

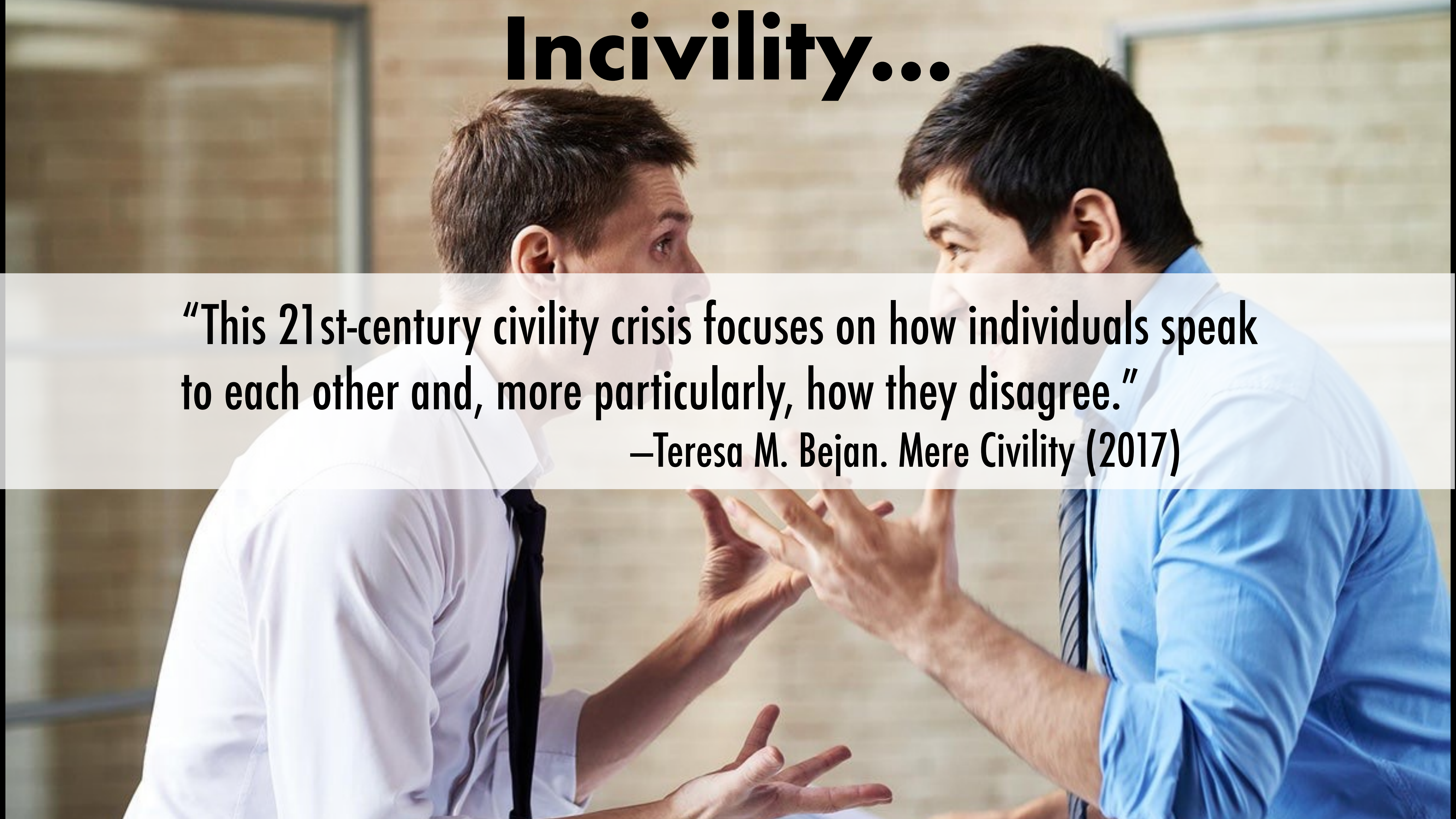
Incivility...



Incivility...

“This 21st-century civility crisis focuses on how individuals speak to each other and, more particularly, how they disagree.”

—Teresa M. Bejan. Mere Civility (2017)



Characterizing and Detecting Incivility in Open Source Code Review Discussions



Isabella Ferreira
Polytechnique Montréal



Jinghui Cheng
Polytechnique Montréal



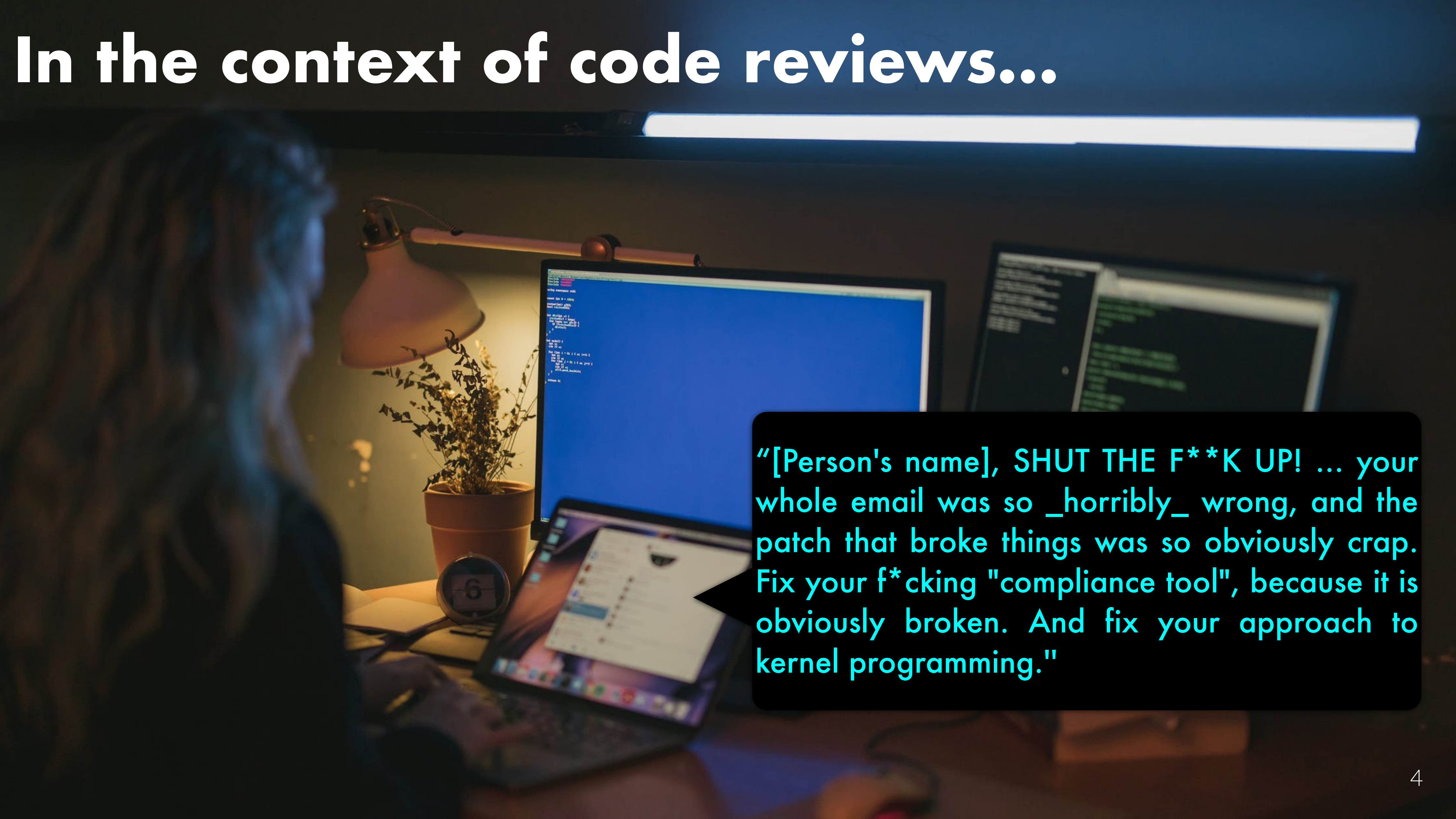
Bram Adams
Queen's University

Incivility is characterized by
heated discussions that involve
personal attacks and
unnecessary disrespectful tone

In the context of code reviews...



In the context of code reviews...



"[Person's name], SHUT THE F**K UP! ... your whole email was so _horribly_ wrong, and the patch that broke things was so obviously crap. Fix your f*cking "compliance tool", because it is obviously broken. And fix your approach to kernel programming."

We don't know much about incivility yet

```
public class User{
    private Long id;
    private String firstName;
    private String lastName;

    public User(Long id, String firstName, String lastName){
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public Long getId(){
        return id;
    }

    public void setId(Long id){
        this.id = id;
    }
}
```

- projects
 - git
 - conf
 - src
 - ?
- test
- target
- resources
- com
 - ac.search
 - ac.rat
 - ac.breck

We don't know much about incivility yet



```
public class User{
    private Long id;
    private String firstName;
    private String lastName;

    public User(Long id, String firstName, String lastName){
        this.id=id;
        this.firstName=firstName;
        this.lastName=lastName;
    }

    public Long getId(){
        return id;
    }

    public void setId(Long id){
        this.id=id;
    }
}
```

- projects
 - git
 - conf
 - src
 - ?
- test
- target
- resources
- com
 - ac.search
 - ac.rat
 - ac.breck

How can we create healthier working environments in OSS development?

