

Assessing Perceived Sentiment in Pull Requests with Emoji: Evidence from Tools and Developer Eye Movements

Kang-il Park, Bonita Sharif

SOFTWARE ENGINEERING RESEARCH AND EMPIRICAL STUDIES LAB

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

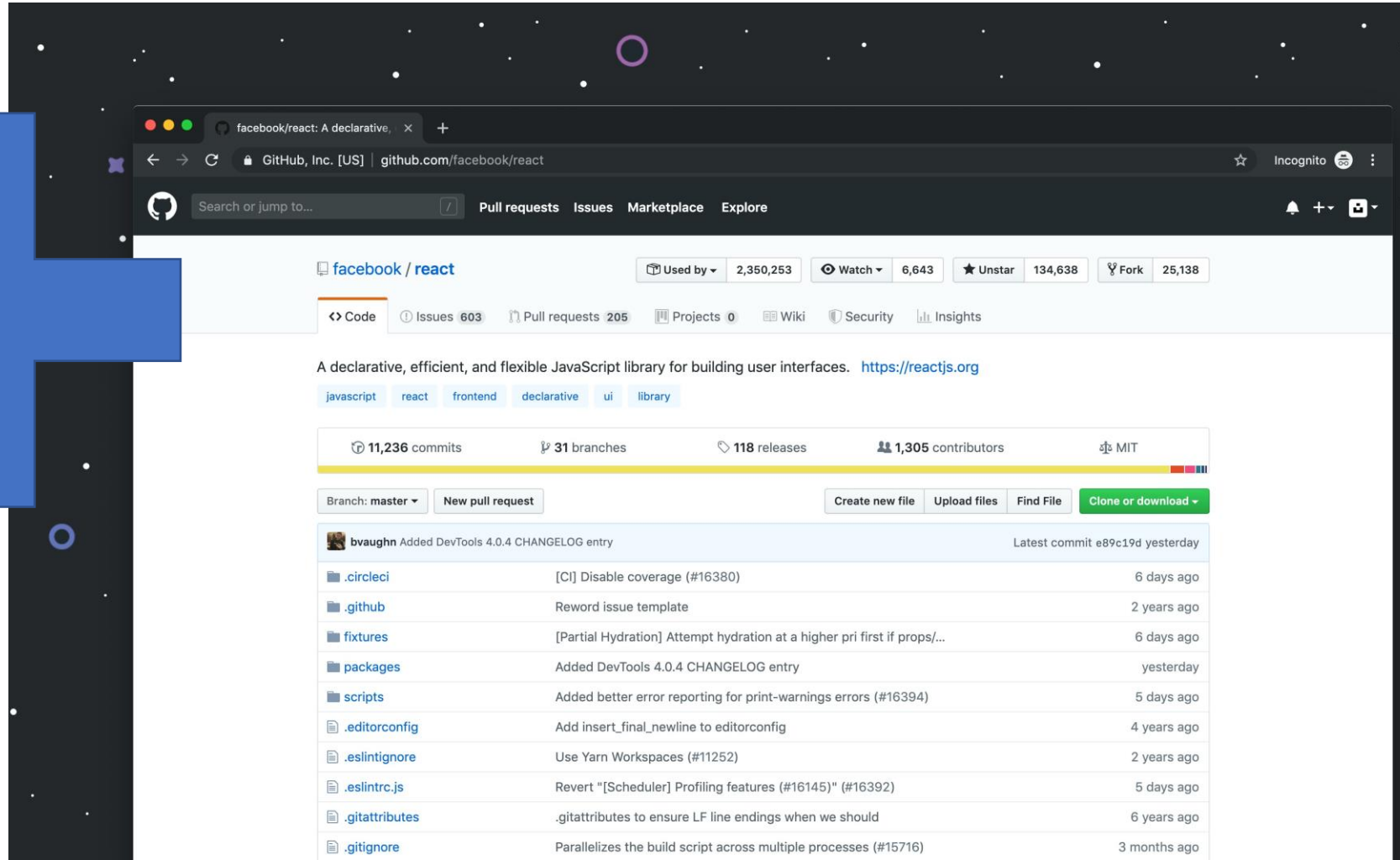
UNIVERSITY OF NEBRASKA – LINCOLN, LINCOLN USA

@KANGIL_PARK, @SHBONITA 



Replication Package





Research Questions

RQ1: How do current state-of-the-art sentiment analysis tools compare against perceived developer sentiment in GitHub PRs with emoji?

RQ2: What elements of the PR page do developers spend most of their attention on when reading PRs with emoji in the context of analyzing sentiment?

I'd say `Another keybinding has precedence` or `there is another keybinding that took precedence`

Yup, much better! 🤖 🚗

Contributor

That's currently our plan. We want to deprecate `workspaceView` and have all access go through the model to eliminate the confusion of which to use. If you absolutely need a view object (which you should only need if you need to do unsafe things to the DOM) then you'll call `workspace.getView(model)` to get the view corresponding to the model.

Member

I bloody ❤️ that!

