Skill Space

Representation of Developer Expertise in Open Source Software

(2021) - Tapajit Dey, Andrey Karnauch, Audris Mockus

Adam Cook

CS540 - Advanced Software Engineering

March 4, 2024

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Outline

- Introduction
    - Defining “Expertise”
    - Social vs. Technical Trust
- Research Problem
    - 5 Hypotheses
- Methodology
    - Skill Space
    - Vector Embedding
- Results
- Future Work

What is “expertise”?

- Two factors play into building trust between developers
  - Social aspect - interactions between developers
    - Only enhance trust in an already-established developer circle
  - Technical aspect - interactions between the developer and projects
    - Referred to as **developer expertise**
- Gauging expertise is necessary to repeat interactions
  - Establish trust between developers
  - Increase pull request acceptance
  - Frequent issue resolution

Research Problem

- “Develop a feasible representation of a developer’s expertise in specific focus areas of software development”
  - Different ways of measuring trust
    - Social = Qualitative
    - Technical = Quantitative

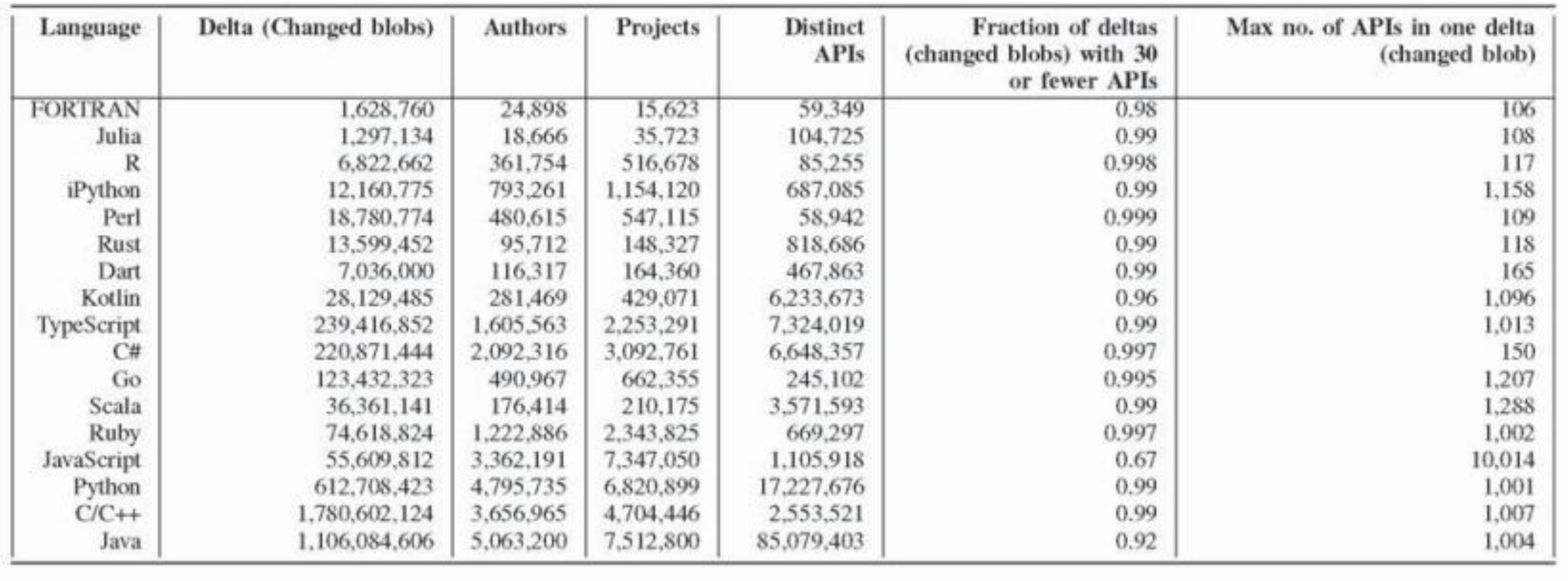
5 Hypotheses

- **H1**: Developers are likely to choose new APIs closer to what they already know
- **H2**: Developers a likely to join new projects that align with them
- **H3**: A project is likely to accept contributions from aligned developers
- **H4**: Developers aligned with projects have higher pull request acceptances
- **H5**: A developer’s API skills are aligned with their own representation

