:
  Github, Twitter and Heise Forums

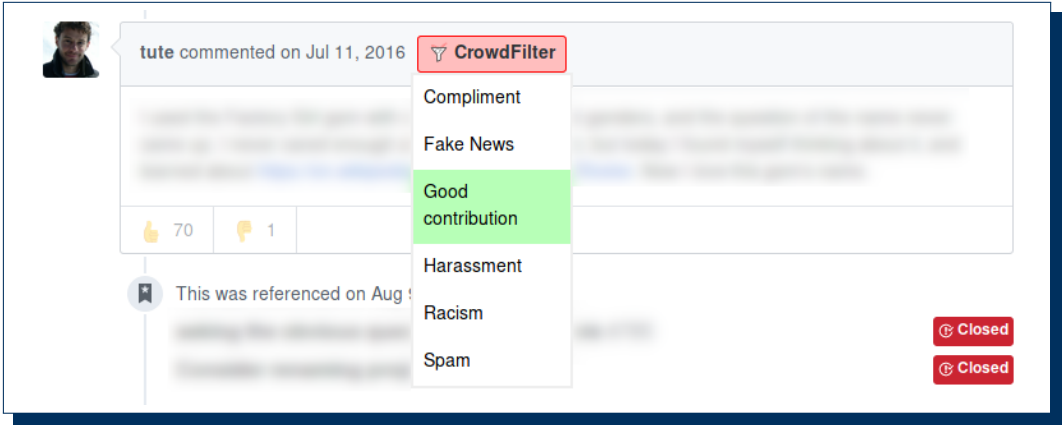


Figure 2: Add-on version 1 on github.com

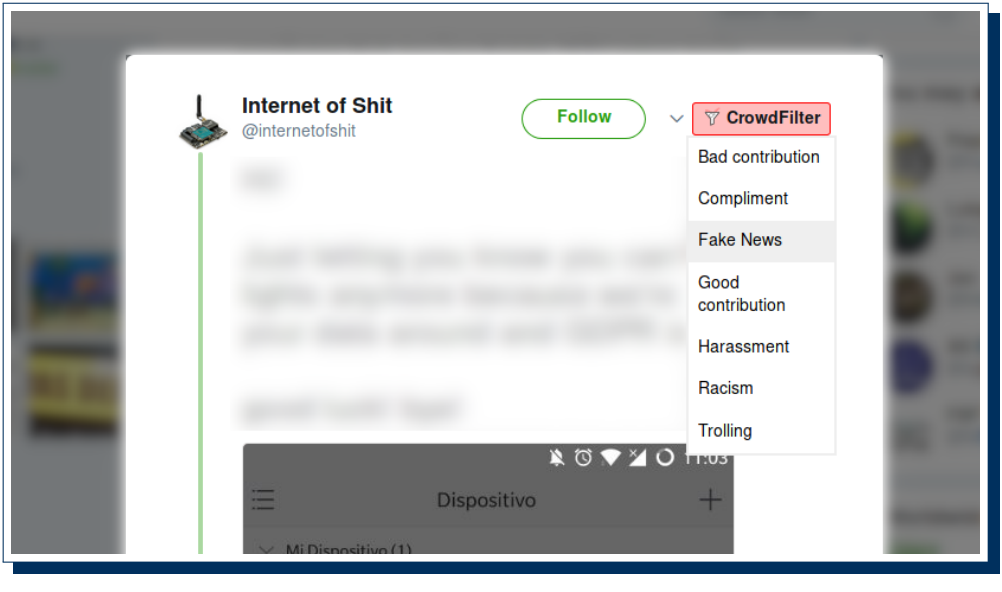


Figure 3: Add-on version 1 on twitter.com

Evaluation add-on version 1

1. DOM elements known: meta data can be extracted
2. Drop down classifications list visible in view
3. Needs script for each website the add-on is used on
4. Updates necessary if website changes layout
5. Handling e.g. AJAX needs workarounds which can fail

