

Characterization of Framing and Openness in GitHub Conversations

Aaksha Meghawat, Steven Moore, Qinlan Shen, Steven Dang

Table of Contents

[Table of Contents](#)

[Introduction](#)

[Theoretical Foundations](#)

[Research Questions](#)

[Data Description](#)

[Operationalization of Openness](#)

[Qualitative Analysis](#)

[Tagging for Openness](#)

[Operationalization of Framing](#)

[Qualitative Analysis](#)

[Tagging for Framing](#)

[Transactivity Chains](#)

[Results](#)

[Prediction of Acceptance](#)

[Challenges](#)

[Future Work](#)

[Works Cited](#)

[Appendix A: Thread Metadata Fields](#)

[Appendix B: Threads for Qualitative Analysis](#)

[Appendix C: Qualitative Framing Analysis Raw Annotations](#)

Introduction

Discourse in online work communities differs from that found in weblogs, online support groups, or news sites -- The task-focused yet social nature of the community influences the nature of the conversation, as discussions are not solely focused on the ideas and issues at hand but also the social dynamics of the group (Richards, 2006). This raises the question of what are the dominant manifestations of this social influence on task discussions. Here we propose an analysis of Github pull-request conversation threads, looking at both how openness and framing are leveraged as well as how the two may interact to influence the success of a community pull-request contribution.

We chose to look at the data from the two lenses of openness and framing to gain insights into the interactions of a thread's contributors and how they might affect a thread's

acceptance. Looking at the data through the perspective of openness, we wanted to see how accepting a project's community was when discussing a pull request. We also looked at the data through a framing lense to see how the topics discussed in a thread were indicative of its success. Combining these two lenses would hopefully provide insights into how a thread's contributors made their ideas heard while also playing their part in the project's community.

Theoretical Foundations

Martin and White (2005) describe the construct of engagement in their appraisal framework and explore the linguistic moves that people use to express their stance towards ideas or objects. The framework follows the Bakhtinian school of thought which assumes that every statement expresses a stance and also reveals the stance that the individual assumes the audience has. We want to study the Github conversations from the point of view of the 'engagement' concept in this framework because we think that the way a developer expresses a stance influences the flow of the discussion and would be insightful in understanding the decision and negotiation process among developers. This perspective may allow us to see if the way an idea is expressed (expansively or contractively) can influence how it is taken up by other participants in the conversation and whether it has a significant impact on the acceptance of the idea by everyone else.

Chong & Druckman (2007) discuss framing as a specific perspective on a given subject that is the result of imposing specific values. Framing can be viewed as a consequence of a generative process that imposes an individual's values on their language production. It can also be viewed as a comprehension process where an individual's values and expectations are acted upon by language. The generative process view implies a language generation model that would lead to greater dominance of certain linguistic practices. More specifically, a user with a particular coding style may tend to chime into conversations using a blend of two topics: the general topic of the conversation and the topic of their specific coding style. On the other hand, the comprehension process perspective implies an influence of certain perspectives across a community. Thus, while a specific conversation thread might be dominated by a topic from the tail of a topic distribution (eg: some specific bug or feature to the project), the values of the community might be reflected by a set of topics with greater central tendency. This might also manifest itself in conversations that lack weight in topics that are more aligned with the central tendency of the community. Thus, perspectives not reflecting these values, as seen by alignment, may be less likely to be successful. Similarly, successful pull requests may reflect shifts in framing that reflect the introduction and adoption of community values.

Research Questions

The objective is to look at the Github conversations from the perspective of openness and framing and see if these two perspectives provide an enhanced view and insight into interactions between developers. This would help in understanding why certain conversations culminate in the acceptance or rejection of pull requests and how the developers build

consensus or negotiate the development of open source artifacts. We examined the following research questions:

- 1) How do we detect openness for an individual in a reliable way?
- 2) How do we detect framing for a conversation in a reliable way?
- 3) What interactions are there between between openness and framing within the same conversations?

Data Description

We analyzed pull request conversations from a subset of GitHub projects for Ruby. The data consists of 1862 projects that contain at least one pull request. From these projects, we picked out threads that have at least three different contributors. For a comprehensive list of all the metadata attached to these threads, please refer to Appendix A.

To pick out threads to examine for the qualitative analysis, we sampled threads from both highly active and less active communities. To do this, we stratified the communities according to total number of pull requests and sample two projects above and below the median number of pull requests. For each of these communities, we sampled 3 pull request threads. The threads in the selected communities were divided into three groups based on the number of comments in each thread (low, medium, and high number of comments). We then sampled a thread from each of the thread-length groups. This resulted in a set of 12 pull request discussions from the following projects: thekompanee/fuubar, mperham/connection_pool, mitchellh/vagrant, and thoughtbot/clearance. Details about the threads can be found in Appendix B.

Operationalization of Openness

Qualitative Analysis

We began our analysis on six threads from the two projects mitchellh/vagrant and thoughtbot/clearance. The comments in each thread were broken up into units based on sentence-level delimiters, such as commas and references of code blocks. Each of these units was then hand coded as being either expansive, contractive, or neutral. To code these units as such, we utilized the engagement system in the appraisal network by Martin and White (2007). Units were analyzed for contractive markers of disclaim and proclaim and expansive markers of entertain and attribute. Patterns of heteroglossic and monoglossic utterances were also noted, along with any special markers or themes found in the text, like the use of an emoticon.

It was the case that the majority of the units didn't fall into the expansive or contractive categories, for example ones stating how the code functions or pasting a code block. Even when a unit wasn't explicitly expansive or contractive, the potentially interesting features of that unit were noted in the hand coding. As a result, almost all of the units have at least one

annotation that was considered when generating the patterns and during the affinity diagramming process.

Following the analysis of the first six threads, a list of expansive and contractive patterns was created from common patterns found between the first six threads. These initial patterns were input into the model where we then checked to see if they were appropriately tagging units as expansive or contractive. With these initial patterns in mind, the remaining six threads from the two projects thekompanee/fuubar and mperham/connection_pool were analysed. The same hand coding process was performed on these three threads, but they had the benefit of building upon the existing list of patterns and potentially interesting markers.

