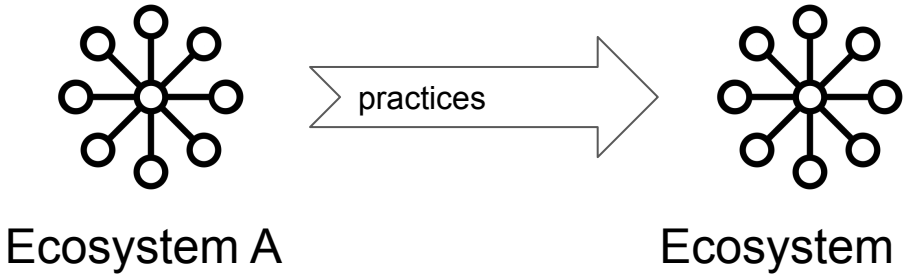
Documentation Practices of Polyglot Developers

Mohammad Eglil
BSc thesis first presentation
Supervised by:
Dr. Pooja Rani
Prof. Timo Kehrer
1st of March 2023

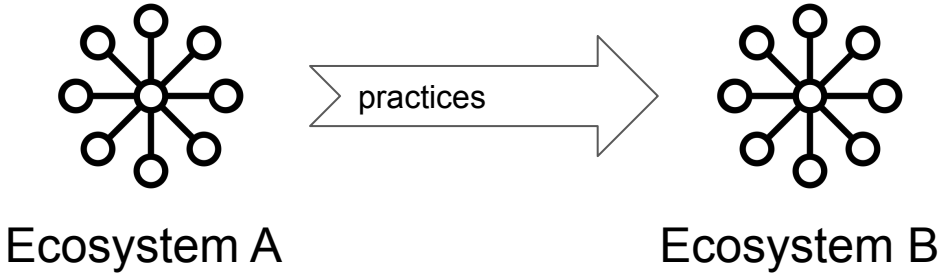
Motivation

Motivation

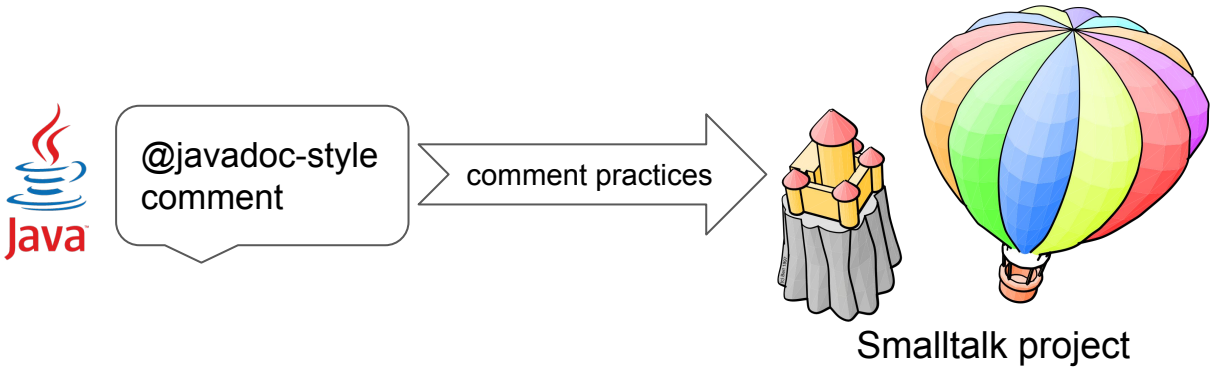


Ma et al, ‘‘World of Code: Enabling a Research Workflow for Mining and Analyzing the Universe of Open Source VCS data”, 2020

Motivation



Ma et al, ‘‘World of Code: Enabling a Research Workflow for Mining and Analyzing the Universe of Open Source VCS data”, 2020



Rani et al, ‘What do class comments tell us? An investigation of comment evolution and practices in Pharo Smalltalk’, 2021

Goal

Is it just an anecdote?

Goal

Is it just an anecdote?

who (developers) carries over commenting practices from Java to Python or vice versa?