Contributor

Maybe something like `sourceBlockSpace` just to differentiate from generated source code / DOM src attribute? Could make it easier to rename/search for object property later

Author Contributor

I had `srcBlockSpace` originally, then changed it for brevity 😊 Good call on searchability, I'll change it back.

blockly-core/core/block_space_editor.js

```
604 - Blockly.clipboard_ = xmlBlock;  
603 + Blockly.clipboard_ = {  
604 +   dom: xmlBlock,  
605 +   src: block.blockSpace
```

Contributor

Maybe something like `sourceBlockSpace` just to differentiate from generated source code / DOM src attribute? Could make it easier to rename/search for object property later

Author Contributor

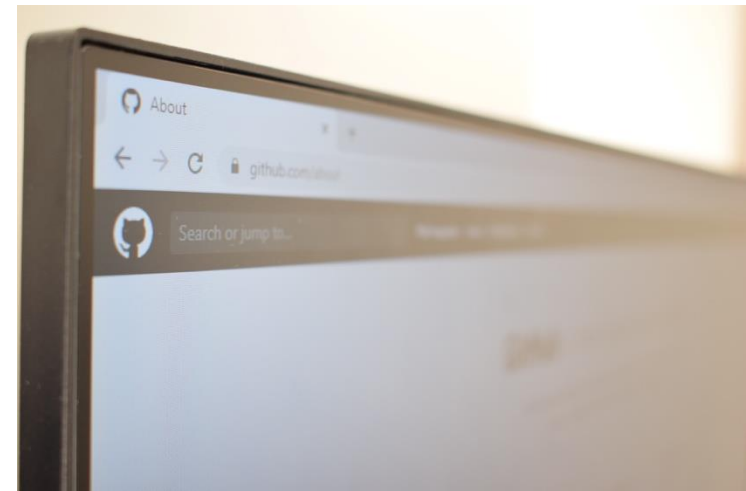
I had `srcBlockSpace` originally, then changed it for brevity 😊 Good call on searchability, I'll change it back.

Author Contributor

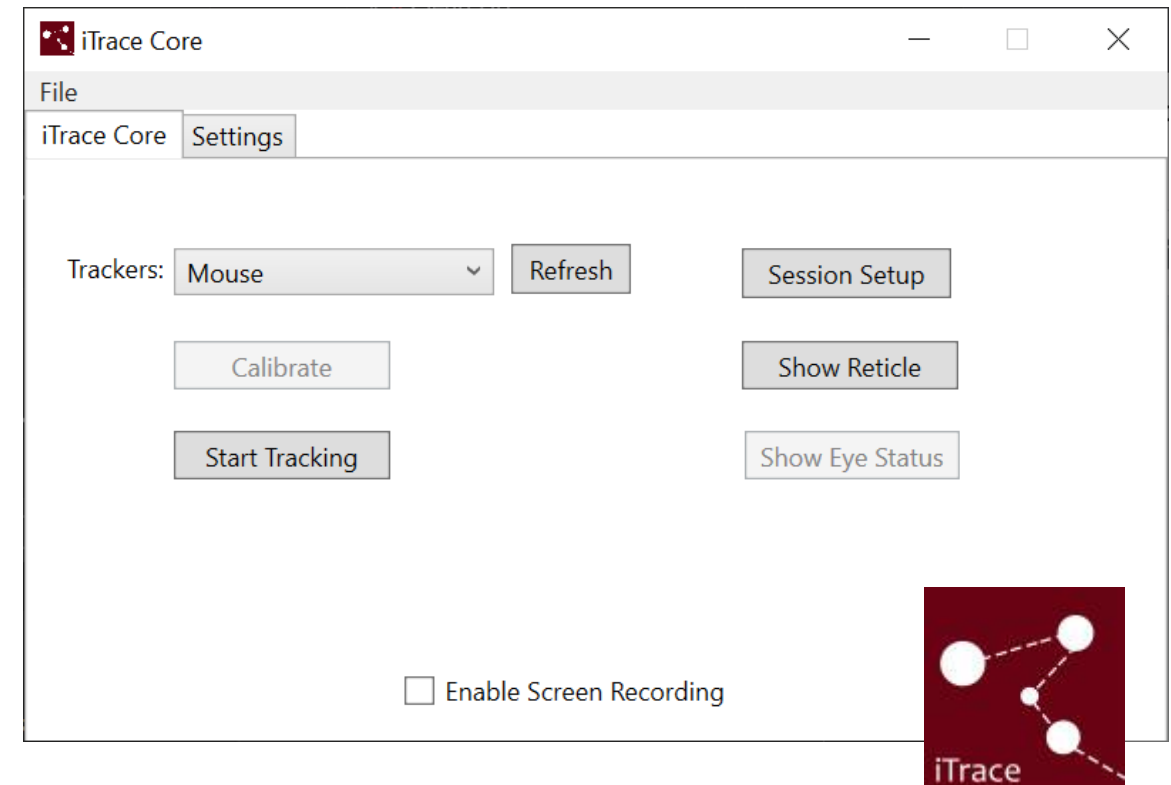
Fixed in [91b09e9](#)

Our Contributions

- An eye tracking study conducted on GitHub PR pages where the developer can scroll and read through comments
- First realistic eye tracking study conducted on GitHub PR pages in the context of assessing perceived sentiment



Pilot Study Design



Add debugging guide by izuzak · x

github.com/atom/atom/pull/2502/

Why GitHub?TeamEnterpriseExploreMarketplacePricing

atom / atom

Watch2.6kStar54kFork15.6k

CodeIssues588Pull requests159ActionsProjectsSecurityInsights

Add debugging guide #2502

Mergedmerged 2 commits into master from iz-debugging-atom-docs on Jun 5, 2014

Conversation13Commits2Checks0Files changed1

commented on Jun 4, 2014

This is the first part of a guide on debugging problems and errors in Atom. I'll add more stuff to this PR (e.g. starting Atom in safe mode), but there's enough text already in here to get some feedback.