Skill Space

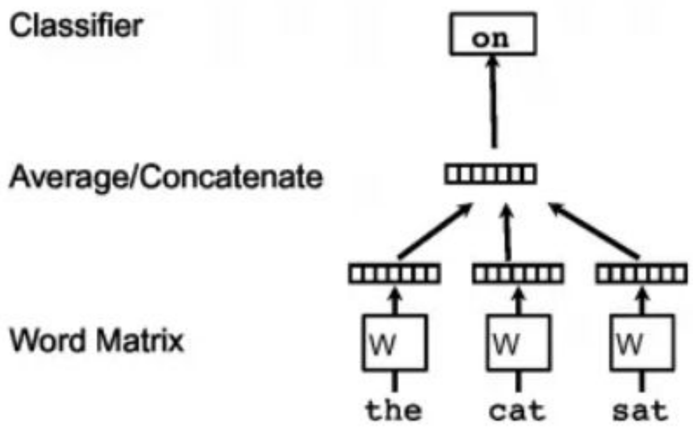
- Topology of developers, projects, APIs, and programming languages
  - “Skill Vectors”
- World of Code used to extract developer, project & API information
  - Only used data starting from February 2019
    - About ~2 years worth of data
- Provides vector representation of expertise
  - API to API
  - Developer to API
  - Project to API representation
  - Developer representation to API

Data Collected



Language	Delta (Changed blobs)	Authors	Projects	Distinct APIs	Fraction of deltas (changed blobs) with 30 or fewer APIs	Max no. of APIs in one delta (changed blob)
FORTRAN	1,628,760	24,898	15,623	59,349	0.98	106
Julia	1,297,134	18,666	35,723	104,725	0.99	108
R	6,822,662	361,754	516,678	85,255	0.998	117
iPython	12,160,775	793,261	1,154,120	687,085	0.99	1,158
Perl	18,780,774	480,615	547,115	58,942	0.999	109
Rust	13,599,452	95,712	148,327	818,686	0.99	118
Dart	7,036,000	116,317	164,360	467,863	0.99	165
Kotlin	28,129,485	281,469	429,071	6,233,673	0.96	1,096
TypeScript	239,416,852	1,605,563	2,253,291	7,324,019	0.99	1,013
C#	220,871,444	2,092,316	3,092,761	6,648,357	0.997	150
Go	123,432,323	490,967	662,355	245,102	0.995	1,207
Scala	36,361,141	176,414	210,175	3,571,593	0.99	1,288
Ruby	74,618,824	1,222,886	2,343,825	669,297	0.997	1,002
JavaScript	55,609,812	3,362,191	7,347,050	1,105,918	0.67	10,014
Python	612,708,423	4,795,735	6,820,899	17,227,676	0.99	1,001
C/C++	1,780,602,124	3,656,965	4,704,446	2,553,521	0.99	1,007
Java	1,106,084,606	5,063,200	7,512,800	85,079,403	0.92	1,004