After all twelve threads were qualitatively coded for expansive/contractive patterns and potentially interesting markers, we proceeded with affinity diagramming. During the affinity diagramming process, the markers that didn't directly fit into expansive or contractive, such as using a particular acronym or '+1', were grouped into various themes. After fitting them into their final groups, the context of the units for which these markers were used into was counted. This provided us with a ratio of many expansive, contractive, or neutral units these markers fell into. Markers that were most commonly found as either expansive or contractive by 75% or more were used as patterns for the respective classification. For example, the acronym 'imo' was used as a contractive pattern and the use of '?' was used as an expansive pattern.

Tagging for Openness

Once we had some initial expansive and contractive patterns from the analysis of the first six threads, from projects mitchellh/vagrant and thoughtbot/clearance, we ran the patterns through the data. For this, the data was broken down into analysis units that would represent one vein of thought/argument, i.e. that fragment of a sentence between two transition words such as 'but' and punctuations. The code parses through these units and tags them as expansive, contractive, or neither based on the set of input expansive and contractive patterns. We then briefly looked over the results to ensure that the tagging was appropriate and reliable. This process was repeated following the final analysis and affinity diagramming of the twelve threads, which resulted in our final pattern set for expansive and contractive.

The end result of the code provide us with a percentage of expansive and contractive units based on the total number of units per comment. It also provides us with a total count of expansive and contractive units per comment. This data was then aggregated so we had metrics for the percentage and count of expansive and contractive units at the thread level.

Openness	% Sample	% Tagging Scheme
Expansive	20.2	16.36
Contractive	3.78	2.67

Table 1: Percentage of units tagged as expansive or contractive (sample vs. tagged dataset)

Operationalization of Framing

Qualitative Analysis

Qualitative analysis began with reviewing 3 conversations annotating conversation fragments for two primary characteristics: framing related discourse moves and linguistic indicators of described framing moves. From this annotation library, an initial coding dictionary was formed by synthesizing the annotations into groups and forming operational definitions of each group. These codes were carried forward into the analysis of an additional 3 conversations in order to evaluate their robustness, ability to differentiate between statements, and coverage, ability to categorize all statements in the data. New framing moves were annotated and new synthesized definitions were formed from the revised groups. This analysis was repeated for the remaining 2 groups of 3 conversations until a final codebook was formed as shown below in Table 2

Code	Definition
Establishing Statement	Introducing or supporting the factual/valid nature of a proposition
Alignment Statement	Stating agreement/disagreement on a specific discussion point including statements of appreciation
Contrasting Statement	Discussing demonstrated or proposed attributes in contrast to alternatives
Alignment Request	Requesting statement of stances on a specific discussion point
Clarification Request	Requesting restatement/elaboration/clarification of a specific discussion point.

Table 2: Framing Code Dictionary

The resulting code book discovered two broad classes of discourse moves related to framing, statements and requests. Statements describe sentences that introduce information to the discourse while requests are statements that solicit information from others to be introduced, where information can be factual or opinion-based in nature. There were 3 categories of statements defined: Establishing, alignment, and contrasting. There were 2 categories of requests defined: Clarification and alignment.

Establishing statements were the foundation of most discourse where they are describing all discourse moves that are introducing information with the purpose of establishing its factual nature. From a framing perspective these statements would be important to identify in the sense of determining which statements are introducing factual information and being able to identify which statements are debated and which are ignored and/or accepted.

```
It's very difficult to reproduce error because it depends on terminal width.
---
The new tests better exercise the intended behavior of TimedStack
---
I think I chose #require_plugin as it's the same as in Vagrant from where it's
called
```

Figure 1: Sample Establishing Statements

Alignment statements are specifically discourse moves which introduce an individual's agreement or disagreement with a particular fact or discussion point. These statements capture categories of discourse moves intended to introduce individual opinion into the conversation.

```
+1
---
Good idea @jferris
---
Couple of comments but otherwise looks good if Travis passes it.
```

Figure 2: Sample Alignment Statements

Contrasting statements are defined as discourse moves that introduce information intended for comparison or contrasting with previously stated or assumed information. These statements specifically contain multiple perspectives and relational information between each perspective.

```
Seems much easier to just rely on JSON for the encoding/decoding and require
ansible >= 1.2.
---
I would phase it out slowly, i.e. have has_plugin? recognize both current name
and the gem name initially, then slowly deprecate it (with a warning possibly if
the old name way is used).
---
This is not a single use case and, has is referenced by the project, it is
supposed to be generic.
```

Figure 3: Sample Contrasting Statements

Alignment Requests are defined as statements explicitly soliciting the opinions of others. Requests differ from statements in that they can also introduce information, but the primary goal of the discourse move is to introduce information with the intent of soliciting opinions of others regarding the stated information.

```
Could you please have a look there and comment?
```

IMHO this is ready for reviewing.

Thoughts? /cc @croaky @jferris @derekprior

Figure 4: Sample Alignment Requests

Clarification Requests are defined as statements explicitly soliciting additional information from others usually with respect to previously introduced or assumed information. These requests may also introduce information, but with the intent of supporting the specifics of the request for additional information.

Could you explain how I can reproduce this so I can work on a regression test
for it?

Why would I need to pass a count to pop? .

If we leave this in: Is this something that needs to be generated during
clearance install, if the users table exists?

Figure 5: Sample Clarification Requests

While the relationship between each of these individual categories of framing are not clearly connected to framing, the dynamic relationships between them and their occurrences across threads and communities is potentially significant. The twelve threads analyzed here seemed to be accepted or rejected based on the nature of the contribution. Usually a contribution is elaborated early in the conversation and one or two specific points are discussed for the remainder of the thread. In the sample analyzed, two threads from the same community introduced comparable features, lazy loading, but each thread presented the value of the idea from different perspectives. The result is that one thread was merged and the other was rejected. On the other hand many of the rejected threads followed a pattern of short elaborated descriptions followed immediately by at least one contrasting statement which is then followed by alignment statements. These patterns indicate that capturing patterns of these base categories of statements could be valuable in understanding the conversational dynamics. Additional topic analysis could be valuable in order to better understand values within a given work community, though this analysis is not proposed for this project.