Evaluation add-on version 1

1. DOM elements known: meta data can be extracted
2. Drop down classifications list visible in view

3. Needs script for each website the add-on is used on
4. Updates necessary if website changes layout
5. Handling e.g. AJAX needs workarounds which can fail

Version 2: generic text selection

- Idea: remove website-specific injection and use a more generic implementation
- The WebExtension API offers handling of text selection and the context menu
- Implementation: list of classifications as context menu
- Introduction with a video appeared after add-on installation

Version 2: generic text selection

- Idea: remove website-specific injection and use a more generic implementation
- The WebExtension API offers handling of text selection and the context menu
- Implementation: list of classifications as context menu
- Introduction with a video appeared after add-on installation

Version 2: generic text selection

- Idea: remove website-specific injection and use a more generic implementation
- The WebExtension API offers handling of text selection and the context menu
- Implementation: list of classifications as context menu
- Introduction with a video appeared after add-on installation

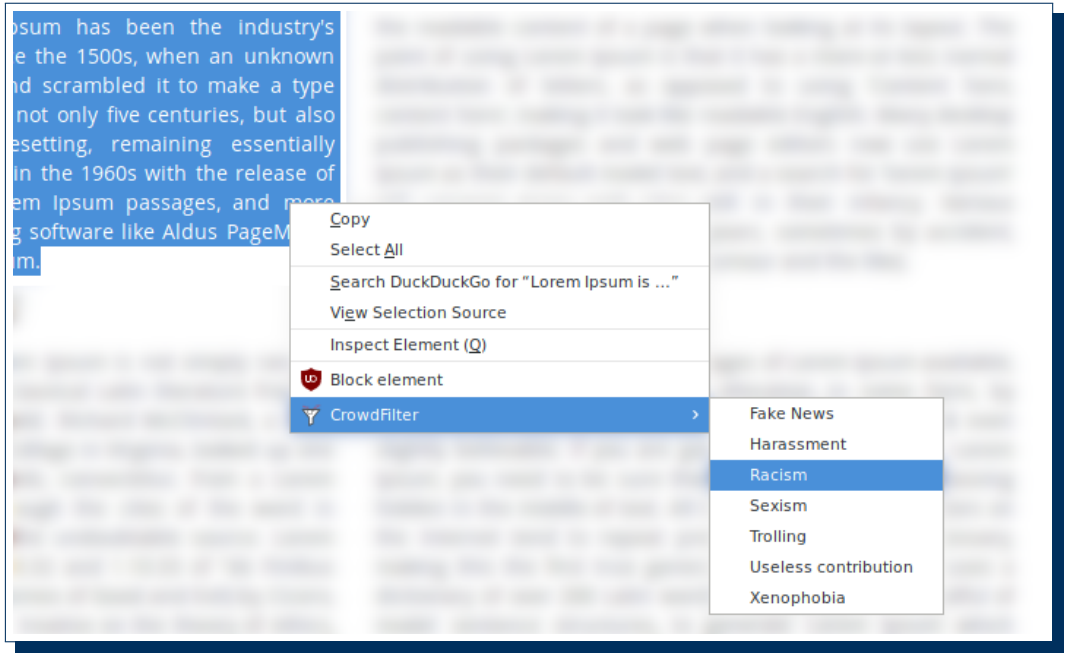


Figure 4: Add-on version 2 with generic text selection

Evaluation add-on version 2

1. Works on every website by default
2. No visual injections, easy to use
3. Does not ask for any permissions during installation

4. Usage might not be clear without introduction
5. Only collects the selected text, no meta data except URL → nearly no context

Evaluation add-on version 2

1. Works on every website by default
2. No visual injections, easy to use
3. Does not ask for any permissions during installation

4. Usage might not be clear without introduction
5. Only collects the selected text, no meta data except URL $\rightarrow$ nearly no context

The back end: Collector

- Written in Python (Flask, PostgreSQL), hosted at TUD
- Offered a HTTPS and Tor Hidden Service endpoint
- Display of live data during the collection phase