★ — 1. Which **features of discussion** can be
★ — found in code review discussions?
★ —

```
public class User{
    private Long id;
    private String firstName;
    private String lastName;

    public User(Long id, String firstName, String lastName){
        this.id=id;
        this.firstName=firstName;
        this.lastName=lastName;
    }

    public Long getId(){
        return id;
    }

    public void setId(Long id){
        this.id=id;
    }
}
```

```
projects
├── .git
├── .conf
├── src
├── ?
├── test
├── target
├── resources
├── com
├── ac.search
├── ac.rat
└── ac.breck
```


★ — 1. Which **features of discussion** can be
★ — found in code review discussions?
★ —



2. **How much incivility** exists in
code review discussions?

★ — 1. Which **features of discussion** can be
★ — found in code review discussions?
★ —



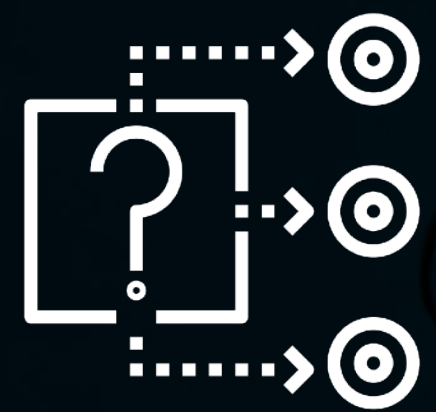
2. **How much incivility** exists in
code review discussions?

☐ ? → ● → ● → ●
3. What are the **causes** and
consequences of incivility?

★ — 1. Which **features of discussion** can be
★ — found in code review discussions?
★ —



2. **How much incivility** exists in
code review discussions?



3. What are the **causes** and
consequences of incivility?



4. How can we create **healthier** working
environments in OSS development?

Analysis on 1,545 code review emails from LKML related to rejected changes

Analysis on 1,545 code review emails from LKML related to rejected changes



We chose to analyze **rejected changes** because:

1. Rejected changes represent **more than 66%** of all changes submitted to LKML (Jiang et al., MSR, 2013)
2. The Linux community often rejects changes using a **harsh language** when reporting the rejection (Alami et al., ICSE, 2019)

Analysis on 1,545 code review emails from LKML related to rejected changes



**Collect code review
emails from LKML**

January 2018 - March 2019



406,719 emails

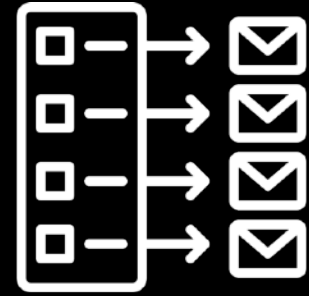
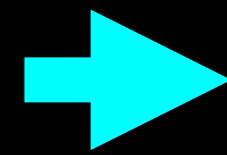
Analysis on 1,545 code review emails from LKML related to rejected changes



Collect code review emails from LKML

January 2018 - March 2019

 406,719 emails



Group individual emails by email threads

Mailboxminer tool

 55,396 email threads

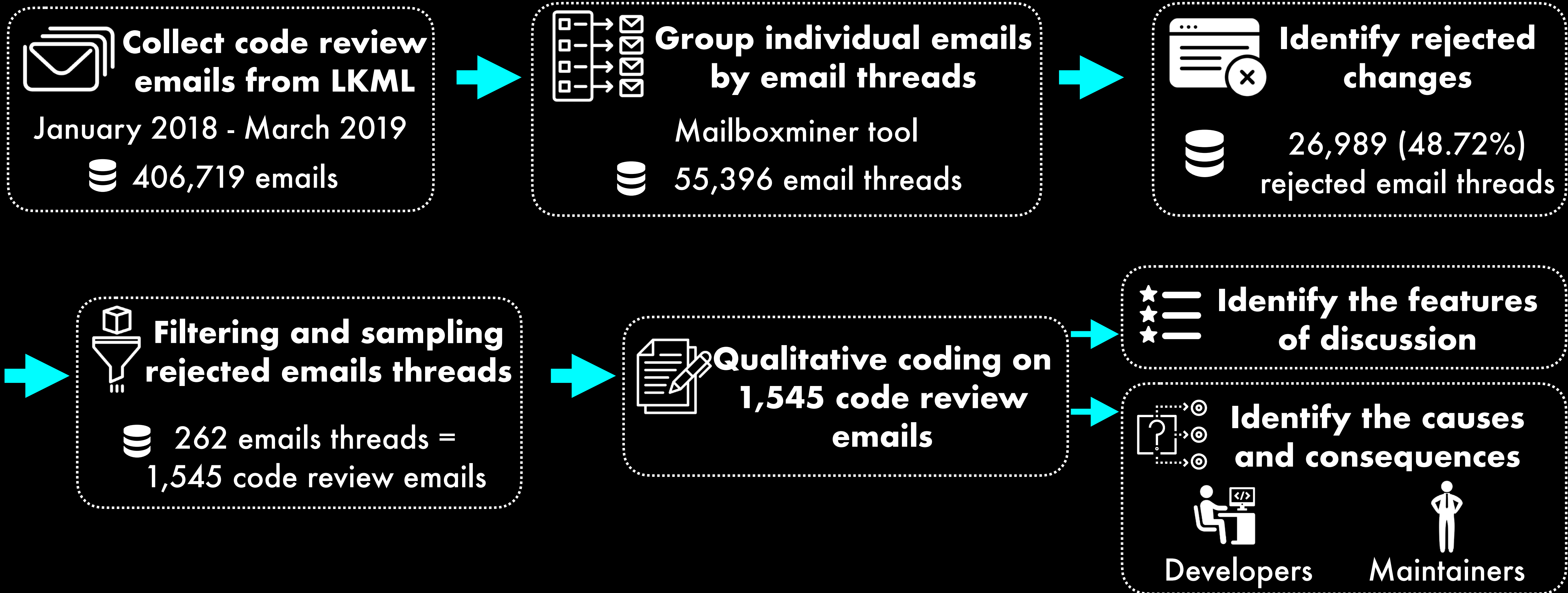
Analysis on 1,545 code review emails from LKML related to rejected changes