Tagging for Framing

Once we have extracted patterns for our framing categories, we tag sentences in each post as belonging into one of the five framing classes. Using the sentence tokenizer in NLTK, we split each post into individual sentences. For each sentence, we then count how many patterns fit the sentence for each of the framing categories. We take the class with the most number of pattern hits as the framing class of the sentence. Sentences that do not exhibit any of

the patterns are assigned to the Establishing Statement class. For each post, we then maintain a count of how the percentage of sentences belonging to each category.

To validate the effectiveness of our tagging scheme, we both compare the percentage of sentences of each category in our hand-coded sample against our entire tagged corpus (Table 3) and randomly sample sentences that were tagged as each of the framing categories.

Framing Category	% Sample	% Tagging Scheme
Establishing Statement	38.64	64.38
Alignment Statement	22.08	14.52
Contrasting Statement	25.00	6.38
Alignment Request	9.09	0.92
Clarification Request	5.19	13.7

Table 3: Percentage of sentences per framing category (sample vs. tagged dataset)

In Table 3, we see that the percentage of Establishing Statements is much higher in our automatically tagged set, in comparison to the hand-coded set. As the Establishing Statements class is used as the class for sentences that do not explicitly match any of the patterns for the other classes, we suspect that the high percentage of Establishing Statements is due to the fact that our current pattern set is missing instances from the other framing categories. The only other framing category with a higher percentage in our automatically tagged dataset than the sample is the Clarification Request tag. The total percentage of Alignment and Clarification Requests is similar to the percentage found in our hand-coded sample, suggesting that we may be picking up on the distinction between requests and statements. The high instance of Clarification Requests in comparison to the Alignment Request class, however, suggests that we are not capturing the distinction between the two types of requests.

While the percentage of sentences tagged in each of the framing categories differed quite widely from the percentages from the hand-coded sample, the sampled sentences from each of the non-Establishing Statement categories generally fit well within our definitions for each category:

merged, thank you

 yeah, that's what i was seeing with some experimentation friday morning.

 thanks @rafaelfranca

Figure 6: Sample Tagged Alignment Statements

i still think we should only support one version of mongoid and i'd go with the

latest one.

seems like they should be treated the same.

it would be nice to mention those results at least in one sentence (maybe with
link to gist) in commit message

Figure 7: Sample Tagged Contrasting Statements

tentative :+1: but i too would like to see @sethvargo's approach

) i would like to draw your attention to this pr which has now been merged and
includes lots of your work.

what do you think?

Figure 8: Sample Tagged Alignment Requests

@alpaca-tc could you please provide a bit more information about registration
of guard::rspec::formatter?

can you add a comment explaining why?

why not have both?

Figure 9: Sample Tagged Clarification Requests

Transactivity Chains

A major aspect of framing is the control of what topics are discussed in a thread. To operationalize this aspect of framing for our model, we introduced the concept of *transactivity chains*, a time-dependent chain of comments in a thread that are similar in what they discuss. The rationale behind these transactivity chains is that they represent different “flows” in the thread about what is being discussed -- one transactivity chain represents the entire lifespan of a certain conversation subject. By examining the different transactivity chains across a thread, we can gain insight into what kind of conversational structures may be indicative of successful collaboration.

We define transactivity chains as a time-ordered chain of comments where posts that are connected have some semantic similarity. To measure the semantic similarity between posts, we run LDA (Blei et al. 2003) over our entire dataset, using 50 topics and treating individual comments as documents. This generates a topic model over our dataset and assigns each post a distribution over the generated topics. We then consider two posts to be semantically similar if they have a cosine distance of less than .5.

We provide a simple example to demonstrate how we represent the transactivity chains as features for our model. Consider Figure 10, where each numbered node represents a comment (time-ordered) in a thread and edges represent that two nodes are part of the same

transactive chain. The thread represented by the figure consists of 6 posts that make up two transactivity chains.

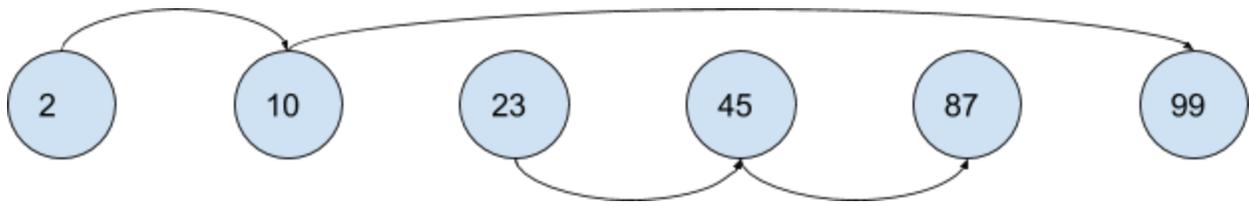


Figure 10: Example transactive chain structure

We encode each of the transactivity threads as 3 features: the number of comments in the transactivity chain (#Comments), the number of comments the transactivity chain spans (ΔT), and the normalized comment distance of the final comment in the chain from the end of the thread (LastTimeSlot). This gives us the following set of features for each of transactivity chains in Figure 10:

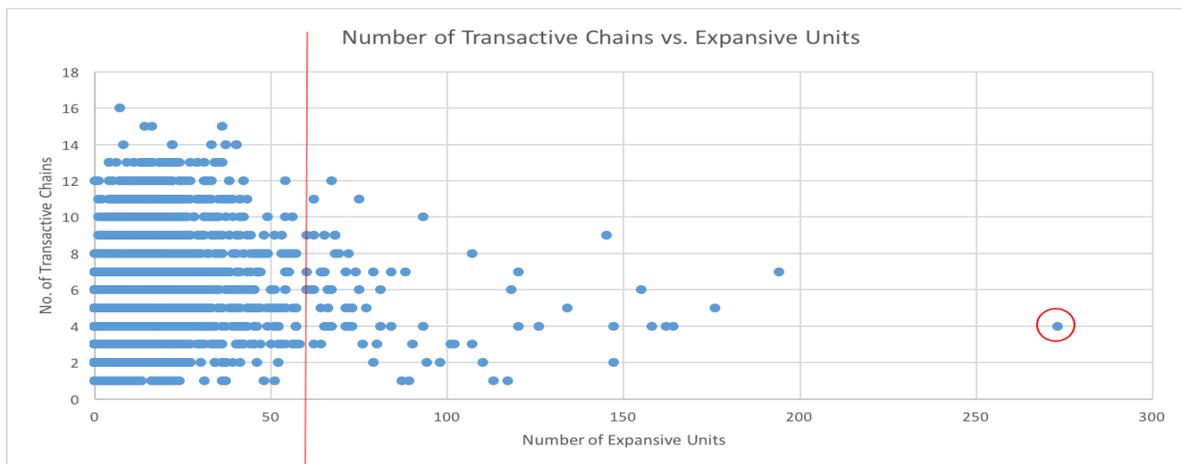
ID	#Comments	ΔT	LastTimeSlot
1	3	6	6/6
2	3	3	5/6

Table 4: Feature representative for transactive train in Figure 10

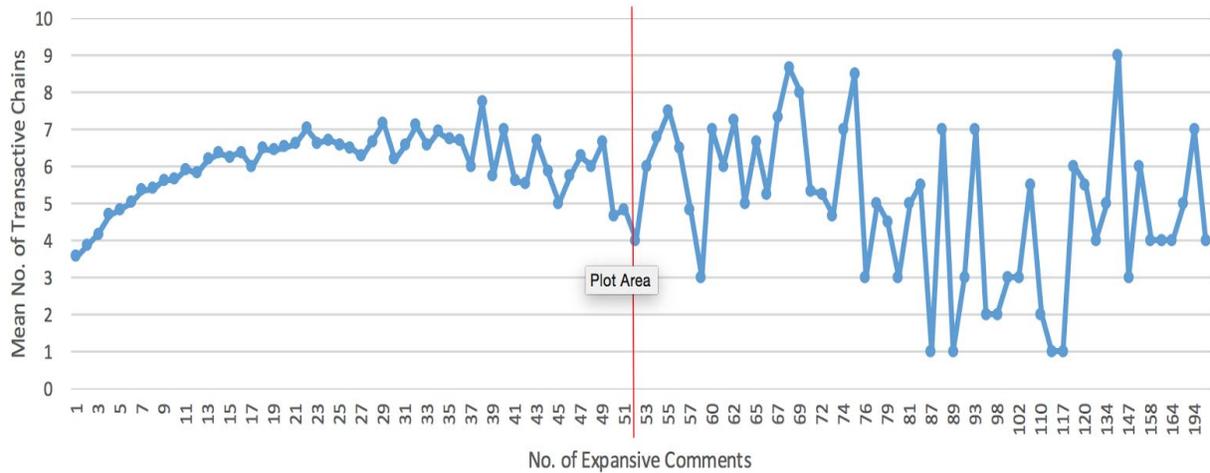
To summarize these features at the thread level, we use the median, mean, min, and max of the transactivity chain features. We also use the median, mean, min, and max values for the percentage and total number of statements tagged as each of the openness and framing features among the transactivity chains of a thread as transactivity chain features.

Results

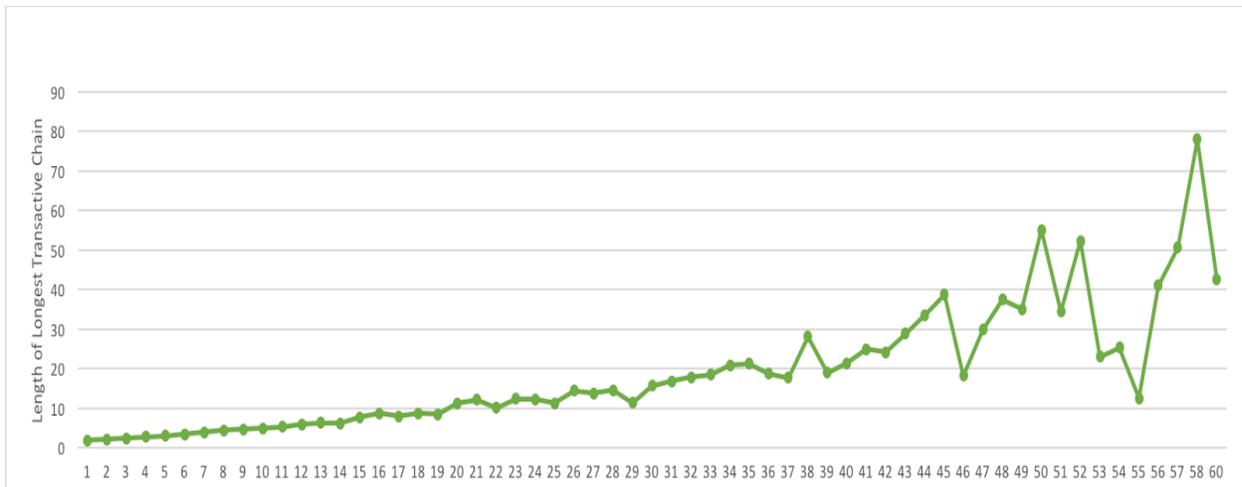
One of the patterns we hoped to observe in the data was the relationship between expansiveness and transactivity. In Graph 1 we see that points with greater than ~50 expansive units are sparse in terms of demonstrating a trend. This also becomes clear from



Graph 2 where there is a lot of variance after the number of expansive units become greater than 50. The general tendency to the left of the red line is that an increase in expansiveness does not necessarily increase the number of transactive chains. On looking at some of the conversations our intuition was that increasing expansiveness created greater dialogic space that encouraged discussion about the current topic, in fact encouraging continuity of a transactive chain and potentially greater consensus building. This intuition is also reflected in Graph 3 where the length of the longest transactive chain increases with increasing presence of expansiveness.