first part of debugging guide

Reviewers

No reviews

Assignees

No one assigned


Labels

Click Button Below to Connect To ServerStart
Session Started - Connected
Write XML Data

Sign up

+100-0

✓ f5bd6a6



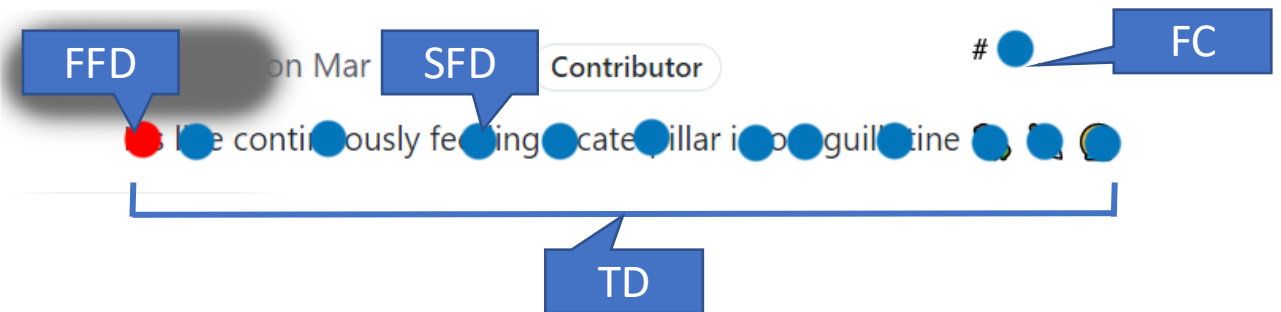
SEmotion'21 ICSE Workshop

May 31, 2021

6

Dependent Variables

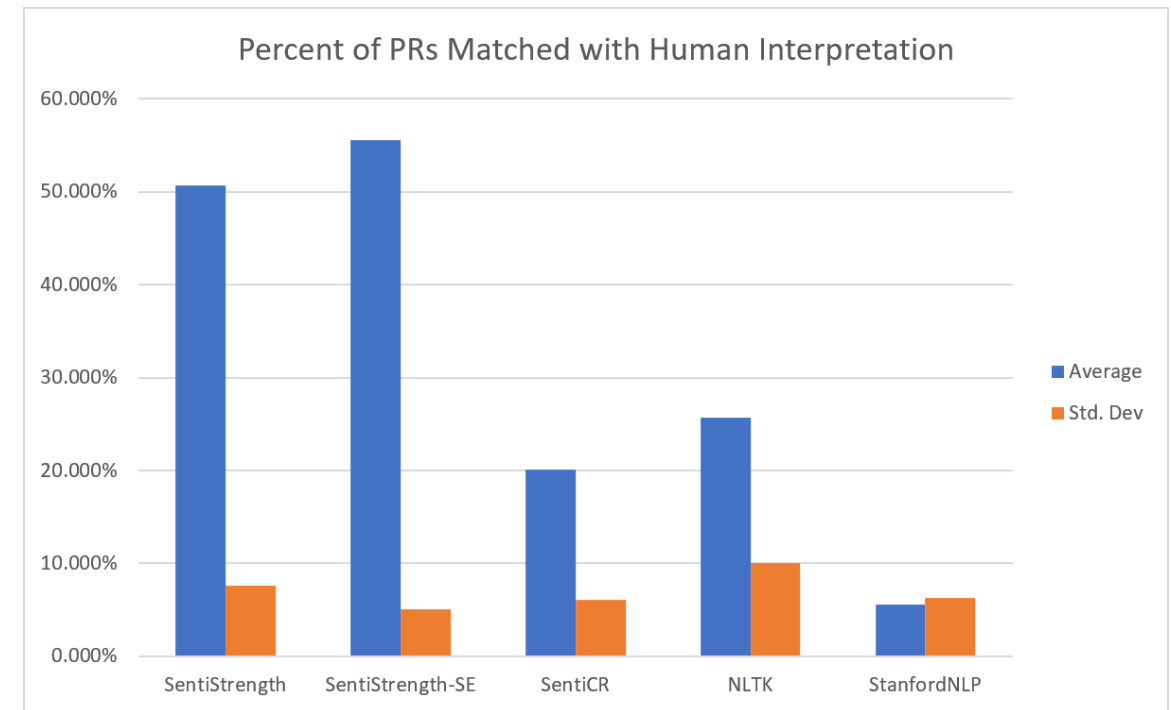
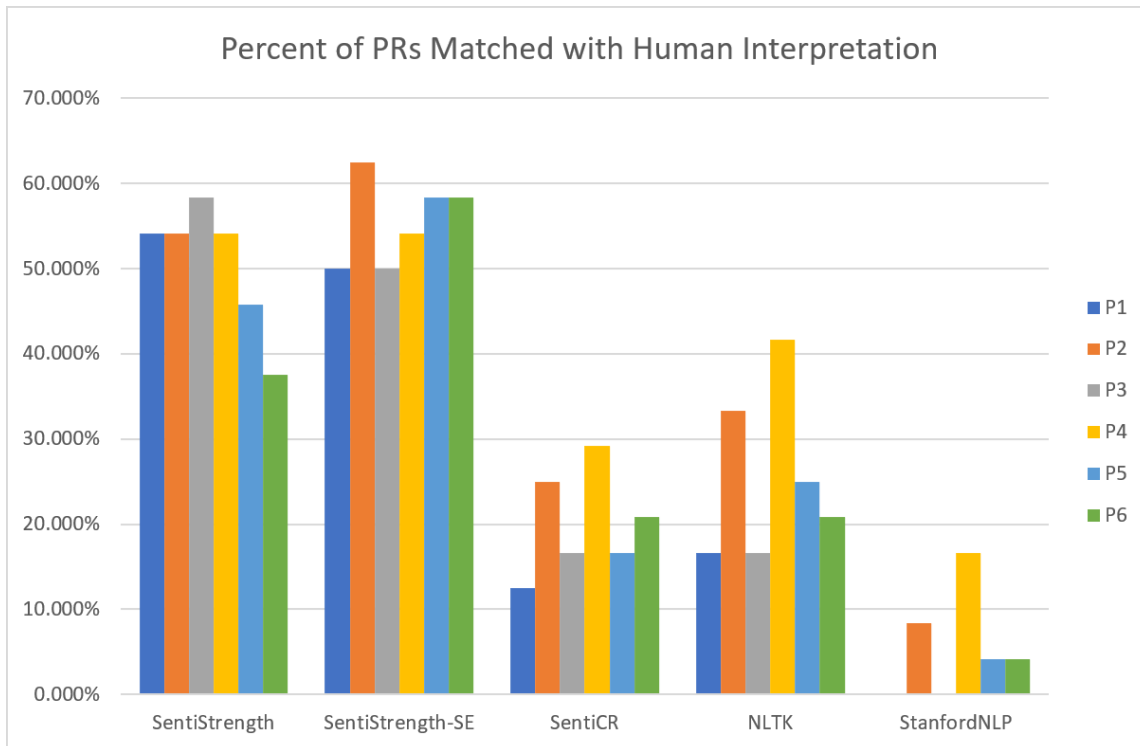
- Developer perceived sentiment
- Developer agreement on tool output
- First Fixation Duration (FFD) *
- Single Fixation Duration (SFD) *
- Total Duration (TD) *
- Fixation Count (FC) *
- Adjusted Fixation Duration (AFD)