Analysis on 1,545 code review emails from LKML related to rejected changes



Analysis on 1,545 code review emails from LKML related to rejected changes

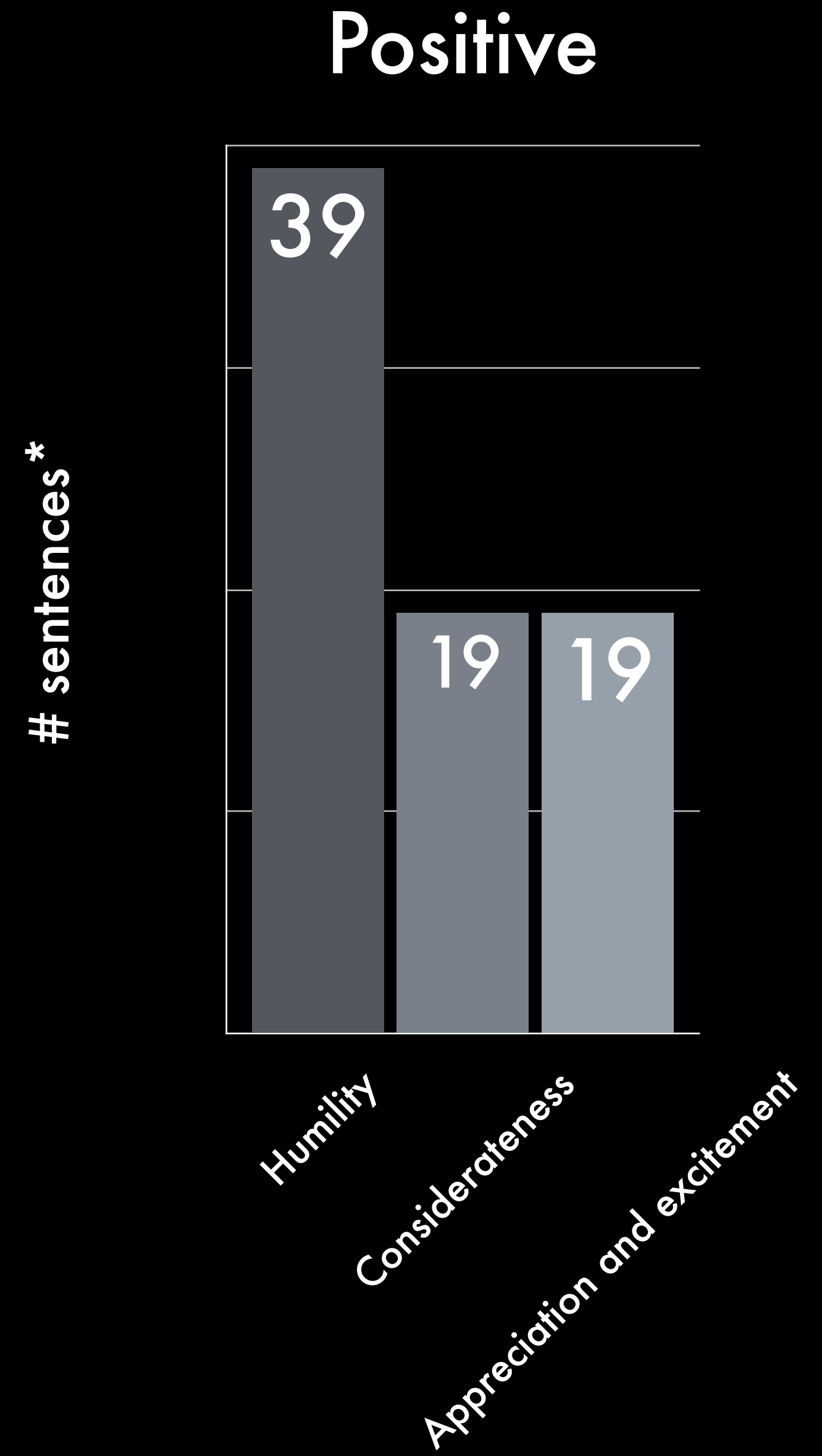




1. Which features of discussion can be found in code review discussions of rejected patches?

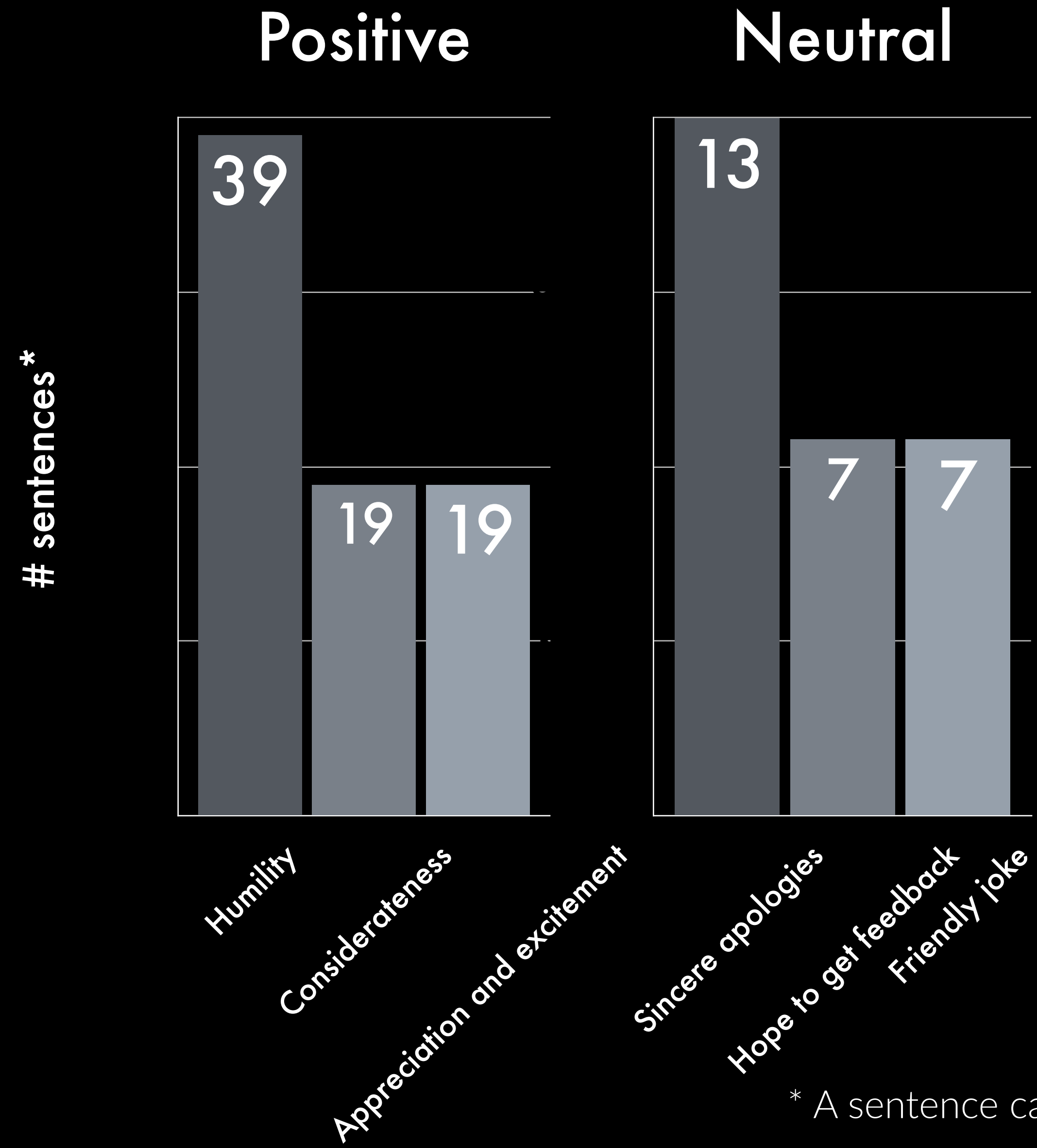
Frustration, name calling and impatience are the most frequent uncivil features

Frustration, name calling and impatience are the most frequent uncivil features



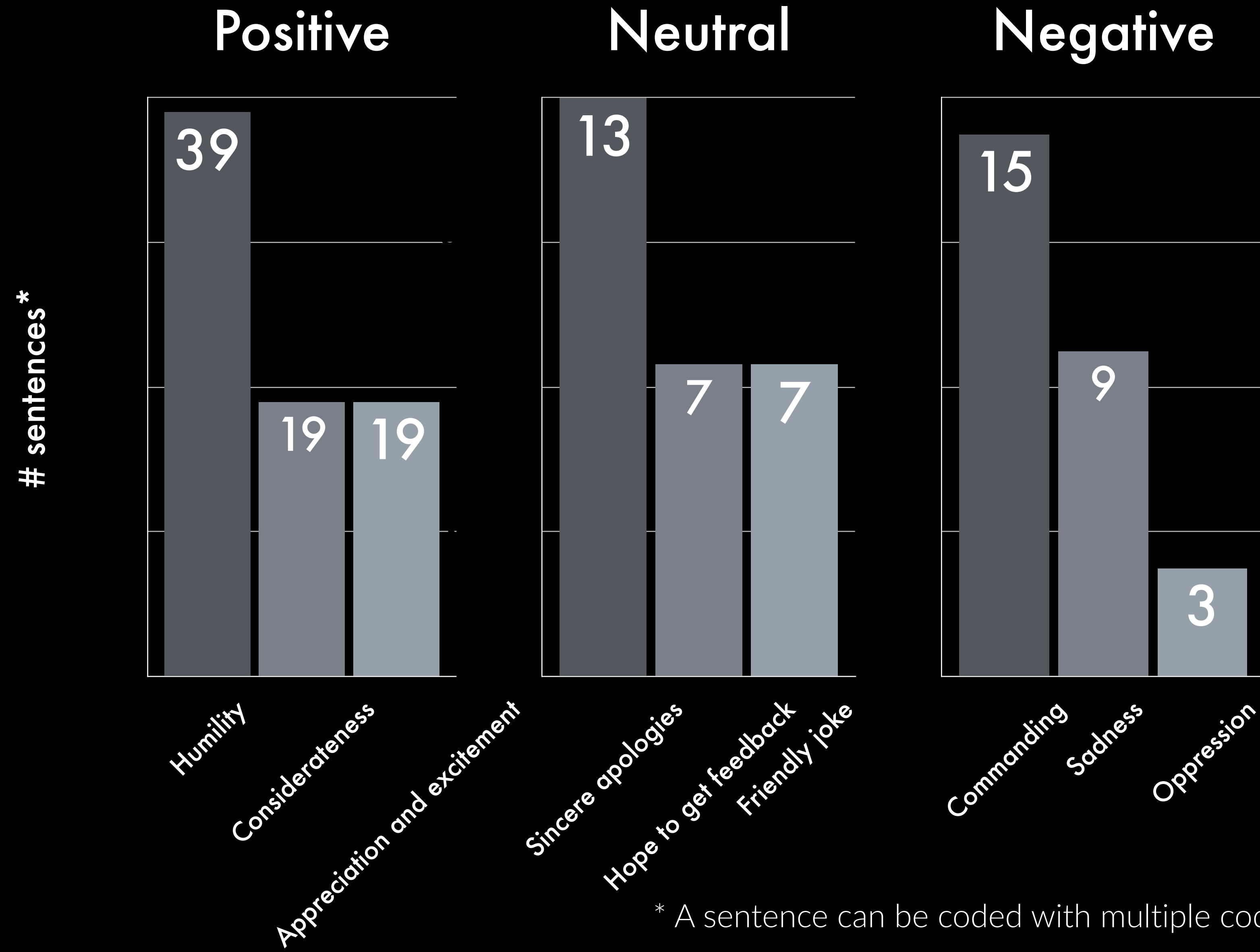
* A sentence can be coded with multiple codes