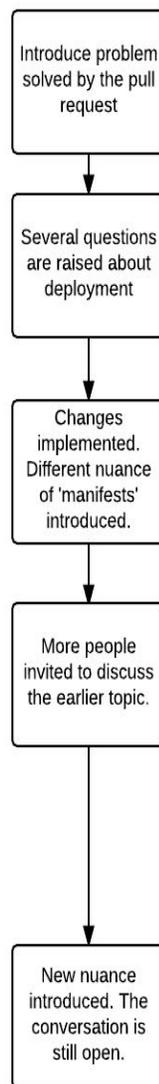
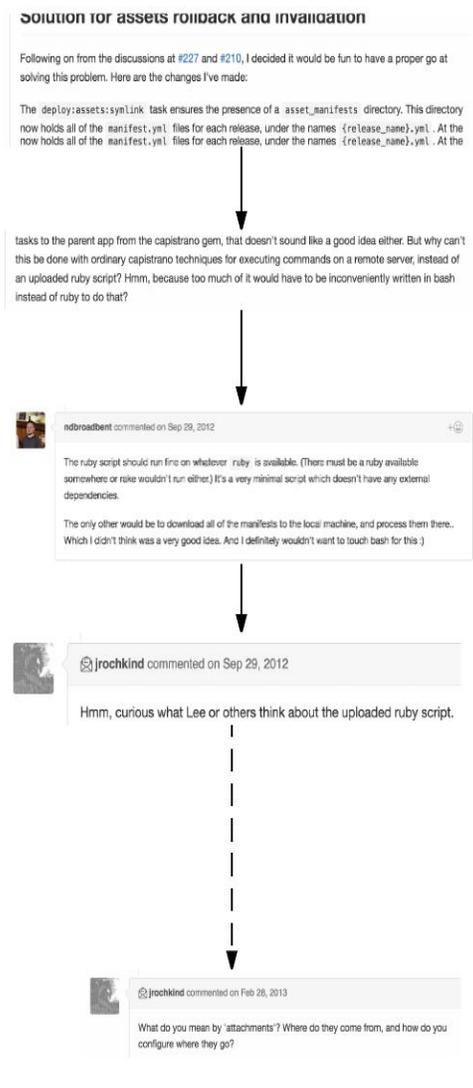


Graph 2: Average number of transactive chains vs. number of expansive comments

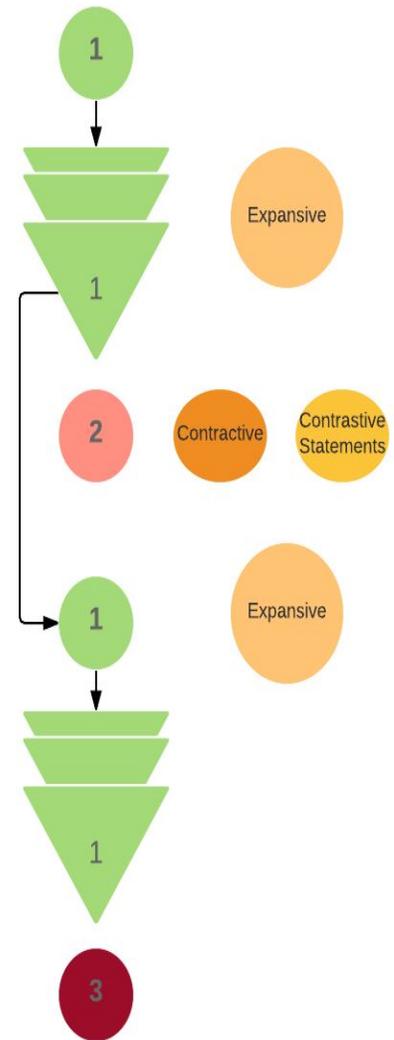


Graph 3: Length of longest transactive chain vs. Expansiveness

As an example we examined the outlier thread (circled in red in Graph 1). We were able to neatly summarize the negotiation that happened in the conversation using our metrics.



Transactive Chain # Annotations



An interesting finding we came across was the interaction between one of our framing categories, clarification requests, and our measure of openness, in particular expansiveness. We found that in threads which had a high percentage of clarification requests and an equally high percentage of expansive units, users were following a common pattern to make their voice heard. Each of the threads shown in Table 5 had users who made posts directly in response to the initial thread comment. In these comments, the users always began by referencing a particular portion of the commit with a code block and then proceeded to ask a question about it. By following such a pattern, the commenters were able to have their question(s) and opinion immediately addressed.

Thread	Clarification Request %	Expansive %
https://github.com/puppetlabs/beaker/pull/48	80%	80%
https://github.com/rails/rails/pull/14668	75%	60%
https://github.com/rubinius/rubinius/pull/2824	57.14%	62.5%

Table 5: Threads and their percentage of clarification requests and expansive units

Another interesting pattern we came across was found in threads that had an above average number of transactive chains and were also comprised of a high percentage, comparatively, of contrastive units. We'd expect threads higher in contrastiveness to have a lower count of transactive chains, as the posts would tend to be monoglossic and dead end. However, we found that in the threads shown in Table 6, the thread's commenters reference other users in the project who have not yet commented on the thread. This brings those users into the thread's conversation where they then contribute to the discussion and generate new transactive chains. It also appears to add to the contrastiveness of the conversation, as it's often the case the users come into the conversation and immediately provide their opinion.