$$adj. \text{ duration} = \frac{\text{duration} * \text{total characters}}{\text{character count of AOI type}}$$

* Adapted from C. M. Robus, C. J. Hand, R. Filik, and M. Pitchford, "Investigating effects of emoji on neutral narrative text: Evidence from eye movements and perceived emotional valence," *Computers in Human Behavior*, p.106361, 2020.

Results – How do current state-of-the-art sentiment analysis tools compare against perceived developer sentiment in GitHub PRs with emoji?



Results – Tool Oddities

PR#	SentiStr	SentiStr-SE	SentiCR	NLTK	Stanford NLP
97	67% Neu	33% Pos	0% Neg	67% Neu	0% Neg
209	83% Pos	83% Pos	0% Neg	17% Neu	0% Neg
299	100% Pos	0% Neu	0% Neg	0% Neg	0% Neg
346	17% Pos	17% Neg	17% Neg	17% Neg	17% Neg
364	50% Pos	50% Pos	0% Neg	0% Neg	0% Neg
408	33% Pos	67% Neu	0% Neg	33% Pos	0% Neg
415	83% Pos	83% Pos	17% Neu	17% Neu	0% Neg
441	33% Pos	33% Pos	0% Neg	67% Neu	0% Neg
685	0% Neg	100% Neu	100% Neu	0% Neg	0% Neg
1683	33% Pos	67% Neu	0% Neg	0% Neg	0% Neg
1776	33% Neg	33% Neg	33% Neg	33% Neg	33% Neg
1794	67% Neu	67% Neu	67% Neu	67% Neu	0% Neg
2291	0% Neg	0% Neg	0% Neg	0% Neg	0% Neg
2426	17% Pos	67% Neu	17% Neg	17% Neg	17% Neg
2457	50% Neu	50% Neu	17% Neg	17% Neg	17% Neg
2502	100% Pos	0% Neu	0% Neg	100% Pos	0% Neg
2603	83% Pos	83% Pos	17% Neu	0% Neg	0% Neg
3142	33% Pos	67% Neu	0% Neg	0% Neg	0% Neg
3241	0% Neg	100% Neu	100% Neu	0% Neg	0% Neg
3394	50% Pos	50% Pos	50% Neu	0% Neg	0% Neg
3433	83% Pos	83% Pos	0% Neg	0% Neg	0% Neg
3633	83% Pos	83% Pos	0% Neg	17% Neu	0% Neg
3697	17% Pos	17% Pos	50% Neg	50% Neg	50% Neg
4973	100% Pos	100% Pos	0% Neu	100% Pos	0% Neu
Average	50.69%	55.56%	20.14%	25.69%	5.56%
Std. Dev	7.65%	5.05%	6.13%	10.01%	6.27%

on May 4, 2015 Contributor

this is fixing an existing bug? maybe add a test to avoid regressions?
also, why is try necessary for project_template_level but not game (at first glance it seems to me that it's not necessary for either)?

on May 5, 2015 Author Collaborator

Yes, fixing an existing bug. Good call, I'll add a test.

I'm not sure why, but game exists on all levels whereas project_template_level does not:

```

irb(main):001:0> Level.all.all?(&:game)
=> true

irb(main):002:0> Level.all.all?(&:project_level_template)
NoMethodError: undefined method `project_level_template' for #<Eval:0x007fbffd380508>

```

on May 5, 2015 Contributor

project_template_level not project_level_template? The method is defined on Level

code-dot-org/dashboard/app/models/level.rb
Line 190 in 10f2e29

```

190 def project_template_level

```

so I don't see why it would say undefined method (it could certainly be a typo)

on May 5, 2015 Author Collaborator

Ah you're totally right. If only I could type 🤔. Disregard.

on May 5, 2015 Author Collaborator

Fixed and added rake test in [bbe93ce](#).