Frustration, name calling and impatience are the most frequent uncivil features

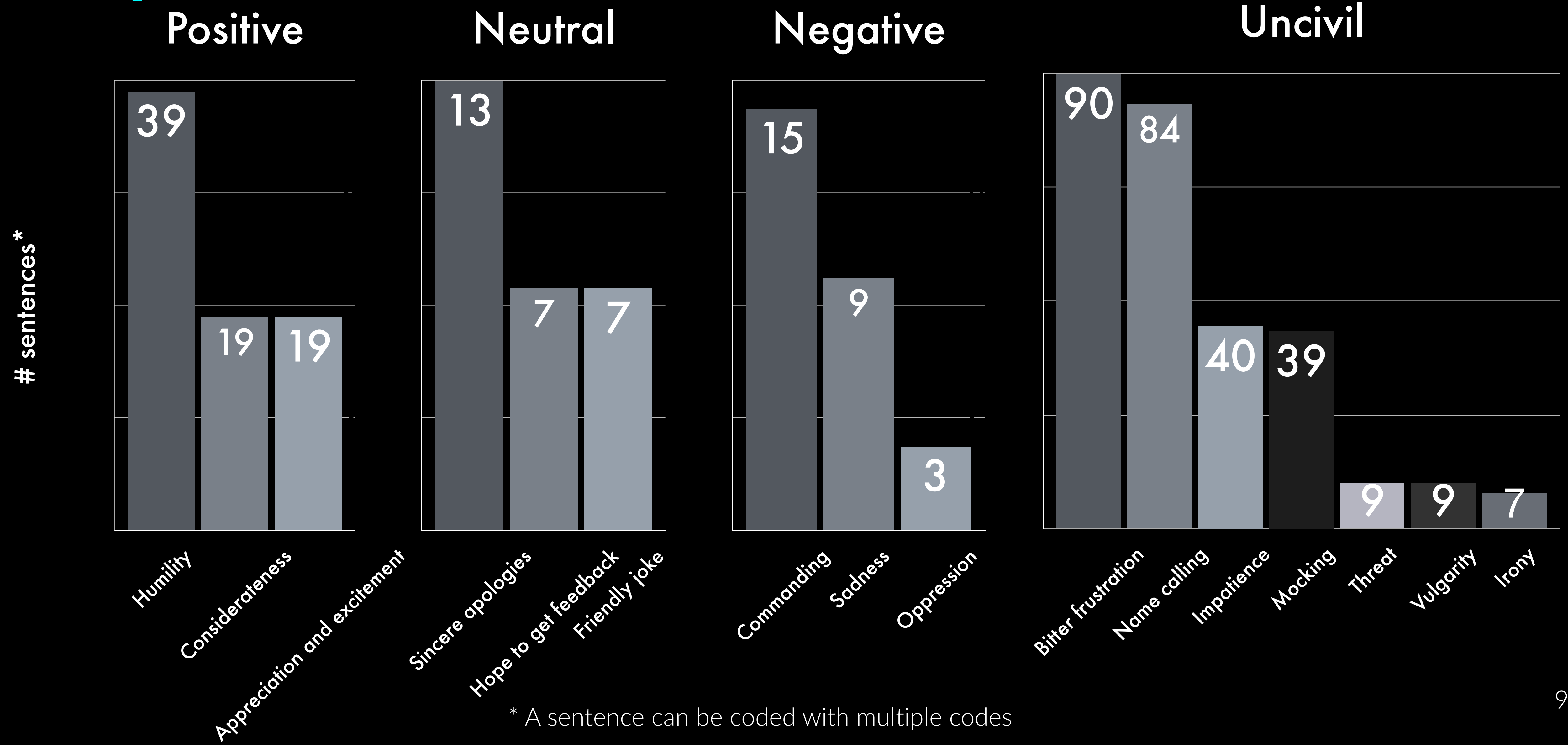


* A sentence can be coded with multiple codes

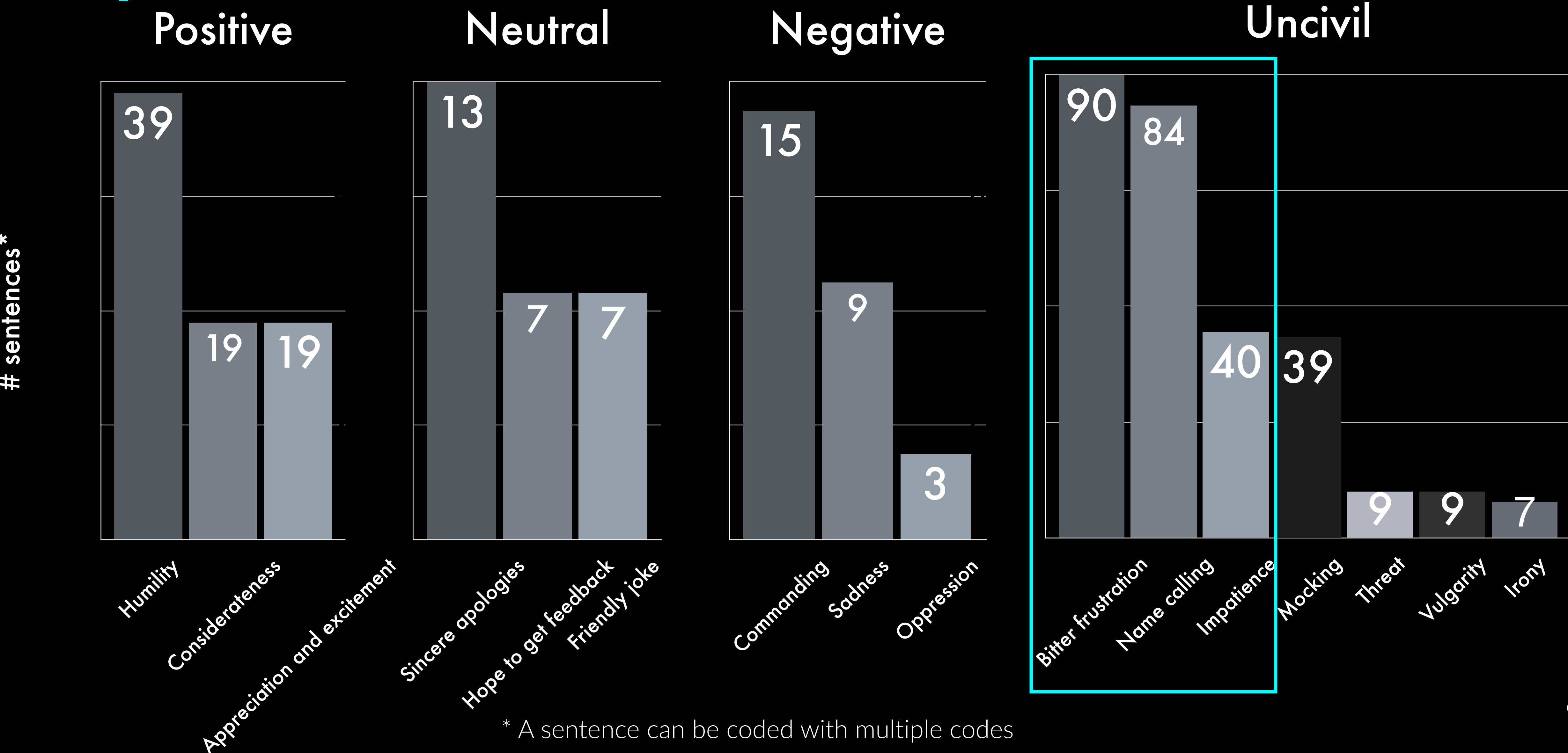
Frustration, name calling and impatience are the most frequent uncivil features



Frustration, name calling and impatience are the most frequent uncivil features



Frustration, name calling and impatience are the most frequent uncivil features





Bitter Frustration

"If people don't care enough about their code to even check the warnings, I'm not going to waste one second pulling the resulting garbage. It's that simple."



Bitter Frustration

"If people don't care enough about their code to even check the warnings, I'm not going to waste one second pulling the resulting garbage. It's that simple."



Name calling

"If you want to provide more accurate documentation then you better come up with something which is helpful instead of a completely useless blurb like the below..."



Bitter Frustration

"If people don't care enough about their code to even check the warnings, I'm not going to waste one second pulling the resulting garbage. It's that simple."



Name calling

"If you want to provide more accurate documentation then you better come up with something which is helpful instead of a completely useless blurb like the below..."



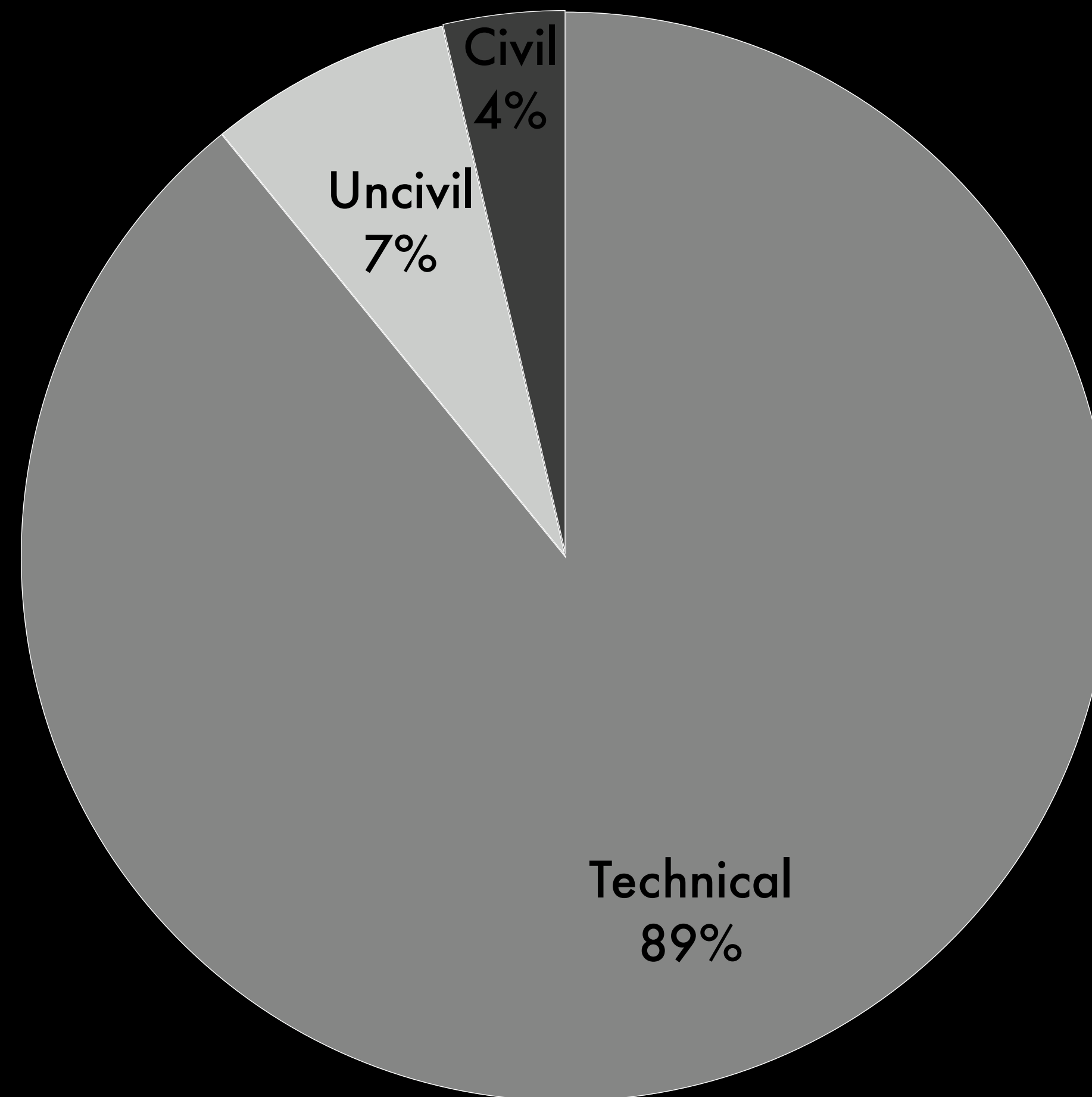
Impatience

"I've looked at your patch for way too long now and still don't see how you've shown it to be correct."