Goal

Is it just an anecdote?

who (developers) carries over commenting practices from Java to Python or vice versa?

Is there a correlation in their expertise and their documentation practices?

Implications

Implications

Industry practitioners could onboard newcomers more easily

Implications

Industry practitioners could onboard newcomers more easily

Linters could be profile-based

Related work

2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering

A Large Scale Study of Multiple Programming Languages and Code Quality

Pavneet Singh Kochhar, Dinusha Wijedasa, and David Lo
School of Information Systems
Singapore Management University

Investigating the Effect of Polyglot Programming on Developers

Cole S. Peterson
Department of Computer Science and Engineering
University of Nebraska - Lincoln
Lincoln, NE USA

An Empirical Assessment of Polyglot-ism in GitHub

Federico Tomassetti
Dept. Control and Computer Engineering
Politecnico di Torino
Turin, Italy
federico.tomassetti@polito.it

Marco Torchiano
Dept. Control and Computer Engineering
Politecnico di Torino
Turin, Italy
marco.torchiano@polito.it

ABSTRACT

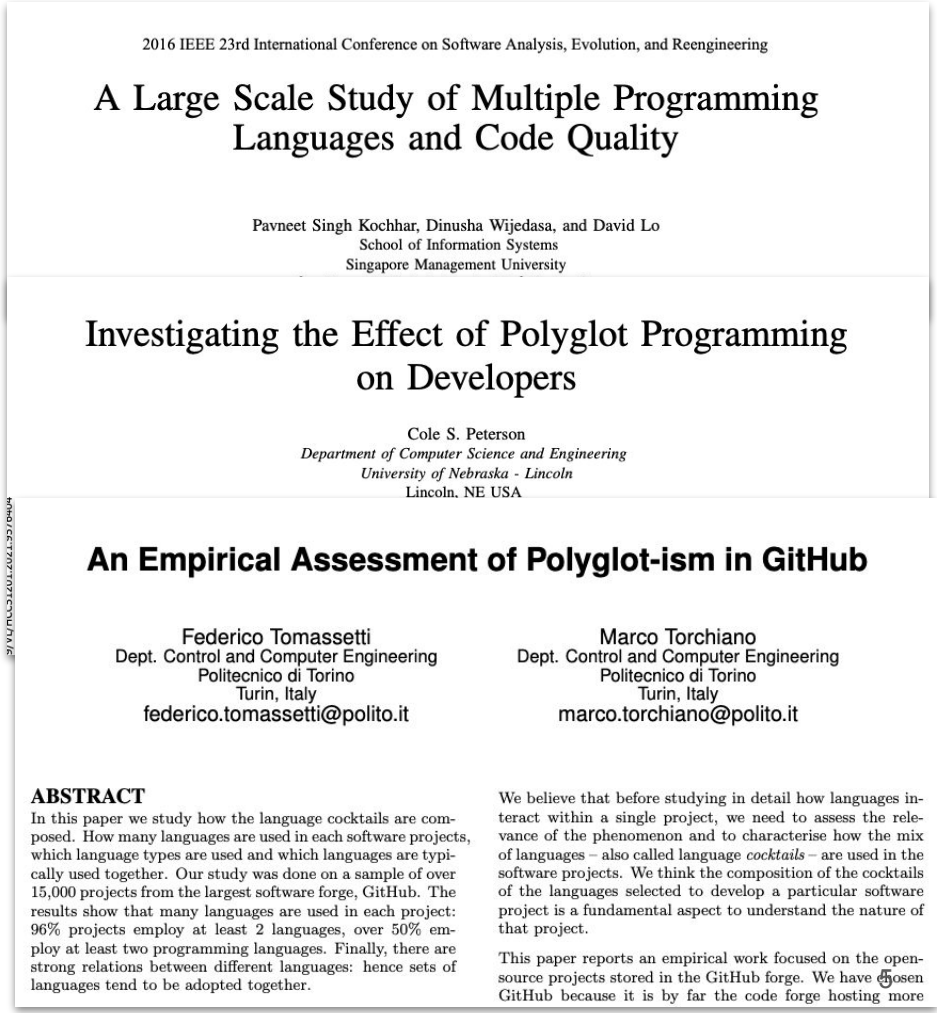
In this paper we study how the language cocktails are composed. How many languages are used in each software projects, which language types are used and which languages are typically used together. Our study was done on a sample of over 15,000 projects from the largest software forge, GitHub. The results show that many languages are used in each project: 96% projects employ at least 2 languages, over 50% employ at least two programming languages. Finally, there are strong relations between different languages: hence sets of languages tend to be adopted together.