“Ah you’re totally right.
If only I could type. 🤔”

Results – More Tool Oddities

PR#	SentiStr	SentiStr-SE	SentiCR	NLTK	Stanford NLP
97	67% Neu	33% Pos	0% Neg	67% Neu	0% Neg
209	83% Pos	83% Pos	0% Neg	17% Neu	0% Neg
299	100% Pos	0% Neu	0% Neg	0% Neg	0% Neg
346	17% Pos	17% Neg	17% Neg	17% Neg	17% Neg
364	50% Pos	50% Pos	0% Neg	0% Neg	0% Neg
408	33% Pos	67% Neu	0% Neg	33% Pos	0% Neg
415	83% Pos	83% Pos	17% Neu	17% Neu	0% Neg
441	33% Pos	33% Pos	0% Neg	67% Neu	0% Neg
685	0% Neg	100% Neu	100% Neu	0% Neg	0% Neg
1683	33% Pos	67% Neu	0% Neg	0% Neg	0% Neg
1776	33% Neg	33% Neg	33% Neg	33% Neg	33% Neg
1794	67% Neu	67% Neu	67% Neu	67% Neu	0% Neg
2291	0% Neg	0% Neg	0% Neg	0% Neg	0% Neg
2426	17% Pos	67% Neu	17% Neg	17% Neg	17% Neg
2457	50% Neu	50% Neu	17% Neg	17% Neg	17% Neg
2502	100% Pos	0% Neu	0% Neg	100% Pos	0% Neg
2603	83% Pos	83% Pos	17% Neu	0% Neg	0% Neg
3142	33% Pos	67% Neu	0% Neg	0% Neg	0% Neg
3241	0% Neg	100% Neu	100% Neu	0% Neg	0% Neg
3394	50% Pos	50% Pos	50% Neu	0% Neg	0% Neg
3433	83% Pos	83% Pos	0% Neg	0% Neg	0% Neg
3633	83% Pos	83% Pos	0% Neg	17% Neu	0% Neg
3697	17% Pos	17% Pos	50% Neg	50% Neg	50% Neg
4973	100% Pos	100% Pos	0% Neu	100% Pos	0% Neu
Average	50.69%	55.56%	20.14%	25.69%	5.56%
Std. Dev	7.65%	5.05%	6.13%	10.01%	6.27%

docs/debugging.md

```

23 +
24 + In some cases, unexpected behavior might be caused by misconfigured or unconfigured settings in Atom
   or in one of the packages.
25 +
26 + Open Atom's Settings View with cmd-` or the Atom > Preferences menu option.

```

Member

The default key is Command+Comma `cmd-,` isn't it?

Also, should we be using `<kbd>` instead of `<code>` for keyboard commands?

to `<kbd>`

to `<kbd>`

Author Member

The default key is Command+Comma `cmd-,` isn't it?

It is -- good catch, thanks! (This will allow me to drop the `<code>` tags and just use backticks 😊)

Also, should we be using `<kbd>` instead of `<code>` for keyboard commands?

I'd be `com` with that, but if we do decide to do that -- we should see this change made in another PR.

Member

Agreed on the `<code>` to `<kbd>` change 😊

“Agreed on the `<code>` to `<kbd>` change 😎”

Results – More Tool Oddities

PR#	SentiStr	SentiStr-SE	SentiCR	NLTK	Stanford NLP
97	67% Neu	33% Pos	0% Neg	67% Neu	0% Neg
209	83% Pos	83% Pos	0% Neg	17% Neu	0% Neg
299	100% Pos	0% Neu	0% Neg	0% Neg	0% Neg
346	17% Pos	17% Neg	17% Neg	17% Neg	17% Neg
364	50% Pos	50% Pos	0% Neg	0% Neg	0% Neg
408	33% Pos	67% Neu	0% Neg	33% Pos	0% Neg
415	83% Pos	83% Pos	17% Neu	17% Neu	0% Neg
441	33% Pos	33% Pos	0% Neg	67% Neu	0% Neg
685	0% Neg	100% Neu	100% Neu	0% Neg	0% Neg
1683	33% Pos	67% Neu	0% Neg	0% Neg	0% Neg
1776	33% Neg	33% Neg	33% Neg	33% Neg	33% Neg
1794	67% Neu	67% Neu	67% Neu	67% Neu	0% Neg
2291	0% Neg	0% Neg	0% Neg	0% Neg	0% Neg
2426	17% Pos	67% Neu	17% Neg	17% Neg	17% Neg
2457	50% Neu	50% Neu	17% Neg	17% Neg	17% Neg
2502	100% Pos	0% Neu	0% Neg	100% Pos	0% Neg
2603	83% Pos	83% Pos	17% Neu	0% Neg	0% Neg
3142	33% Pos	67% Neu	0% Neg	0% Neg	0% Neg
3241	0% Neg	100% Neu	100% Neu	0% Neg	0% Neg
3394	50% Pos	50% Pos	50% Neu	0% Neg	0% Neg
3433	83% Pos	83% Pos	0% Neg	0% Neg	0% Neg
3633	83% Pos	83% Pos	0% Neg	17% Neu	0% Neg
3697	17% Pos	17% Pos	50% Neg	50% Neg	50% Neg
4973	100% Pos	100% Pos	0% Neu	100% Pos	0% Neu
Average	50.69%	55.56%	20.14%	25.69%	5.56%
Std. Dev	7.65%	5.05%	6.13%	10.01%	6.27%

```
src/editor-component.coffee
685 + # it will be merged into another one
686 + # reference to the new selection that
687 + rowSelection = _.find editor.getSelections()
688 + selection.intersectsBufferRange(
```