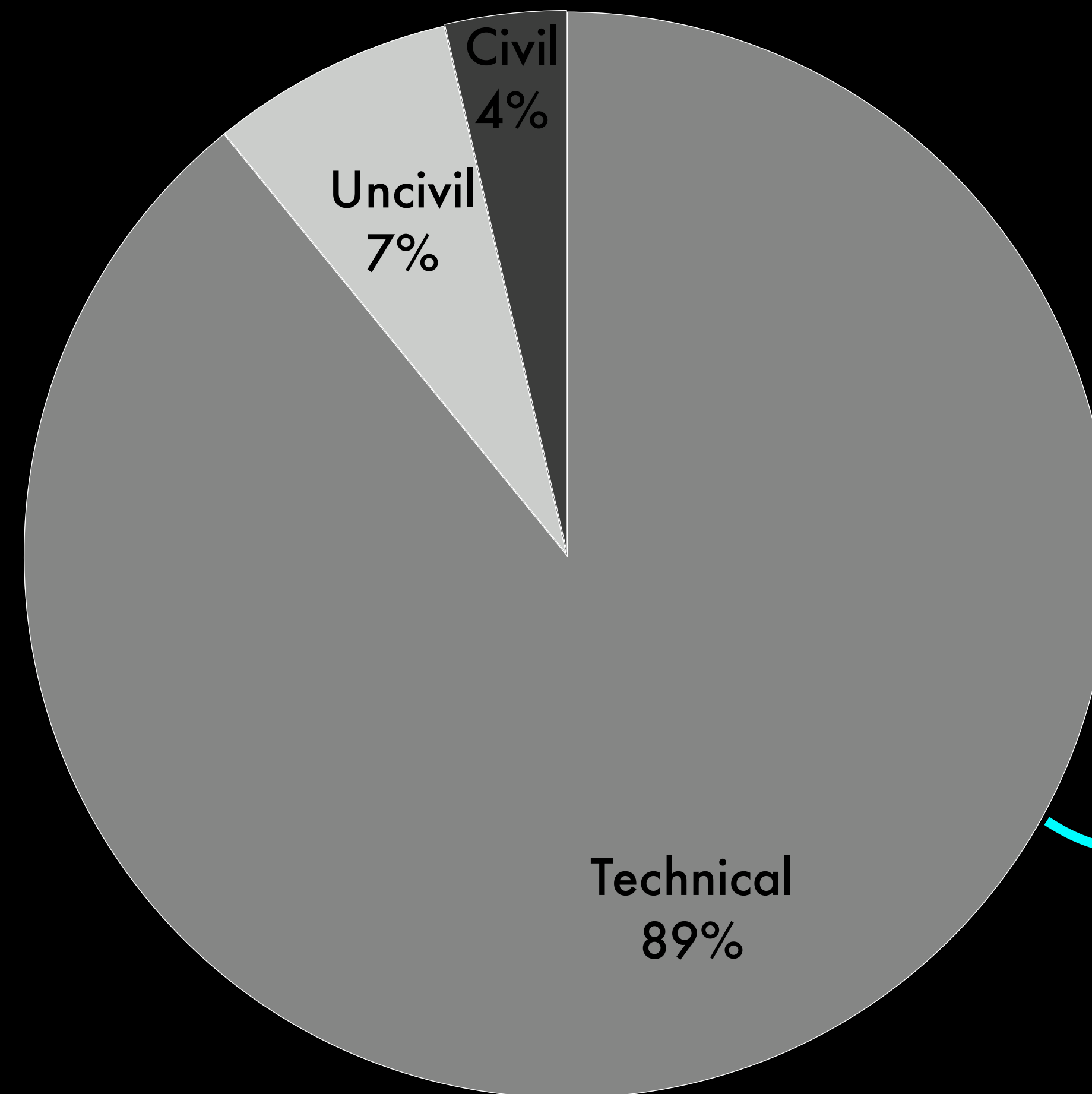


2. How much incivility exists in code review discussions of rejected patches?

As expected, most of the code review emails are technical



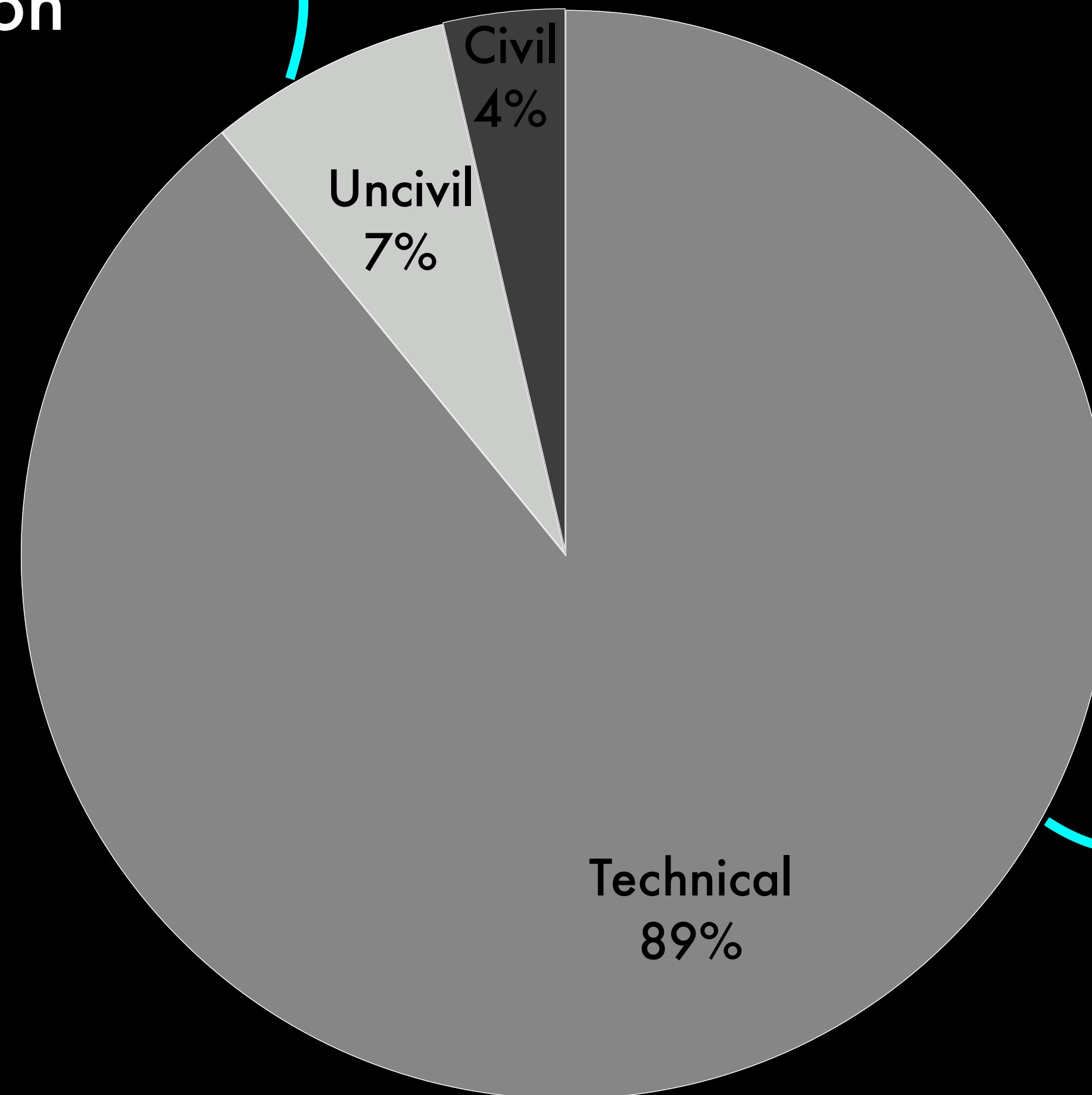
As expected, most of the code review emails are technical



Technical emails contain straightforward technical discussion and do not contain any feature of discussion