Conclusion

- Implementation dependent on many diverse users
- Incentives for participation key feature
- Both versions have their pros and cons
- Regarding push back: starting with empty database makes generation of first results difficult

Conclusion

- Implementation dependent on many diverse users
- Incentives for participation key feature
- Both versions have their pros and cons
- Regarding push back: starting with empty database makes generation of first results difficult

Conclusion

- Implementation dependent on many diverse users
- Incentives for participation key feature
- Both versions have their pros and cons
- Regarding push back: starting with empty database makes generation of first results difficult

Conclusion

- Implementation dependent on many diverse users
- Incentives for participation key feature
- Both versions have their pros and cons
- Regarding push back: starting with empty database makes generation of first results difficult

Developing a different approach

- The add-on provided a useful channel between users and our database
- But the participation showed weaknesses of this approach
- This iteration tries to find another solution for crowd-sourced classification of text content

The aim remains

Finding an applicable way to populate a database with classified text content.

Developing a different approach

- The add-on provided a useful channel between users and our database
- But the participation showed weaknesses of this approach
- This iteration tries to find another solution for crowd-sourced classification of text content

The aim remains

Finding an applicable way to populate a database with classified text content.

Developing a different approach

- The add-on provided a useful channel between users and our database
- But the participation showed weaknesses of this approach
- This iteration tries to find another solution for crowd-sourced classification of text content

The aim remains

Finding an applicable way to populate a database with classified text content.

Developing a different approach

- The add-on provided a useful channel between users and our database
- But the participation showed weaknesses of this approach
- This iteration tries to find another solution for crowd-sourced classification of text content

The aim remains

Finding an applicable way to populate a database with classified text content.

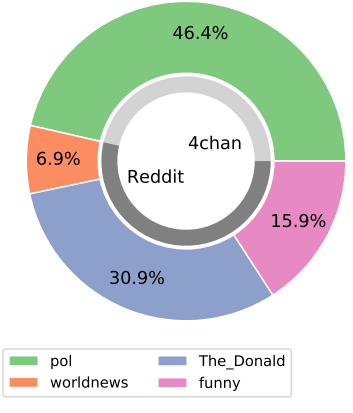
Crawling

- Sources for text content: Reddit */r/The_Donald*, */r/funny* and */r/worldnews* and the 4chan */pol/* board.
- Both websites offer APIs for content crawling
- Crawler scripts written in Python fetched submissions, stored in database with some meta data

Crawling

- Sources for text content: Reddit */r/The_Donald*, */r/funny* and */r/worldnews* and the 4chan */pol/* board.
- Both websites offer APIs for content crawling
- Crawler scripts written in Python fetched submissions, stored in database with some meta data

Crawled Comments sources



Source	Comments	Share
4chan /pol/	105.427	46,4%
Reddit	121.707	53,6%
▷ funny	36.008	(15,9%)
▷ the_donald	70.074	(30,9%)
▷ worldnews	15.625	(6,9%)
Total	227.134	

Figure 5: Shares of Comment sources from Reddit and 4chan

Web interface for crowd-source evaluations

- Based on the collected Comments pool
- Cleans up Comment text
- Presents the user with Sampleset with 10 Comments:
  3 from 4chan, 4 from Reddit and 3 recycled
- Identification with 4-digit PIN, reusable to continue evaluating