Author Member

See #3433 (comment) for some notes on this (the comment was hidden by another commit I needed to make to make coffeelint happy). 😊

Contributor

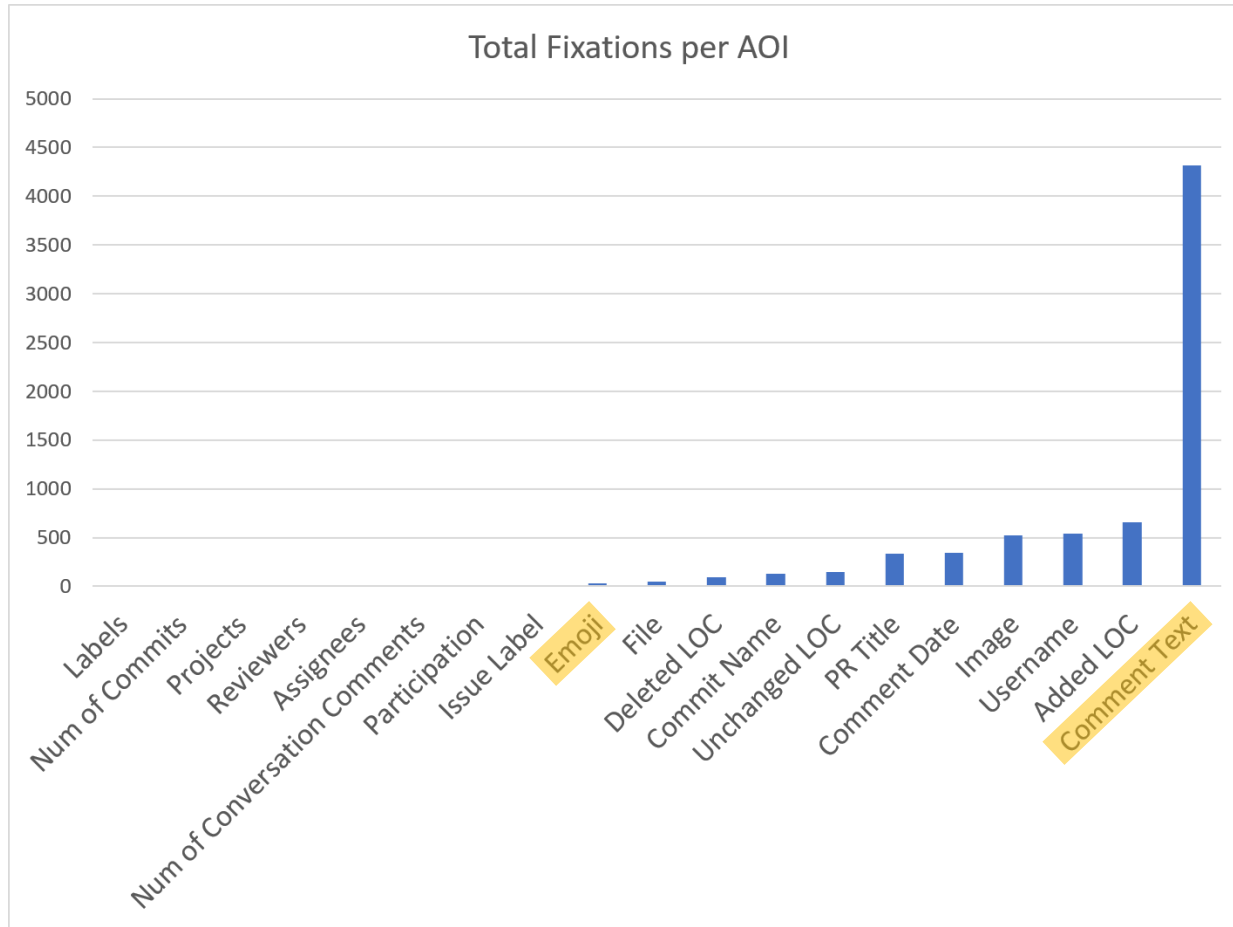
This looks reasonable for now, but I'm wondering if we should revisit the way in which selections are merged after certain API interactions. Maybe adding or updating a selection should always favor the new/modified selection when merging. Or maybe you could call `mergeIntersectingSelections` on the selection itself so your reference remains stable.

Author Member

This looks reasonable for now, but I'm wondering if we should revisit the way in which selections are merged after certain API interactions. Maybe adding or updating a selection should always favor the new/modified selection when merging. Or maybe you could call `mergeIntersectingSelections` on the selection itself so your reference remains stable.