Thread	Number of Transactive Chains	Contrastive %
https://github.com/jekyll/jekyll/pull/1682	5	25%
https://github.com/mitchellh/vagrant/pull/1592	5	21.43%
https://github.com/resque/resque/pull/772	6	21.05%
https://github.com/vcr/vcr/pull/404	5	21.48%

Table 6: Threads and their percentage of clarification requests and expansive units

Prediction of Acceptance

One final experiment that we ran was to use our features to train a classifier to predict whether a pull request would be accepted or rejected. We train a logistic regression classifier and use 5-fold cross validation on our fully tagged dataset (which includes our openness tags, framing category tags, and transactivity chain features) to compare the performance of four different feature configurations:

- Openness: % and total number of each openness category
- Framing: % and total number of each framing category
- Openness + Framing: combination of Openness and Framing features
- Openness + Framing + TC: Openness, Framing, and Transactivity Chain Features (described in the Transactivity Chains section)

We compare these models to a baseline model, which predicts the majority class (Accepted).

Model	Accuracy	Precision	Recall	F1
Baseline	53.80	26.90	50.00	34.98
Openness	53.87	52.45	51.50	47.41
Framing	59.24	59.25	57.72	56.67
Openness + Framing	59.31	59.14	57.97	57.24
Openness + Framing + TC	59.55	59.24	58.44	58.04

Table ###: Classification results for predicting the acceptance of a pull request

At best, we are able to produce a marginally more predictive model than a naive baseline. This has two possible implications. First, our constructs may not be sufficient to describe the dynamics of the work being discussed and the conversations themselves in order to predict the outcome of the conversation. Second, the model of our constructs may not be sufficiently accurate to capture the dynamic of our proposed constructs such that we can leverage them for analysis. The qualitative analysis pointed to weak signals within each construct pointing towards the final outcome. At best, framing played a clear role in only 2 out of 12 conversations. This points to a further refinement of the qualitative analysis of the dataset in order to better capture the conversation dynamics and the constructs that should be leveraged to model the dynamics more accurately to serve predictive purposes.

Challenges

Computational detection of openness and framing is a challenging problem that requires a great deal of insight into the nuances of how and why people express their viewpoints. That being said, our project is more conservative in how it chooses to define openness and framing. For instance, while tagging units as expansive, contractive, or neither based on patterns limits us to labelling texts similar to language we have seen before, the snippets of text that do get labelled as expansive or contractive tend to reliably fit their assigned category. The idea behind using topics to create transactive chains is also a safer approach for capturing the flow of a conversation, as the use of topic to measure the focus of a conversation is fairly well-supported and common (Jo et al. 2011) (Nguyen et al. 2014). Defining an entirely new framework related to framing, on the other hand, is less well-supported. Nevertheless, we were able to use our framing categories to find an interesting example of how users can direct what is being discussed and to predict the acceptance or rejection of a pull request.

By looking at the data through the lense of openness, in particular engagement theory, we chose to ignore the other qualities that make a particular piece of text open in nature. When looking at the data through a framing lense as operationalized by Chong & Druckman, originally the annotations were focused on implicit values embedded in the statements and the values of the statements that spawned discussion versus the ones that did not. However, as the categories in the code book were developed, these categories varied too much between

communities, so more generic categories emerged from the analysis. Further, the strictness of the definitions was frequently challenged each time a new community was coded with the latest code-book as new data would revise the interpretations of the codes and the perceived coverage of the codebook. In the end, a topic agnostic codebook was designed with the hopes that linguistic patterns would be more recognizable across projects, and a pivot to a concept of framing that is more akin to Gee's (2011) building of significance was leveraged in our analysis.

One challenge of applying computational analyses to our data was that quoting another user was included as text of a comment. When a user quotes another user's comment from earlier in the thread, any computational models would iterate through the text again. This can lead to a misrepresentation of both framing and openness counts, as the sentences and units for the quoted text are parsed multiple times. In a similar vein, when a URL is mentioned in the comments of a thread, the openness model splits it by string delimiters. This can cause another misrepresentation of the number of total units within a comment, since the URL itself is being broken into five or more units. Finally, one limitation of trying to run classification experiments to see what we can gain from using our openness and framing features is that a classifier would need reliable feature sets to get reliable results. While we performed qualitative and quantitative analysis on the results of our tagging schemes to assess how well we were able to categorize texts, mislabelling the category of a text will affect both the performance and reliability of a classifier.

Future Work

In the future, we would like to analyze more GitHub data to generate additional patterns and refine our existing sets of patterns for both openness and framing. This would hopefully allow us to detect more interesting interactions between our two categories of features, as well as provide a more reliable basis for building and analyzing a pull-request acceptance classifier. While qualitatively reviewing the twelve sample threads, we also noticed a good amount of politeness theory coming into play in how users interacted with each other. Incorporating this into the project may yield interesting results when integrated with framing and openness.

Also, framing in this analysis followed Gee's model of framing more so than Chong & Druckman. However, as mentioned in the end of the qualitative analysis of framing, the codes and transactive chain annotations would be interesting to combine with a more topic model based analysis in order to look at inferred values within threads and communities and how that interacts with discussion. Based on the qualitative observations made in this project, this could be a fruitful area for further study.

Works Cited

- Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., & Levitan, S. (2007). Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6), 802-822.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3, 993-1022.

- Chattergee, M. (2007). Textual engagements of a different kind? In *Proceedings from Australian Systemic Functional Linguistics Association Congress*.
- Chong, Dennis, and James N. Druckman. "Framing theory." *Annu. Rev. Polit. Sci.* 10 (2007): 103-126.
- Cosentino, Valerio, Javier Luis Cánovas Izquierdo, and Jordi Cabot. "Three Metrics to Explore the Openness of GitHub projects." *arXiv preprint arXiv:1409.4253* (2014).
- Howley, I., Mayfield, E., & Rosé, C. P. (2012). Linguistic analysis methods for studying small groups. *The International Handbook of Collaborative Learning*.
- Gee, J. (2011). *An Introduction to Discourse Analysis: Theory and Method*, Routledge.
- Jo, Y., & Oh, A. H. (2011, February). Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 815-824). ACM.
- Martin, J. & White, P. (2005). *The Language of Evaluation: Appraisal in English*, Palgrave
- Nguyen, V. A., Boyd-Graber, J., Resnik, P., Cai, D. A., Midberry, J. E., & Wang, Y. (2014). Modeling topic control to detect influence in conversations using nonparametric topic models. *Machine Learning*, 95(3), 381-421.
- Read, J., & Carroll, J. (2012). Annotating expressions of appraisal in English. *Language Resources and Evaluation*, 46(3), 421-447.
- Richards, K. (2006). *Language and Professional Identity: Aspects of Collaborative Interaction*, Palgrave, Chapter 3
- Taboada, M., & Grieve, J. (2004, March). Analyzing appraisal automatically. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text* (pp. 158-161). AAAI Press.
- Tsay, J., Dabbish, L., & Herbsleb, J. (2014). Let's talk about it: evaluating contributions through discussion in GitHub. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 144-154). ACM.