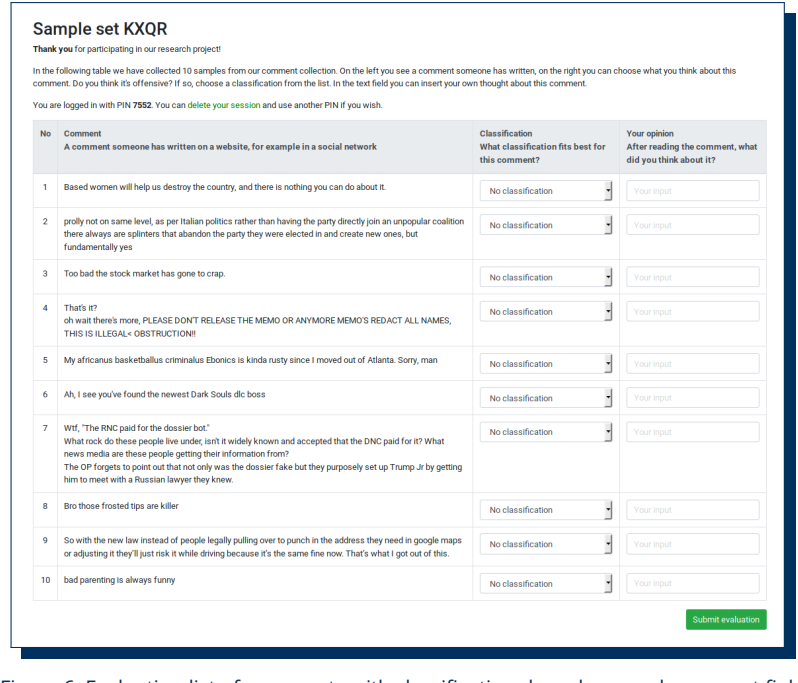


Figure 6: Evaluation list of comments with classification drop down and comment field

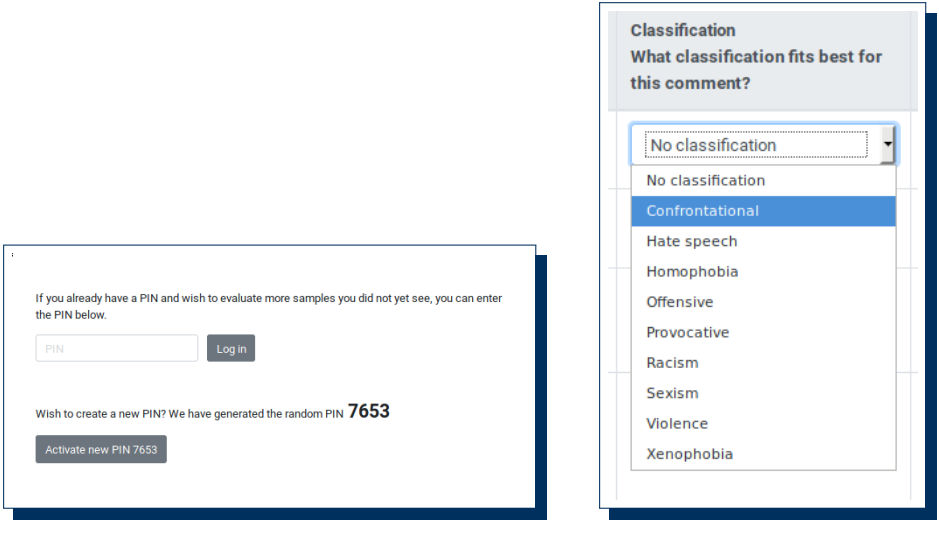


Figure 7: Screenshots of (a) the PIN login form and (b) the classification drop down

Evaluation

- Tool seems more approachable, resulting in more Classifications than in iteration 1 and 2
- Feedback received from participants
  1. Interface is simple enough to be understood instantly
  2. Automated loading of new comments encourages continuing
  3. Classification without context difficult (irony, satire)
  4. Using labels without explanation unclear
  5. Positive labels also welcome

Evaluation

- Tool seems more approachable, resulting in more Classifications than in iteration 1 and 2
- Feedback received from participants
  1. Interface is simple enough to be understood instantly
  2. Automated loading of new comments encourages continuing
  3. Classification without context difficult (irony, satire)
  4. Using labels without explanation unclear
  5. Positive labels also welcome