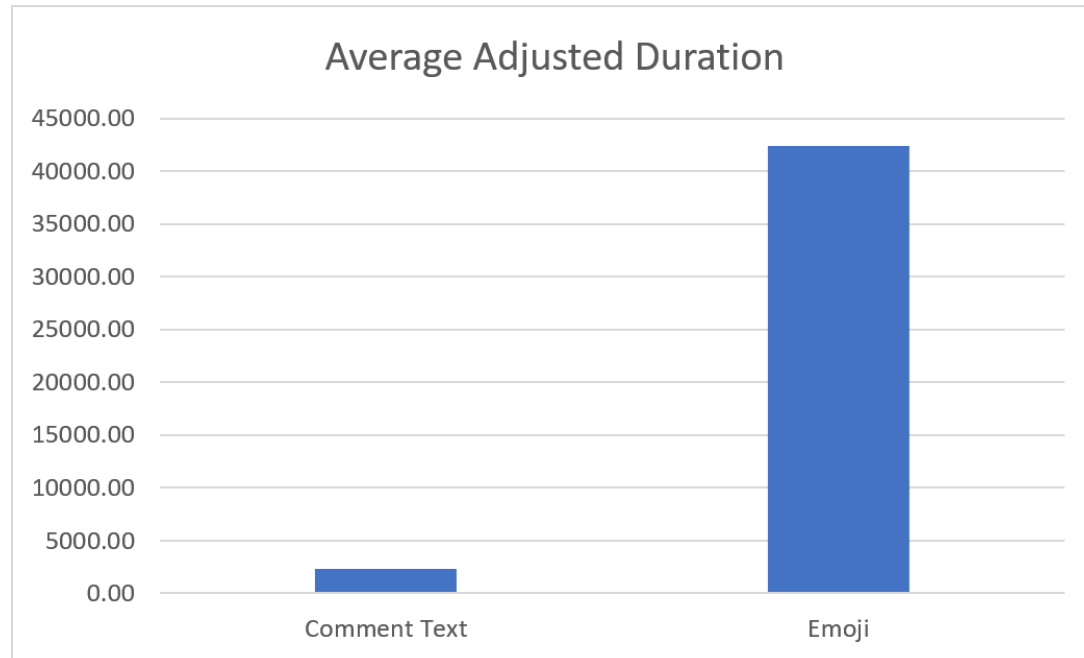
I wanted to respond to this since I have a method missing in the Editor or Selection API which I can use. Since I'm new to these APIs -- I'm not really sure what would work best, but having a `mergeIntersectingSelections` method on Selections themselves sounds like a good idea to me. 👍

Results – What elements of the PR page do developers spend most of their attention on when reading PRs with emoji in the context of analyzing sentiment?



	P1	P2	P3	P4	P5	Avg
Overview						
Emoji	129.50	232.33	289.60	230.14	257.50	227.82
File	71.50	242.42	255.12	166.25	136.07	174.27
Image	130.07	142.58	192.60	170.13	161.91	159.46
Username	117.08	159.15	157.20	231.21	121.30	157.19
Deleted LOC	112.03	149.33	384.87	0.00	136.96	156.64
Unchanged LOC	120.55	149.34	158.48	207.50	131.75	153.52
Commit Name	103.15	213.04	173.42	143.00	123.51	151.22
Comment Text	113.57	170.21	184.57	149.38	133.72	150.29
PR Title	117.83	179.52	237.91	133.00	83.00	150.25
Other	112.02	157.29	183.93	168.65	128.38	150.05
Added LOC	112.99	172.43	171.40	123.21	146.28	145.26
Comment Date	115.67	157.74	156.10	163.57	113.15	141.25
of Conversation Comm	0.00	0.00	83.00	449.00	0.00	106.40
Participation	0.00	149.50	149.00	0.00	83.00	76.30
Issue Label	0.00	0.00	149.00	0.00	82.67	46.33
Labels	0.00	0.00	183.00	0.00	0.00	36.60
Num of Commits	0.00	149.00	0.00	0.00	0.00	29.80
Projects	0.00	149.00	0.00	0.00	0.00	29.80
Assignees	0.00	141.00	0.00	0.00	0.00	28.20
Reviewers	0.00	99.00	0.00	0.00	0.00	19.80

Emoji vs. Comments



adjusted fixation duration

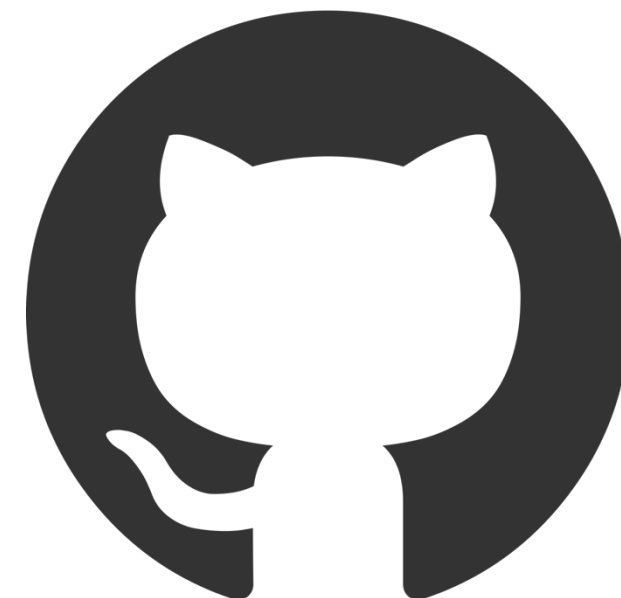
$$= \frac{\text{duration} * \text{total characters}}{\text{character count of AOI type}}$$

	P1	P2	P3	P4	P5	Avg
Overview						
Emoji	53013.28	55101.27	20427.68	19701.49	71780.10	44004.76
Comment Text	1853.23	2923.53	3065.33	904.04	1479.71	2045.17