We believe that before studying in detail how languages interact within a single project, we need to assess the relevance of the phenomenon and to characterise how the mix of languages – also called language *cocktails* – are used in the software projects. We think the composition of the cocktails of the languages selected to develop a particular software project is a fundamental aspect to understand the nature of that project.

This paper reports an empirical work focused on the open-source projects stored in the GitHub forge. We have chosen GitHub because it is by far the code forge hosting more

Related work

Technical background and language preference in polyglot environments is not much explored



2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering

A Large Scale Study of Multiple Programming Languages and Code Quality

Pavneet Singh Kochhar, Dinusha Wijedasa, and David Lo
School of Information Systems
Singapore Management University

Investigating the Effect of Polyglot Programming on Developers

Cole S. Peterson
Department of Computer Science and Engineering
University of Nebraska - Lincoln
Lincoln, NE USA

An Empirical Assessment of Polyglot-ism in GitHub

Federico Tomassetti
Dept. Control and Computer Engineering
Politecnico di Torino
Turin, Italy
federico.tomassetti@polito.it

Marco Torchiano
Dept. Control and Computer Engineering
Politecnico di Torino
Turin, Italy
marco.torchiano@polito.it

ABSTRACT

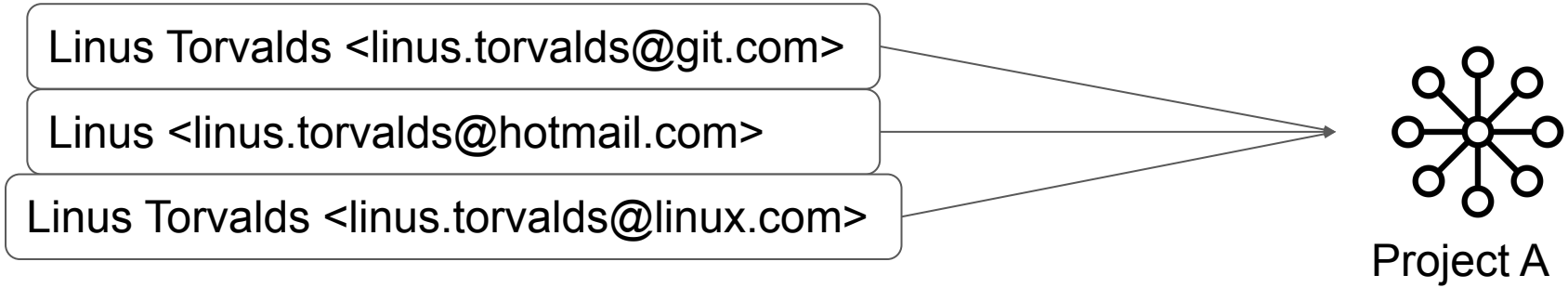
In this paper we study how the language cocktails are composed. How many languages are used in each software projects, which language types are used and which languages are typically used together. Our study was done on a sample of over 15,000 projects from the largest software forge, GitHub. The results show that many languages are used in each project: 96% projects employ at least 2 languages, over 50% employ at least two programming languages. Finally, there are strong relations between different languages: hence sets of languages tend to be adopted together.

We believe that before studying in detail how languages interact within a single project, we need to assess the relevance of the phenomenon and to characterise how the mix of languages – also called language *cocktails* – are used in the software projects. We think the composition of the cocktails of the languages selected to develop a particular software project is a fundamental aspect to understand the nature of that project.