Appendix A: Thread Metadata Fields

1. Repo name
2. Owner username
3. Pull request number
4. Pull request state (open/closed)
5. Created time
6. Time of last update
7. Time pull request is closed (can be 0)
8. Accepted? (Y/N)
9. Submitter username
10. Closer username
11. Number of comments

12. Number of lines added
13. Number of lines deleted
14. Number of commits in pull
15. Number of files changed

Appendix B: Threads for Qualitative Analysis

thekompanee/fuubar

- Thread 31 (6 comments): <https://github.com/thekompanee/fuubar/pull/31>
- Thread 32 (6 comments): <https://github.com/thekompanee/fuubar/pull/32>
- Thread 34 (3 comments): <https://github.com/thekompanee/fuubar/pull/34>

mperham/connection_pool

- Thread 37 (12 comments): https://github.com/mperham/connection_pool/pull/37
- Thread 47 (3 comments): https://github.com/mperham/connection_pool/pull/47
- Thread 52 (5 comments): https://github.com/mperham/connection_pool/pull/52

mittchellh/vagrant

- Thread 1255 (3 comments): <https://github.com/mittchellh/vagrant/pull/1255>
- Thread 2131 (8 comments): <https://github.com/mittchellh/vagrant/pull/2131>
- Thread 2617 (22 comments): <https://github.com/mittchellh/vagrant/pull/2617>

thoughtbot/clearance

- Thread 312 (13 comments): <https://github.com/thoughtbot/clearance/pull/312>
- Thread 313 (4 comments): <https://github.com/thoughtbot/clearance/pull/313>
- Thread 334 (8 comments): <https://github.com/thoughtbot/clearance/pull/334>

Appendix C: Raw Initial Annotations from Qualitative Framing Analysis

<https://github.com/thekompanee/fuubar/pull/32>

jfelchner commented on Aug 4, 2012

"Here are the necessary changes to get Fuubar working with the upcoming ruby-progressbar v1.0 release."

- The original requester opens immediately with a statement of the purpose of this pull request
- There is an immediate significance attributed to the request by stating a combination of "Necessary changes" and "working with"
- "Working with [external project] release" points to the value that this comment is supposed to create

"I put out a gem as 1.0.0rc1 so you should be able to test it out and see if it will work for you."

- "[Action] so you should be able to test" highlights the aspects of this request that will allow others to either confirm the claimed value, or find unmentioned issues.
- This comment centers around testing the stated value via "see if IT will work for you", where IT refers to the value mentioned in the previous sentence.

"I took a couple liberties with formatting to take advantage of the new features in v1.0 and you may not agree with them so adjust to taste. :)"

- Here the poster is beginning to mention aspects of his contribution that may not fit with expected conventions and is both highlighting the feature "took a couple liberties" and hedging "you may not agree" and "adjust to taste".
- He punctuates with a positive emoticon to demonstrate a positive attitude around this potentially negative feature in order to encourage open discussion around a potential problem.
- Here the poster begins a new paragraph to separate the content of this issue from the main contribution

"The bar in this PR will output something like:"

- Here the author illustrates the contribution in order to make it more easily digestible. By demonstrating the feature, the author is highlighting a preference for convenience of comprehension.

"I also removed quite a few specs due to the fact that they were now testing across boundaries. You may not agree with that either so just let me know if you want me to revert any of the changes."

- Here the author introduces a new contribution. In this case, it is the removal of some community standards (specs). The author justifies the actions; [action] due to the fact [reasoning].
- The author hedges "You may not agree" and even opens the contribution to changes in this area to conform to values "revert any changes"

"Let me know if you have any questions."

- The author concludes with a friendly request for feedback, but states it as though everything should be correct, and any issues are likely due to misunderstandings.

jfelchner commented on Aug 4, 2012

I also wanted to mention that I am going to keep strictly to SemVer with this gem, so you can safely put ~> 1.0 in your gemspec when we're ready for a final release.

- Here the author adds an additional mention of a feature that he viewed as favorable to the contribution, but did not mention in the original posting.
- This feature highlights conformance with a specific standard that will ensure future smooth integration with known updates

nofxx commented on Aug 12, 2012

"Waiting for this =D"

- Another member supports the contribution as highlighted by both the positive emoticon and the mention that it is an anticipated and thus desired feature by this user.

jeffkreeftmeijer commented on Aug 14, 2012

I released 1.1.0.rc1, it'd be great if you guys could take it for a spin and let me know if everything still works for you.

- Another member mentions a relevant contribution that would be impacted by this one and requests a compatibility check

Thanks a lot for your work, @jfelchner! :)

- Concludes with a statement of appreciation

jfelchner commented on Aug 22, 2012

Seems to be working for me thus far. I went ahead and released ruby-progressbar 1.0.0 final. Feel free to update

at your leisure. :)

- Original poster affirms that the contribution meets the questioned criterion, compatibility with another contribution/release
- The author states that the dependent project that makes this contribution essential had an update released. This introduced an increased urgency to benefit from the contribution.

jeffkreeftmeijer commented on Sep 27, 2012

I just released 1.1.0 (<https://rubygems.org/gems/fuubar/versions/1.1.0>), which uses ruby-progressbar 1.0.0. Thanks again! :)

- Here the first person to challenge the post mentions that the change was pulled in by noting that the latest update utilizes a dependency that wouldn't be possible without this contribution.