Vector Embedding with doc2vec



Classifier

on

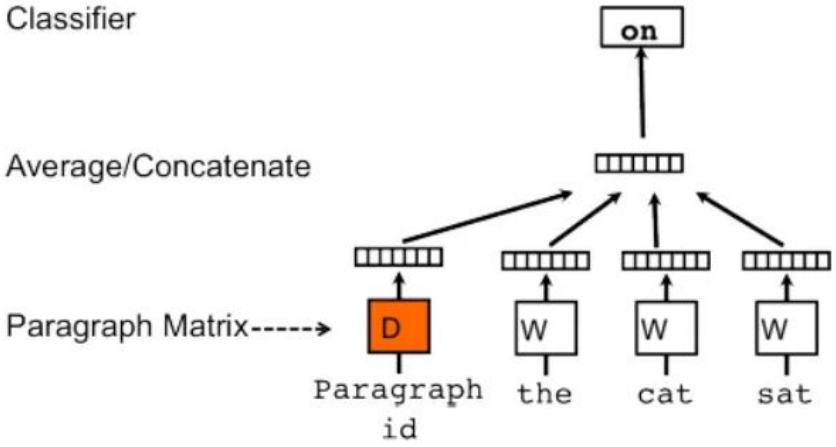
Average/Concatenate

Word Matrix

W W W

the cat sat

word2vec



Classifier

on

Average/Concatenate

Paragraph Matrix----->

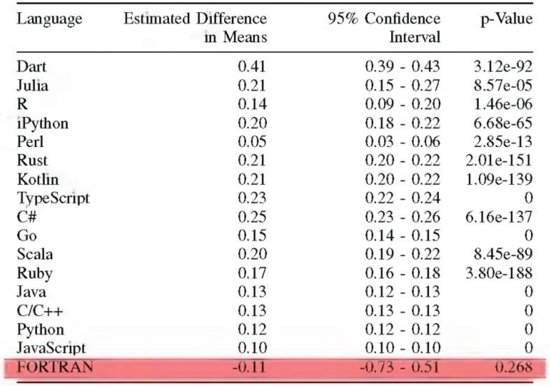
D W W W

Paragraph id the cat sat

doc2vec

Results (H1)

- APIs used in the future are more closely compared to random APIs they didn’t use
  - Most all P-values are much smaller than or equal to 0
  - Sample space for FORTRAN was relatively small



Language	Estimated Difference in Means	95% Confidence Interval	p-Value
Dart	0.41	0.39 - 0.43	3.12e-92
Julia	0.21	0.15 - 0.27	8.57e-05
R	0.14	0.09 - 0.20	1.46e-06
iPython	0.20	0.18 - 0.22	6.68e-65
Perl	0.05	0.03 - 0.06	2.85e-13
Rust	0.21	0.20 - 0.22	2.01e-151
Kotlin	0.21	0.20 - 0.22	1.09e-139
TypeScript	0.23	0.22 - 0.24	0
C#	0.25	0.23 - 0.26	6.16e-137
Go	0.15	0.14 - 0.15	0
Scala	0.20	0.19 - 0.22	8.45e-89
Ruby	0.17	0.16 - 0.18	3.80e-188
Java	0.13	0.12 - 0.13	0
C/C++	0.13	0.13 - 0.13	0
Python	0.12	0.12 - 0.12	0
JavaScript	0.10	0.10 - 0.10	0
FORTRAN	-0.11	-0.73 - 0.51	0.268

Results (H2)

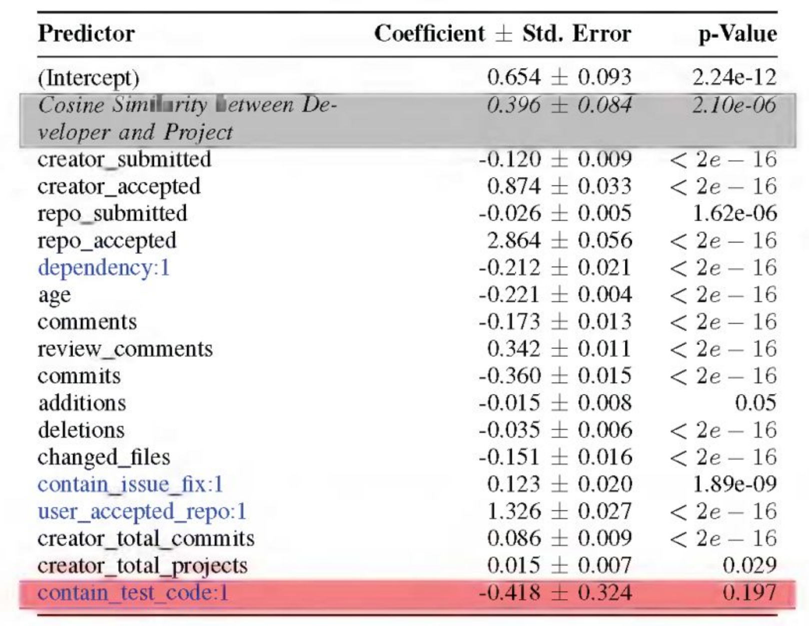
- Significant difference between estimated means and confidence interval
  - Validate expectation that a developer will join new projects that are more aligned with them than a random project
  - Used t-test to measure significant difference between aligned vs. random projects of the same language
    - P-value < 2.2e-16
    - 95% confidence interval of [0.013, 0.021]

Results (H3)