As expected, most of the code review emails are technical

Uncivil emails are those that express at least one uncivil feature of discussion

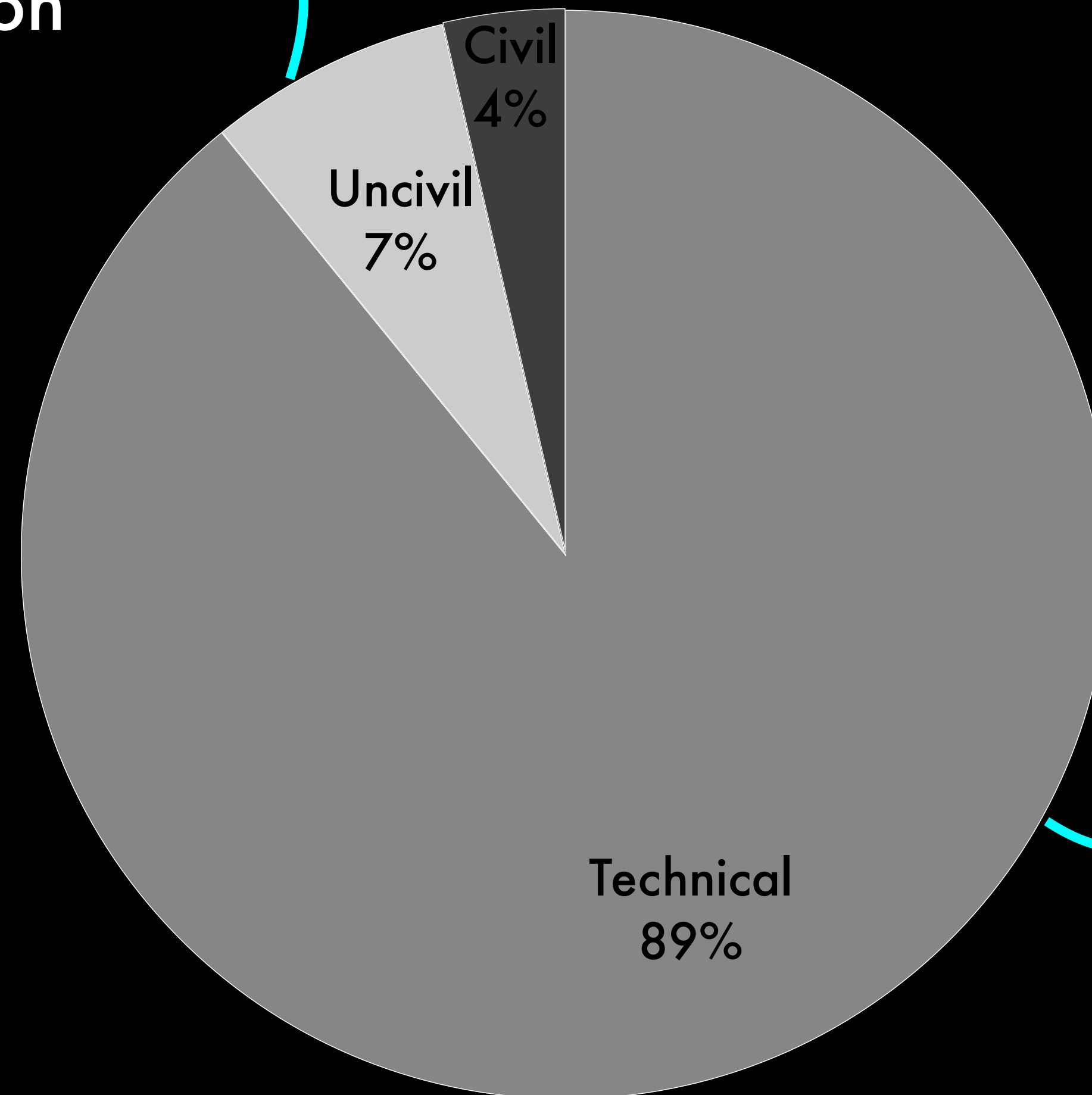


Technical emails contain straightforward technical discussion and do not contain any feature of discussion

As expected, most of the code review emails are technical

Uncivil emails are those that express at least one uncivil feature of discussion

Civil emails are those that express positive, neutral and/or negative features of discussion.

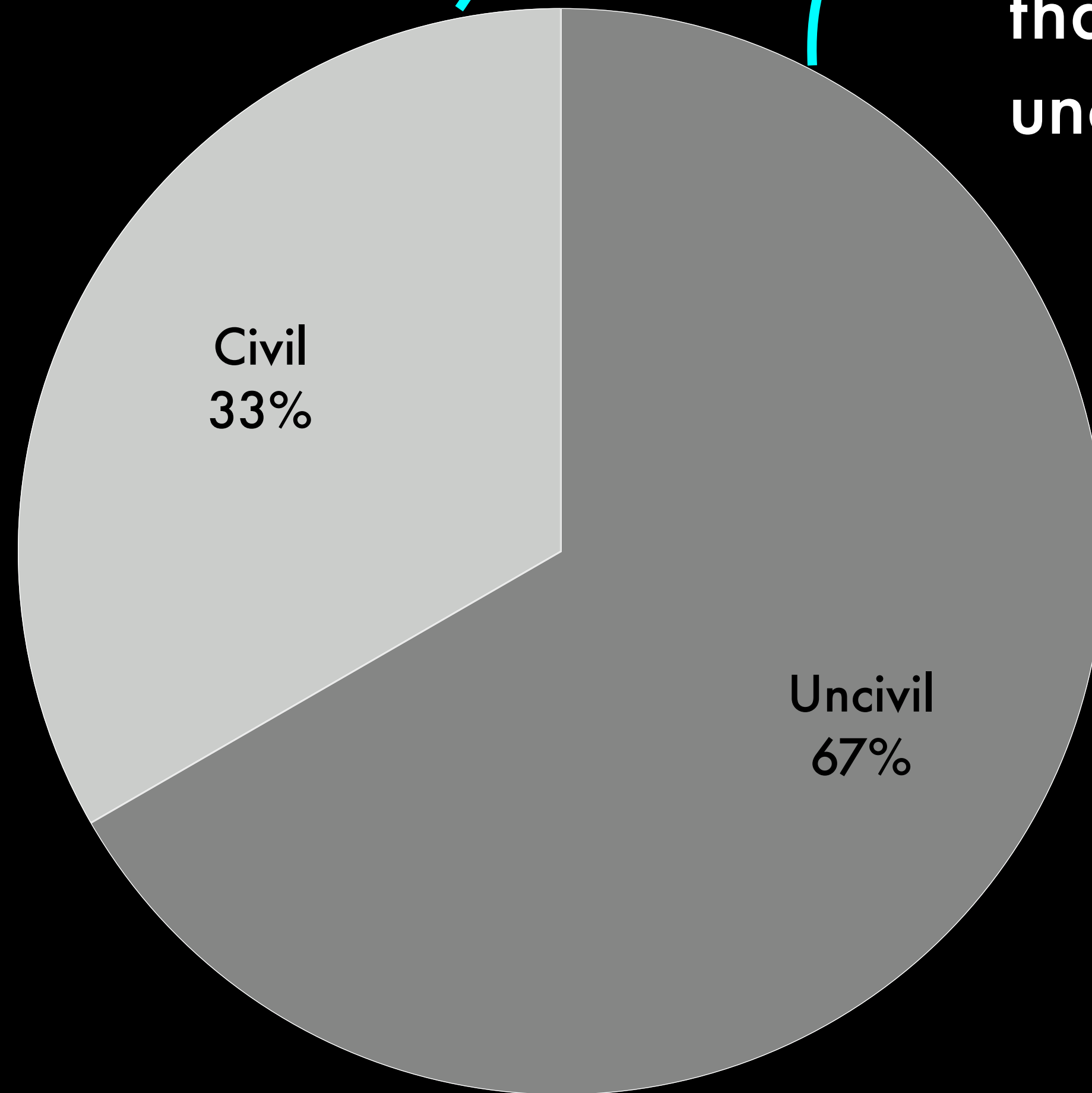


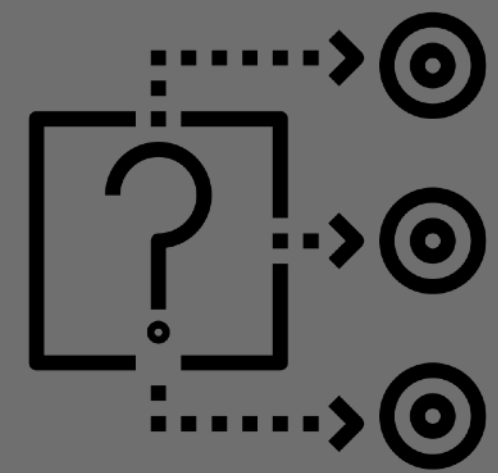
Technical emails contain straightforward technical discussion and do not contain any feature of discussion

2/3 of the non-technical emails are uncivil

Civil emails are those that express positive, neutral and/or negative features of discussion.