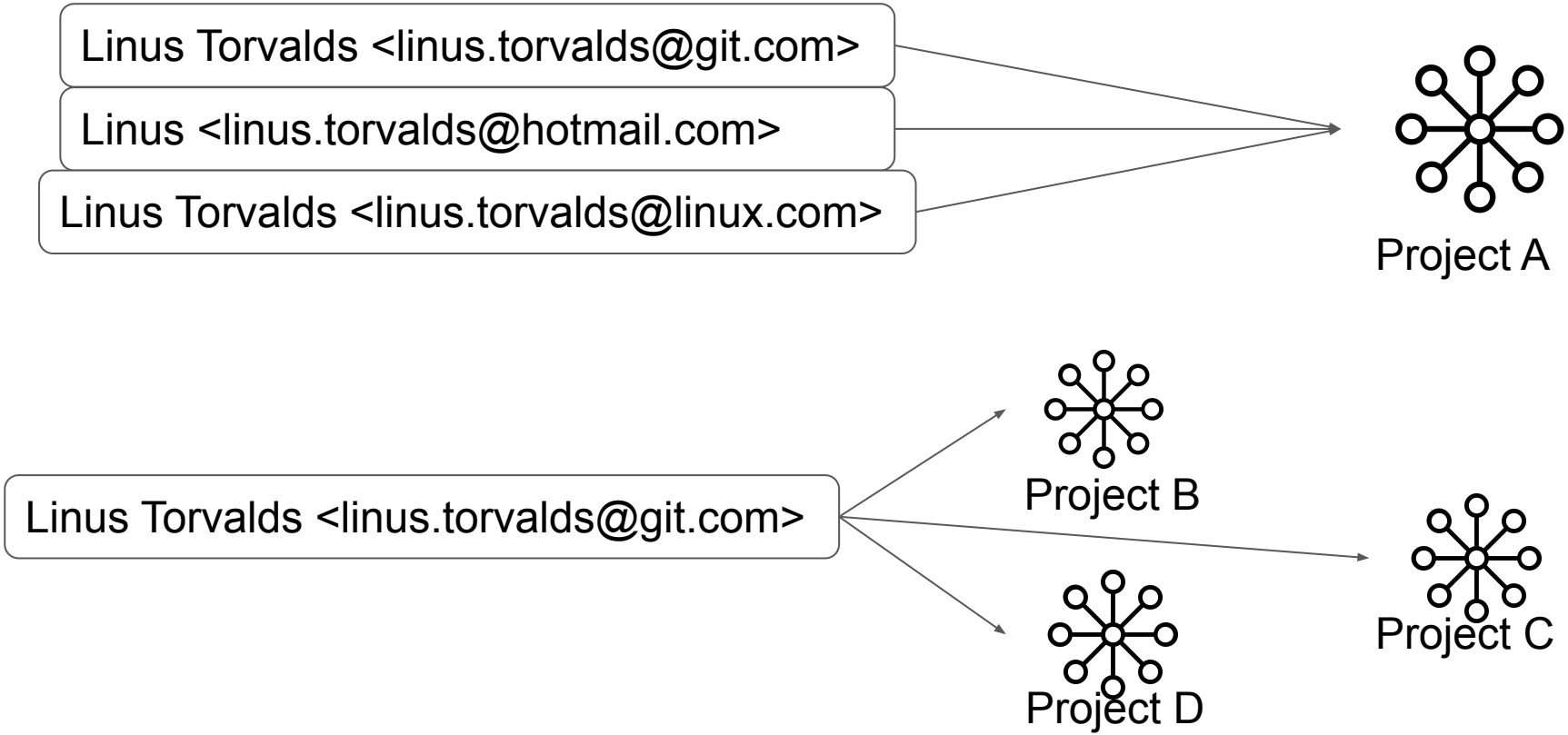
This paper reports an empirical work focused on the open-source projects stored in the GitHub forge. We have chosen GitHub because it is by far the code forge hosting more