<https://github.com/thekompanee/fuubar/pull/34>

hron commented on Aug 5, 2012

"Printing "\e[0m" is not a problem for major terminal emulators, but not for Emacs' dump terminal..."

- The author opens with a statement of the problem this contribution is solving.
- The value of this particular contribution is limited to a particular user group "not a problem ... but not for [noun phrase]"

"Progress bar is just not shown when fuubar works inside Emacs' terminal, even when colors are disabled."

- Elaboration of the problem statement. Problem is stated as a complete failure to perform what sounds like a standard functionality. The severity/significance of the problem is stated by pointing out a possible course of action for a work around that is also invalid "even when ..."

"With this patch a user is able to disable colors and have progress bar displayed in Emacs."

- Concludes with a statement of the value of the contribution.

hron commented on Aug 5, 2012

"Oops. Sorry for the guard commit. ;)"

- Original poster notices an error in the pull request that would interfere with modifications or contributions to this thread by other authors. This makes the proposed contribution have a clear collaboration issue if it is to be incorporated.

jeffkreeftmeijer commented on Aug 6, 2012

Hey Aleksei,

Thanks for helping out! Could you take out the two Guard-commits (e0f7ca0 and 6769091) and add a test to make sure I won't break this in a future release?

Thanks again!

-- Jeff

- Here another admin asks to remove the aspects of the contribution that would lead to collaboration issues if incorporated as evidenced by referencing "Guard-commits" as previously mentioned.
- The admin also asks the author to add an additional test that will help prevent future collaboration issues that may be introduced by this feature.

jfelchner commented on Aug 15, 2013

@hron added here: 6c93377

Thanks!

- Here another contributor has extracted the core of the contribution and made the change in another contribution/branch. This allows for the original author not to have to undo the errors made in this request.
-

<https://github.com/thekompanee/fuubar/pull/31>

[DarthSim](#) commented [on Aug 3, 2012](#)

Old versions of ruby-progressbar occur SystemStackError in some cases

- Opens with a simple statement of the contribution.
- This contribution highlights an error that causes an undesired behavior in the system "Old [behavior] occur ... in some cases"
- The introduction is very short and blunt. It seems the contribution is implicit based on the problem statement

[jeffkreeftmeijer](#) commented [on Aug 5, 2012](#)

Hey Sergey,

Could you explain how I can reproduce this so I can work on a regression test for it?

Thanks for helping out!

-- Jeff

- The first response asks for additional information that will enable collaboration "Could you explain... so I can work..."
- This also mentions the development of a test so that future development will not be affected by the contribution.

[DarthSim](#) commented [on Aug 5, 2012](#)

Hi Jeff,

It's very difficult to reproduce error because it depends on terminal width. In some cases calculation of bar width may fall into infinite loop.

- Author points out the specifics of the issues because of the previous difficulty in finding the significance of the contribution because others couldn't replicate the issues it claims to fix.
- By elaborating the details of the exact circumstances that produce the issue "In some cases [issue]", "difficult ... because [reason]".

I fixed this error and author of ruby-progressbar committed my pull request.

- Reiterates the value of the contribution and tries to reinforce the significance and value by pointing out the value that it brought to another project through their inclusion of a similar contribution.

I had a headache when I tried to understand why my empty tests fail.

-

As I have noticed just now, jfelchner didn't push updated code to rubygems.org, so this pull request makes no sense. He develops new version of ruby-progressbar, it may be good to use it.

- Seems to retract a claim made earlier in this statement about the significance signaled by inclusion by another project.
- The author starts to devalue his contribution "pull request makes no sense"
- He points to the other project and how it may provide the claimed value.

[jeffkreeftmeijer](#) commented [on Aug 14, 2012](#)

I just released 1.1.0.rc1, which uses ruby-progressbar 1.0.0.rc1. Could you try that and let me know if that solves your problem? :)

- Admin points to a separate contribution which utilizes the other project and thus should have incorporated the originally claimed fix.
- The admin requests the author to test whether the new release still demonstrates the same issue as far as the author could reproduce it.

[DarthSim](#) commented [on Aug 14, 2012](#)

Yep it does, thank you!

Just a note: version from rubygems.org still uses old ruby-progressbar v0.0.10

- Here the original author comes into agreement that the alternative proposed path was acceptable.

[jeffkreeftmeijer](#) commented [on Aug 14, 2012](#)

Hey Sergey, Thanks for letting me know. However, the rc

(<https://rubygems.org/gems/fuubar/versions/1.1.0.rc1>) actually uses the correct version, but the gem page (<https://rubygems.org/gems/fuubar>) still shows fuubar 1.0.0, as the rc is a prerelease. :)

[jfelchner](#) commented [on May 29, 2013](#)

[@jeffkreeftmeijer](#) this can be closed.

https://github.com/mperham/connection_pool/pull/47

rbrain commented on Feb 5, 2014

*.swp for vim temporary files

.rdoc for TAGS generation via the rdoc-tags plugin

TAGS for easily navigating source with editors

Ignore more development files ...

*.swp for vim temporary files

.rdoc for TAGS generation via the rdoc-tags plugin

TAGS for easily navigating source with editors

- Here the author highlights the problem implicitly in the commit text.
- The contribution text is an enumeration of each of the specific contributions that have been added

mperham commented on Feb 5, 2014

This is the type of stuff that should go in your global gitignore, yes?

- Here another contributor questions whether this contribution is of the appropriate type. "This is the type that should go...your [location]?"

djanowski commented on Feb 5, 2014

I was about to make the same comment as @mperham.

- An additional contributor reiterates support for the first commenter's point as evidenced by "@mperham" and "I ... same as [other person]"

drbrain commented on Feb 5, 2014

I guess? I put this kind of stuff in my project .gitignores, but my Rakefile also handles things like creating TAGS files and so-forth for me too.

- Here the original author is hesitant to agree with the stated norms of the community as evidenced by following a statement of uncertain agreement "I guess" with an explanation of reasoning for the significance of the contribution.