Uncivil emails are those that express at least one uncivil feature of discussion





3. What are the causes and consequences of incivility?



Frequent causes and consequences in developers' uncivil emails



Frequent causes and consequences in developers' uncivil emails

Most frequent causes

Most frequent consequences



Frequent causes and consequences in developers' uncivil emails

Most frequent causes	Most frequent consequences
Maintainer's feedback	Escalate uncivil conversation Discontinue further conversation Discuss in a civil way



Frequent causes and consequences in developers' uncivil emails

Most frequent causes	Most frequent consequences
Maintainer's feedback	Escalate uncivil conversation Discontinue further conversation Discuss in a civil way
Violation of community conventions	Discontinue further conversation



Frequent causes and consequences in developers' uncivil emails

Most frequent causes	Most frequent consequences
Maintainer's feedback	Escalate uncivil conversation Discontinue further conversation Discuss in a civil way
Violation of community conventions	Discontinue further conversation
Communication issues	Escalate uncivil conversation



Frequent causes and consequences in developers' uncivil emails

Most frequent causes	Most frequent consequences
Maintainer's feedback	Escalate uncivil conversation Discontinue further conversation Discuss in a civil way
Violation of community conventions	Discontinue further conversation
Communication issues	Escalate uncivil conversation
Maintainer's behavior	Reinforce standpoint Escalate uncivil conversation



Frequent causes and consequences in developers' uncivil emails

Most frequent causes	Most frequent consequences
Maintainer's feedback	Escalate uncivil conversation Discontinue further conversation Discuss in a civil way
Violation of community conventions	Discontinue further conversation
Communication issues	Escalate uncivil conversation
Maintainer's behavior	Reinforce standpoint Escalate uncivil conversation
Inappropriate solution	Provide a technical explanation Escalate uncivil conversation



Frequent causes and consequences in maintainers' uncivil emails



Frequent causes and consequences in maintainers' uncivil emails

Most frequent causes

Most frequent consequences



Frequent causes and consequences in maintainers' uncivil emails

Most frequent causes	Most frequent consequences
Inappropriate solution	Accepted the mantainer's criticism Discontinue further conversation Discuss in a civil way



Frequent causes and consequences in maintainers' uncivil emails

Most frequent causes	Most frequent consequences
Inappropriate solution	Accepted the maintainer's criticism Discontinue further conversation Discuss in a civil way
Poor code quality	Discuss in a civil way Accept maintainer's criticism



Frequent causes and consequences in maintainers' uncivil emails

Most frequent causes	Most frequent consequences
Inappropriate solution	Accepted the maintainer's criticism Discontinue further conversation Discuss in a civil way
Poor code quality	Discuss in a civil way Accept maintainer's criticism
Violation of community's conventions	Discontinue further conversation Accept maintainer's criticism



Frequent causes and consequences in maintainers' uncivil emails

Most frequent causes	Most frequent consequences
Inappropriate solution	Accepted the maintainer's criticism Discontinue further conversation Discuss in a civil way
Poor code quality	Discuss in a civil way Accept maintainer's criticism
Violation of community's conventions	Discontinue further conversation Accept maintainer's criticism
Communication issues	Accept maintainer's criticism Escalate uncivil conversation



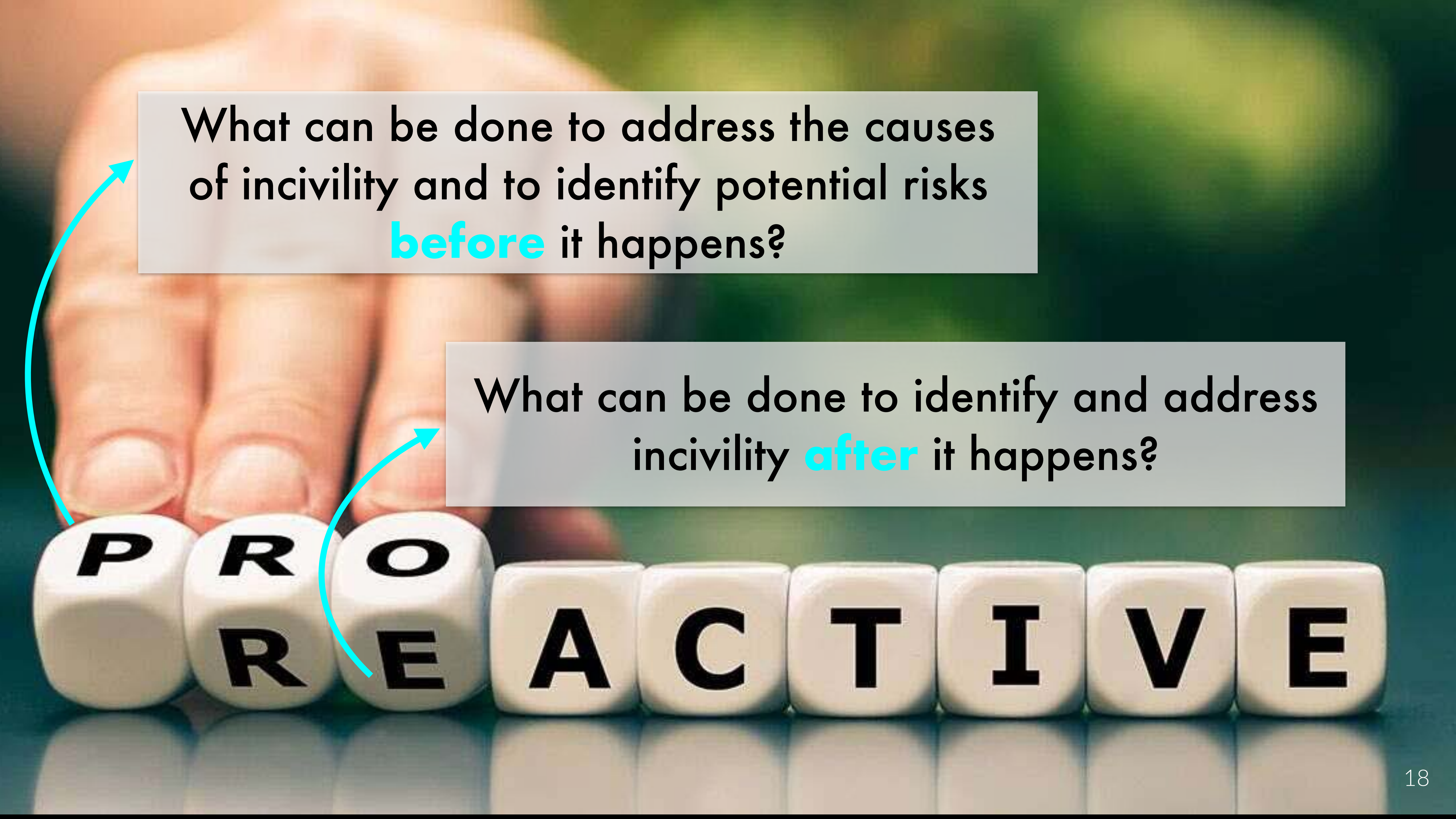
4. How can we create healthier working environments in OSS development?

A close-up photograph of a human hand, palm facing down, hovering just above a row of white letter tiles. The tiles are arranged to spell out the word 'PROACTIVE' in a bold, black, sans-serif font. The tiles are placed on a dark, reflective surface, creating a subtle reflection. The background is a soft, out-of-focus green, suggesting an outdoor setting with foliage. The lighting is warm and focused on the hand and the tiles.

P R O A C T I V E

What can be done to address the causes
of incivility and to identify potential risks
before it happens?

P R O A C T I V E



What can be done to address the causes
of incivility and to identify potential risks
before it happens?

What can be done to identify and address
incivility **after** it happens?

P R O A C T I V E

Proactive suggestions

Proactive suggestions

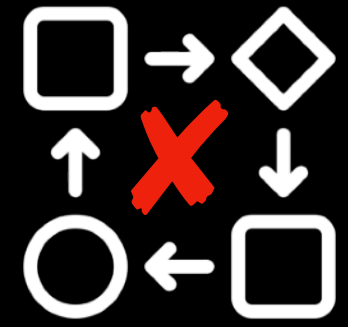


Proactive suggestions



Include a training for newcomers and developers to ensure that everyone is aware about the community conventions.

Proactive suggestions

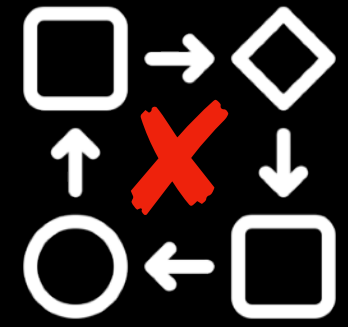


Violation of community conventions

Include a training for newcomers and developers to ensure that everyone is aware about the community conventions.

Maintainers should always include why the patch was rejected.

Proactive suggestions



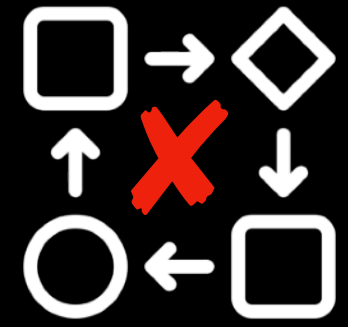
Violation of community conventions

Include a training for newcomers and developers to ensure that everyone is aware about the community conventions.

Gamify the review process so that developers that follow the community conventions gain more reputation and status.

Maintainers should always include why the patch was rejected.

Proactive suggestions



Violation of community conventions

Include a training for newcomers and developers to ensure that everyone is aware about the community conventions.

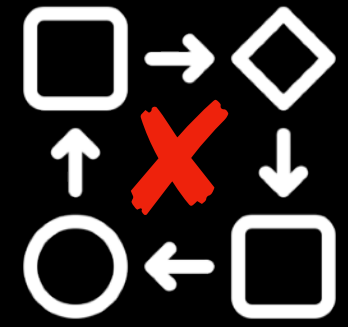
Gamify the review process so that developers that follow the community conventions gain more reputation and status.

Maintainers should always include why the patch was rejected.



Communication issues

Proactive suggestions



Violation of community conventions

Include a training for newcomers and developers to ensure that everyone is aware about the community conventions.

Gamify the review process so that developers that follow the community conventions gain more reputation and status.

Maintainers should always include why the patch was rejected.



Communication issues

Develop a code of conduct focused on the code review process by providing guidelines on how to communicate constructive feedback (maintainers' side) and how to interpret the feedback (developers' side).