Evaluation

- Tool seems more approachable, resulting in more Classifications than in iteration 1 and 2
- Feedback received from participants
  1. Interface is simple enough to be understood instantly
  2. Automated loading of new comments encourages continuing
  3. Classification without context difficult (irony, satire)
  4. Using labels without explanation unclear
  5. Positive labels also welcome

Evaluation

- Tool seems more approachable, resulting in more Classifications than in iteration 1 and 2
- Feedback received from participants
  1. Interface is simple enough to be understood instantly
  2. Automated loading of new comments encourages continuing
  3. Classification without context difficult (irony, satire)
  4. Using labels without explanation unclear
  5. Positive labels also welcome

Evaluation

- Tool seems more approachable, resulting in more Classifications than in iteration 1 and 2
- Feedback received from participants
  1. Interface is simple enough to be understood instantly
  2. Automated loading of new comments encourages continuing
  3. Classification without context difficult (irony, satire)
  4. Using labels without explanation unclear
  5. Positive labels also welcome

Evaluation

- Tool seems more approachable, resulting in more Classifications than in iteration 1 and 2
- Feedback received from participants
  1. Interface is simple enough to be understood instantly
  2. Automated loading of new comments encourages continuing
  3. Classification without context difficult (irony, satire)
  4. Using labels without explanation unclear
  5. Positive labels also welcome

Classifications

- 748 Evaluations in total, 539 (72%) „No classification“
- 209 Classifications over 12 labels

Classifications

- 748 Evaluations in total, 539 (72%) „No classification“
- 209 Classifications over 12 labels