Final Observations

- The percentage of predictions agreeing with human participants ranges from 55.56% (SentiStrength-SE) to 5.56% (Stanford CoreNLP)
- Because sentiment analysis tools do not take emojis into consideration, tool prediction results are low vs. participants
- Emojis were looked at far longer vs. rest of text even after adjusting for length

Overview	P1	P2	P3	P4	P5	Avg
Emoji	129.50	232.33	289.60	230.14	257.50	227.82
File	71.50	242.42	255.12	166.25	136.07	174.27
Image	130.07	142.58	192.60	170.13	161.91	159.46
Username	117.08	159.15	157.20	231.21	121.30	157.19
Deleted LOC	112.03	149.33	384.87	0.00	136.96	156.64
Unchanged LOC	120.55	149.34	158.48	207.50	131.75	153.52
Commit Name	103.15	213.04	173.42	143.00	123.51	151.22
Comment Text	113.57	170.21	184.57	149.38	133.72	150.29
PR Title	117.83	179.52	237.91	133.00	83.00	150.25
Other	112.02	157.29	183.93	168.65	128.38	150.05
Added LOC	112.99	172.43	171.40	123.21	146.28	145.26
Comment Date	115.67	157.74	156.10	163.57	113.15	141.25
of Conversation Comm	0.00	0.00	83.00	449.00	0.00	106.40
Participation	0.00	149.50	149.00	0.00	83.00	76.30
Issue Label	0.00	0.00	149.00	0.00	82.67	46.33
Labels	0.00	0.00	183.00	0.00	0.00	36.60
Num of Commits	0.00	149.00	0.00	0.00	0.00	29.80
Projects	0.00	149.00	0.00	0.00	0.00	29.80
Assignees	0.00	141.00	0.00	0.00	0.00	28.20
Reviewers	0.00	99.00	0.00	0.00	0.00	19.80



Replication Package



Emoji vs. Comments


AOI	Obs	Mean	Std Dev
Emoji			
FFD	11	48609.20	46717.17
SFD	3	65917.00	33214.02
TD	18	77618.07	84460.62
FC	33	42337.13	43821.77
Comment Text			
FFD	15	552.48	839.81
SFD	0	NaN	NA
TD	61	#####	#####
FC	4317	2300.01	6900.21

$$\begin{aligned} & \text{adjusted fixation duration} \\ = & \frac{\text{duration} * \text{total characters}}{\text{character count of AOI type}} \end{aligned}$$

Results – More Tool Oddities


docs/debugging.md


```
23 +
24 + In some cases, unexpected behavior might be caused by misconfigured or unconfigured settings in Atom
    or in one of the packages.
25 +
26 + Open Atom's Settings View with cmd-` or the Atom > Preferences menu option.
```


 Member ...

The default key is Command+Comma `cmd-,` isn't it?


Also, should we be using `<kbd>` instead of `<code>` for keyboard commands?

 Member ...


 to `<kbd>`


 Author Member ...


The default key is Command+Comma `cmd-,` isn't it?

It is -- good catch, thanks! (This will allow me to drop the `<code>` tags and just use backticks )

Also, should we be using `<kbd>` instead of `<code>` for keyboard commands?

I'd be  with that, but if we do decide to do that -- we should do it in other guides as well. For that reason, I'd rather see this change made in another PR.

 Member ...

Agreed on the `<code>` to `<kbd>` change 

```
src/editor-component.coffee
685 + # it will be merged into another one. Therefore, we need to obtain a
686 + # reference to the new selection that contains the originally selected row
687 + rowSelection = _.find editor.getSelections(), (selection) ->
688 +   selection.intersectsBufferRange(bufferRange)
```

See #3433 (comment) for some notes on this (the comment was hidden by another commit I needed to make to make coffeelint happy). 😊

This looks reasonable for now, but I'm wondering if we should revisit the way in which selections are merged after certain API interactions. Maybe adding or updating a selection should always favor the new/modified selection when merging. Or maybe you could call `mergeIntersectingSelections` on the selection itself so your reference remains stable.

This looks reasonable for now, but I'm wondering if we should revisit the way in which selections are merged after certain API interactions. Maybe adding or updating a selection should always favor the new/modified selection when merging. Or maybe you could call `mergeIntersectingSelections` on the selection itself so your reference remains stable.

I wanted to respond to this since I haven't before. While working on these changes, I felt like there was a method missing in the Editor or Selection API which would allow this code to be more elegant in the EditorComponent. Since I'm new to these APIs -- I'm not really sure what would work best, but the having a `mergeIntersectingSelections` method on Selections themselves sounds like a good idea to me. 👍

Threats to Validity

- There may not be enough repositories used for the study to be representative of all of GitHub.
 - To mitigate this threat, all projects are active and have at least 100 stars so that smaller repositories such as abandoned personal projects are not selected.
- In order to provide consistency and ease of replication for the study, all sentiment analysis tools are run with out-of-the-box default settings, despite multiple settings and parameters being available for each.
- Settings customized for the data of this study can negatively affect the accuracy of developer comments not included in the study.

Related Work

- Novielli et al. emphasized the domain-dependent nature of sentiment analysis by running 100 Stack Overflow questions through SentiStrength.
- Huq et al. conducted a statistical study on GitHub PRs to see the difference in sentiment based on whether or not a PR introduced a Fix-Inducing Change (FIC), or code that introduces bugs to a given system
- Lu et al. aimed to determine how other developers interpret these already written issues, PRs, and comments using text representation learning methods to determine an emoji's possible meaning.