Proactive suggestions


Proactive suggestions



Proactive suggestions



Maintainer's feedback



Include a training for maintainers on how to give constructive feedback.

Proactive suggestions



Maintainer's feedback

Include a training for maintainers on how to give constructive feedback.

Include a training for developers on how to handle rejections. Rejections are not a failure!

Proactive suggestions



Maintainer's feedback

Include a training for maintainers on how to give constructive feedback.

Include a training for developers on how to handle rejections. Rejections are not a failure!

Make coaching or mentoring sessions available for maintainers.

Proactive suggestions

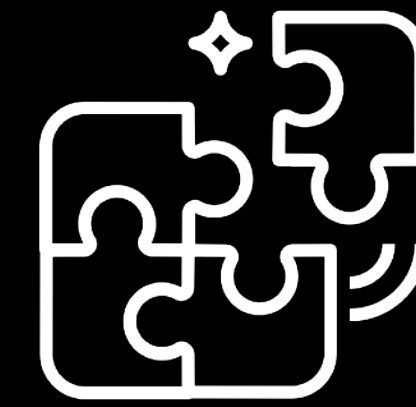


Maintainer's feedback

Include a training for maintainers on how to give constructive feedback.

Include a training for developers on how to handle rejections. Rejections are not a failure!

Make coaching or mentoring sessions available for maintainers.



Inappropriate solution

Proactive suggestions

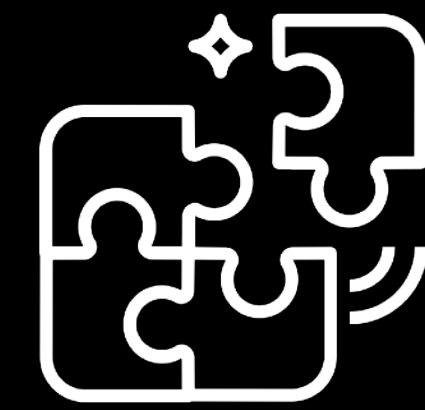


Maintainer's feedback

Include a training for maintainers on how to give constructive feedback.

Include a training for developers on how to handle rejections. Rejections are not a failure!

Make coaching or mentoring sessions available for maintainers.



Inappropriate solution

Developers should always include a technical rationale of their solution, including the negative side effects of the solution (if there are any), the motivation of the proposed patch, and the limitations.

Proactive suggestions

Proactive suggestions




Poor code quality



Proactive suggestions



Poor code quality



Include a training for newcomers and developers to ensure that everyone is aware of the community's expectations in terms of code quality.

Proactive suggestions



Poor code quality



Include a training for newcomers and developers to ensure that everyone is aware of the community's expectations in terms of code quality.

Adopt existing code analysis tools, integrating them into the developers' workflow.

Proactive suggestions



Poor code quality



Include a training for newcomers and developers to ensure that everyone is aware of the community's expectations in terms of code quality.

Adopt existing code analysis tools, integrating them into the developers' workflow.

In addition to addressing the causes, tools could be used...

Proactive suggestions



Poor code quality



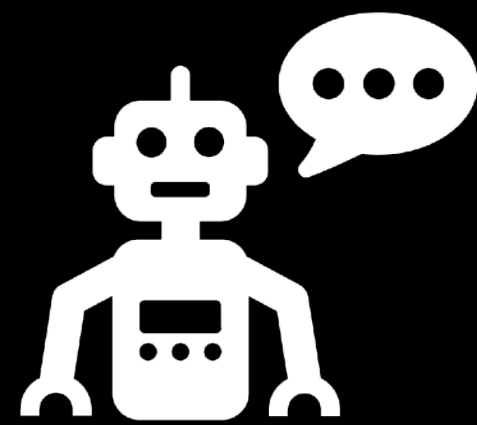
Include a training for newcomers and developers to ensure that everyone is aware of the community's expectations in terms of code quality.

Adopt existing code analysis tools, integrating them into the developers' workflow.

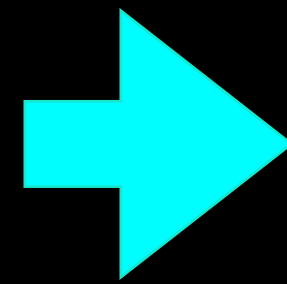
In addition to addressing the causes, tools could be used...

Contributors could use tools to check if their emails are uncivil before they are sent to the mailing list. A more fine-grained tool could let contributors know the type of incivility (e.g bitter frustration, name calling etc).

Reactive suggestions



Use bots that are constantly checking if the emails sent to the mailing list are civil or uncivil.



Community leaders can be warned to assess the situation and take the appropriate measures.

What about automated **incivility** detection?



Incivility cannot be reliably captured just by analyzing the sentiment of a text

Incivility cannot be reliably captured just by analyzing the sentiment of a text

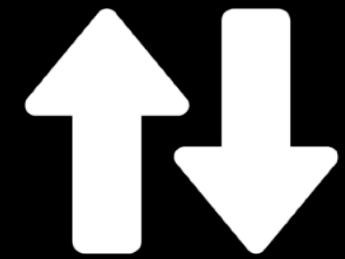


We tested three SE-specific sentiment analysis tools: **Senti4SD**, **SentiStrength-SE**, **SentiCR**.

Incivility cannot be reliably captured just by analyzing the sentiment of a text



We tested three SE-specific sentiment analysis tools: Senti4SD, SentiStrength-SE, SentiCR.



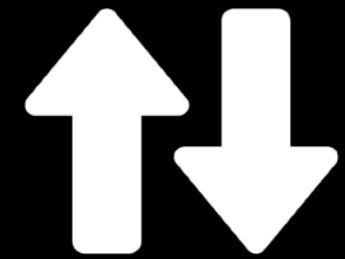
The three specific SE-tools tend to have high precision and low recall. The overall performance tend to be very low.

E.g.: The tools detect the negative sentiment with ~73% of precision, but it would miss up to 91% of the cases.

Incivility cannot be reliably captured just by analyzing the sentiment of a text



We tested three SE-specific sentiment analysis tools: Senti4SD, SentiStrength-SE, SentiCR.



The three specific SE-tools tend to have high precision and low recall. The overall performance tend to be very low.

E.g.: The tools detect the negative sentiment with ~73% of precision, but it would miss up to 91% of the cases.



Incivility has many **dimensions not captured by tools**: the context of the conversation, the familiarity among people, the granularity of analysis. Some discussion features may be hard to be detected: irony, mocking, threat.

Building incivility-specific detectors

Building incivility-specific detectors

- ✓ Heuristics can be developed to identify each discussion feature.

Building incivility-specific detectors

- ✓ Heuristics can be developed to identify each discussion feature.
- ✓ Heuristics need to consider the context of the conversation, the familiarity among people, and the granularity of analysis.

Building incivility-specific detectors

- ✓ Heuristics can be developed to identify each discussion feature.
- ✓ Heuristics need to consider the context of the conversation, the familiarity among people, and the granularity of analysis.
- ✓ An incivility specific lexicon for each discussion feature can be built to improve the performance of classifiers.

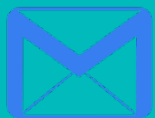
Building incivility-specific detectors

- ✓ Heuristics can be developed to identify each discussion feature.
- ✓ Heuristics need to consider the context of the conversation, the familiarity among people, and the granularity of analysis.
- ✓ An incivility specific lexicon for each discussion feature can be built to improve the performance of classifiers.
- ✓ Our dataset can be extended to train machine learning models to detect incivility.



Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD
candidate seeking for
a SE research
opportunity in the
industry.





Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD candidate seeking for a SE research opportunity in the industry.

Frustration, name calling and impatience are the most frequent uncivil features of discussion



Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD candidate seeking for a SE research opportunity in the industry.

Frustration, name calling and impatience are the most frequent uncivil features of discussion

2/3 of the non-technical discussions are uncivil



Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD candidate seeking for a SE research opportunity in the industry.

Frustration, name calling and impatience are the most frequent uncivil features of discussion

2/3 of the non-technical discussions are uncivil

We identified various causes and consequences of uncivil emails sent by developers and maintainers



Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD candidate seeking for a SE research opportunity in the industry.

Frustration, name calling and impatience are the most frequent uncivil features of discussion

2/3 of the non-technical discussions are uncivil

We identified various causes and consequences of uncivil emails sent by developers and maintainers

We suggest many proactive and reactive approaches, and ideas on how to build incivility-specific detectors.



Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD candidate seeking for a SE research opportunity in the industry.

Frustration, name calling and impatience are the most frequent uncivil features of discussion

2/3 of the non-technical discussions are uncivil

We identified various causes and consequences of uncivil emails sent by developers and maintainers

We suggest many proactive and reactive approaches, and ideas on how to build incivility-specific detectors.

Download our paper for more details:

<https://arxiv.org/pdf/2108.09905.pdf>



SCAN ME



Isabella Ferreira

Polytechnique Montréal



isabella.ferreira@polymtl.ca



/isabellaferreira



/isabellavieira



@isaferreira_57

HIRE ME!

I'm a 4th year PhD candidate seeking for a SE research opportunity in the industry.

Frustration, name calling and impatience are the most frequent uncivil features of discussion

2/3 of the non-technical discussions are uncivil

We identified various causes and consequences of uncivil emails sent by developers and maintainers

We suggest many proactive and reactive approaches, and ideas on how to build incivility-specific detectors.

Download our paper for more details:

<https://arxiv.org/pdf/2108.09905.pdf>



SCAN ME

Interview with FLOSS contributors on (in)civility

Have you ever experienced/witnessed disrespectful conversations in FLOSS discussions?

Participate in our interview! <https://bit.ly/3xE2SqV>



SCAN ME