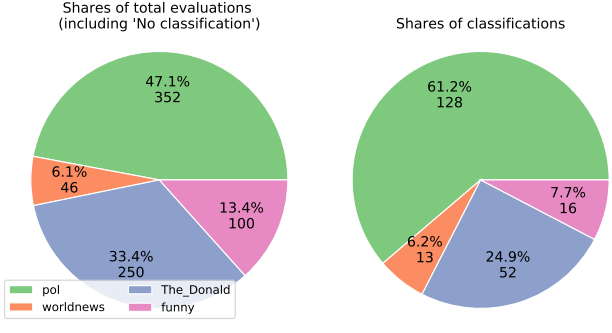


Figure 8: Classification usage shares by comment source

Usage of labels in Classifications

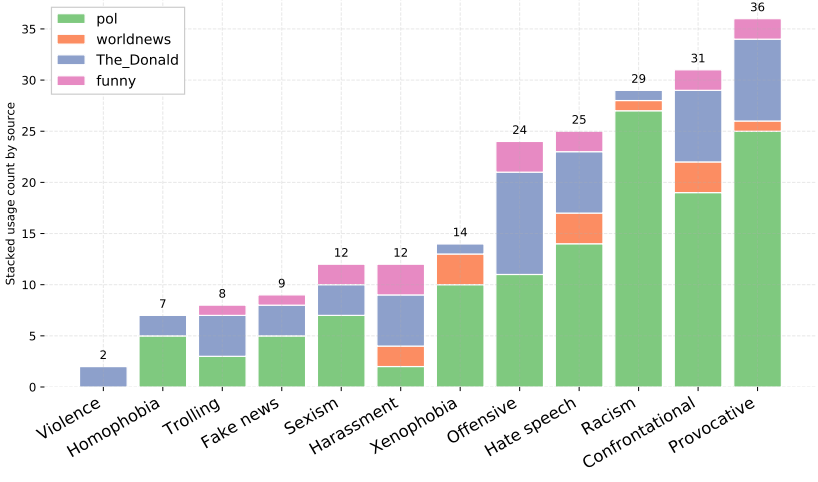


Figure 9: Classification label usages

Looking at meta data

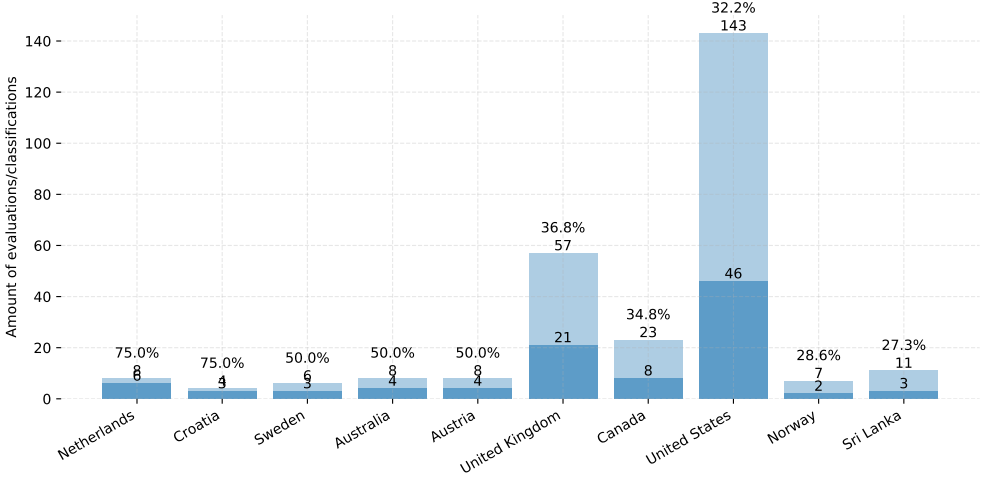


Figure 10: Percentage of Evaluations versus Classifications on 4chan /pol/ by country meta tag

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Deciding on a list of non-binary labels

1. What labels should be used?
2. Should positive labels be included as well?
3. If labels are more generic than others, should they be visualised in a hierarchical structure?
4. Should the interface provide an input element for custom labels?
5. And if yes, should it recommend existing labels?
6. How could user biases be distinguished?
7. Should the user be able to classify a Comment with multiple labels?

Conclusion

- Initial aim: shift content handling from platform to user backed by a crowd-sourced classification of content
- Browser add-on applicable tool, but high entry barrier
- Classification of pre-collected comments more accessible approach
- Combination of classification interface and add-on most promising
- Label choice difficult to decide → needs interdisciplinary cooperation

Conclusion

- Initial aim: shift content handling from platform to user backed by a crowd-sourced classification of content
- Browser add-on applicable tool, but high entry barrier
- Classification of pre-collected comments more accessible approach
- Combination of classification interface and add-on most promising
- Label choice difficult to decide → needs interdisciplinary cooperation

Conclusion

- Initial aim: shift content handling from platform to user backed by a crowd-sourced classification of content
- Browser add-on applicable tool, but high entry barrier
- Classification of pre-collected comments more accessible approach
- Combination of classification interface and add-on most promising
- Label choice difficult to decide → needs interdisciplinary cooperation

Conclusion

- Initial aim: shift content handling from platform to user backed by a crowd-sourced classification of content
- Browser add-on applicable tool, but high entry barrier
- Classification of pre-collected comments more accessible approach
- Combination of classification interface and add-on most promising
- Label choice difficult to decide → needs interdisciplinary cooperation

Conclusion

- Initial aim: shift content handling from platform to user backed by a crowd-sourced classification of content
- Browser add-on applicable tool, but high entry barrier
- Classification of pre-collected comments more accessible approach
- Combination of classification interface and add-on most promising
- Label choice difficult to decide → needs interdisciplinary cooperation

Related projects

- Add-ons against Fake News:
  FiB (Princeton 2016), Open Mind (Yale 2017)
- Social Bots: „Social Bots, Fake News und Filterblasen“ (Michael Kreil, 34C3)
- Platforms: Twitter and Facebook roll out UI features for reporting bad content (2017/2018)
- Hate speech: real time analysis of Tweets (Antwerp and Hildesheim, 2018)
- Hiding instead of deletion: Github rolls out „minimized comments“ (2018)

Sources, references and additional information

These slides and the associated report with further references will be published on my website `https://dpataky.eu` and can be used under the CC BY-SA 4.0 license.