- Compared skill vectors between aligned developers and randomly chosen developers
- Differences between alignments were significant using the t-test
    - P-value < 2.2e-16
    - 96% confidence interval on [0.126, 0.156]

Results (H4)

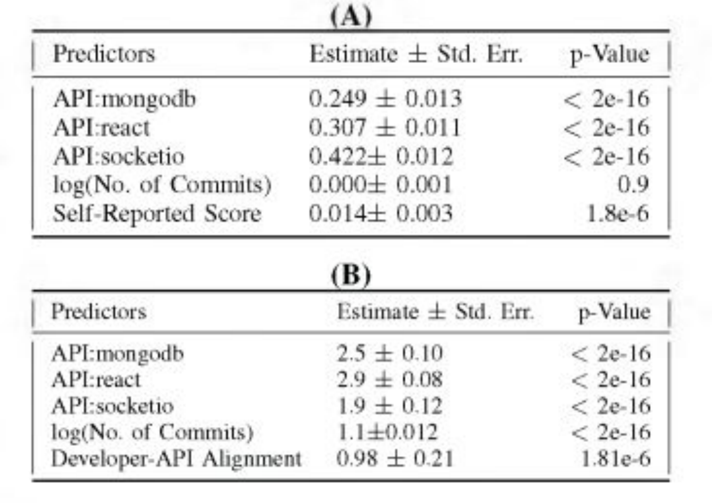
- The closer a developer’s alignment is to a project, the higher the chance their pull request is accepted
  - Data restricted to Pull Requests
  - Logistic regression shows positive coefficient, even after factoring social aspects
  - ‘deletions’ found to be insignificant, only because of sampled data



Predictor	Coefficient ± Std. Error	p-Value
(Intercept)	0.654 ± 0.093	2.24e-12
Cosine Similarity between Developer and Project	*0.396 ± 0.084*	*2.10e-06*
creator_submitted	-0.120 ± 0.009	$< 2e - 16$
creator_accepted	0.874 ± 0.033	$< 2e - 16$
repo_submitted	-0.026 ± 0.005	1.62e-06
repo_accepted	2.864 ± 0.056	$< 2e - 16$
dependency:1	-0.212 ± 0.021	$< 2e - 16$
age	-0.221 ± 0.004	$< 2e - 16$
comments	-0.173 ± 0.013	$< 2e - 16$
review_comments	0.342 ± 0.011	$< 2e - 16$
commits	-0.360 ± 0.015	$< 2e - 16$
additions	-0.015 ± 0.008	0.05
deletions	-0.035 ± 0.006	$< 2e - 16$
changed_files	-0.151 ± 0.016	$< 2e - 16$
contain_issue_fix:1	0.123 ± 0.020	1.89e-09
user_accepted_repo:1	1.326 ± 0.027	$< 2e - 16$
creator_total_commits	0.086 ± 0.009	$< 2e - 16$
creator_total_projects	0.015 ± 0.007	0.029
contain_test_code:1	-0.418 ± 0.324	0.197

Results (H5)

- An increase in skill alignment has a positive relation with self-reported score
  - Obtained survey data from GH users
  - Compared to Javascript libraries:*mongodb, socketio, react*
  - Tables shows as self-reported score increases, API alignment (A) and number of commits (B) also increase



(A)

Predictors	Estimate ± Std. Err.	p-Value
API:mongodb	0.249 ± 0.013	$< 2e-16$
API:react	0.307 ± 0.011	$< 2e-16$
API:socketio	0.422 ± 0.012	$< 2e-16$
log(No. of Commits)	0.000 ± 0.001	0.9
Self-Reported Score	0.014 ± 0.003	1.8e-6

(B)

Predictors	Estimate ± Std. Err.	p-Value
API:mongodb	2.5 ± 0.10	$< 2e-16$
API:react	2.9 ± 0.08	$< 2e-16$
API:socketio	1.9 ± 0.12	$< 2e-16$
log(No. of Commits)	1.1 ± 0.012	$< 2e-16$
Developer-API Alignment	0.98 ± 0.21	1.81e-6

Future Work

- Branch out to non-technical skills
  - Communication, collaboration
- Further applications:
  - Determine if a “developer” is a bot account
  - Check alignment of skill vectors of different developers for identity resolution
  - Infer transparency of corresponding software supply chains