Data collection

Data collection



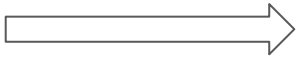
Data collection



Infrastructure

Infrastructure

World of Code



very detailed

Infrastructure

World of Code ⟹ very detailed

GitHub API ⟹ very efficient

Infrastructure

World of Code → very detailed

GitHub API → very efficient

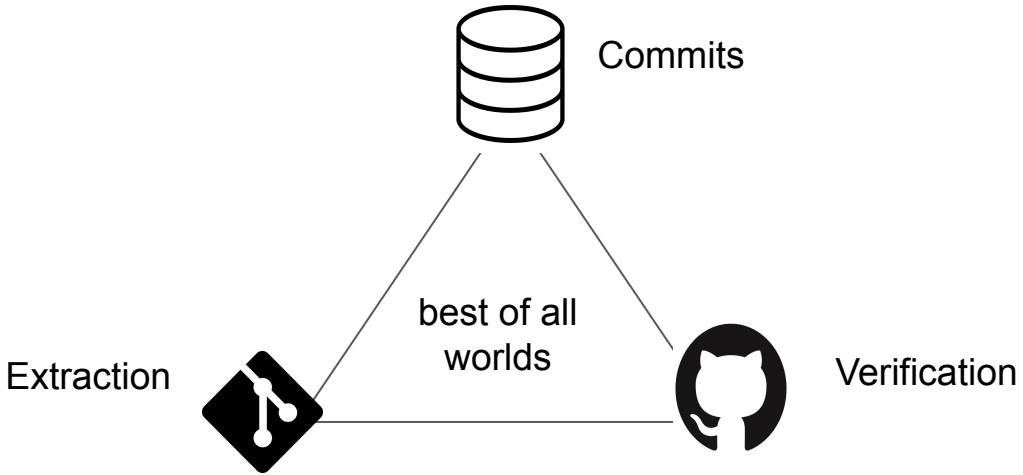
Git → very adaptable

Infrastructure

World of Code → very detailed

GitHub API → very efficient

Git → very adaptable



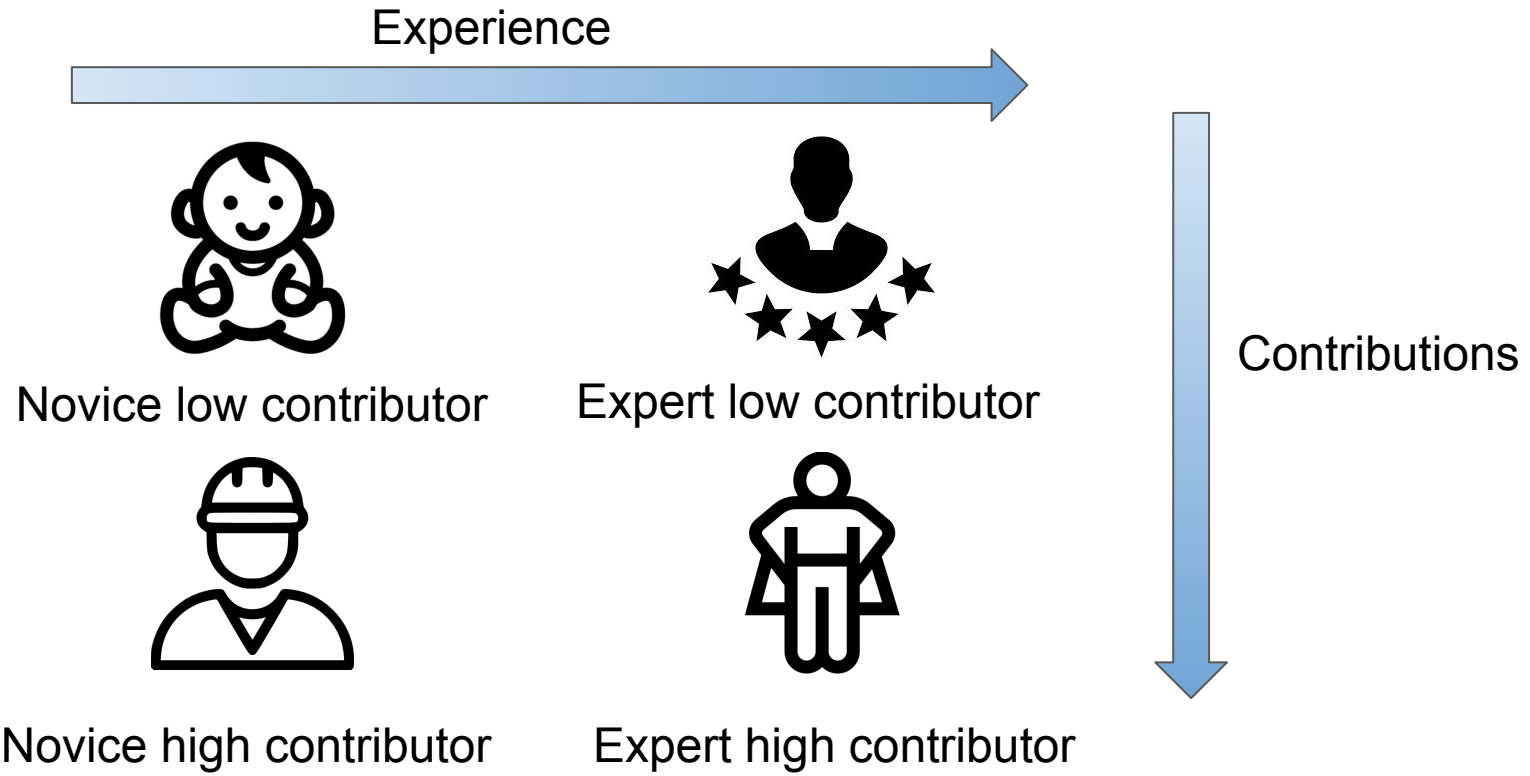
Overview Java projects

Projects	Authors	Commits	Changed code lines
Guice	101	2k	2 M
Guava	407	5k	2 M
Spark	2173	27k	20 M
Vaadin	242	18k	4 M
Eclipse	304	34k	11 M
Hadoop	503	24k	10 M

Overview Python projects

Projects	Authors	Commits	Changed code lines
Mailpile	192	5k	1.4 Mio
Requests	691	4k	0.5 Mio
Pipenv	392	5k	1.8 Mio
iPython	849	19k	3.2 Mio
Pandas	2472	22k	3.8 Mio
Django	2298	28k	4.8 Mio
Pytorch	2356	33k	6.7 Mio

Contributors



Metrics

```
331  348                 if not self.query.standard_ordering:
332  349                     field = field.copy()
333  350                     field.reverse_ordering()
334       -             if isinstance(field.expression, F) and (
335       -                 annotation := self.query.annotation_select.get(
336       -                     field.expression.name
337       -                 )
     351  +             select_ref = selected_exprs.get(field.expression)
     352  +             if select_ref or (
     353  +                 isinstance(field.expression, F)
     354  +                 and (select_ref := selected_exprs.get(field.expression.name))
338  355                 ):
339       -                 field.expression = Ref(field.expression.name, annotation)
340       -             yield field, isinstance(field.expression, Ref)
     356  +                 # Emulation of NULLS (FIRST|LAST) cannot be combined with
     357  +                 # the usage of ordering by position.
     358  +                 if (
     359  +                     field.nulls_first is None and field.nulls_last is None
     360  +                 ) or self.connection.features.supports_order_by_nulls_modifier:
     361  +                     field.expression = select_ref
     362  +                 # Alias collisions are not possible when dealing with
     363  +                 # combined queries so fallback to it if emulation of NULLS
     364  +                 # handling is required.
     365  +                 elif self.query.combinator:
     366  +                     field.expression = Ref(select_ref.refs, select_ref.source)
     367  +             yield field, select_ref is not None
341  368                 continue
342  369         if field == "?":  # random
343  370             yield OrderBy(Random()), False
```

Metrics

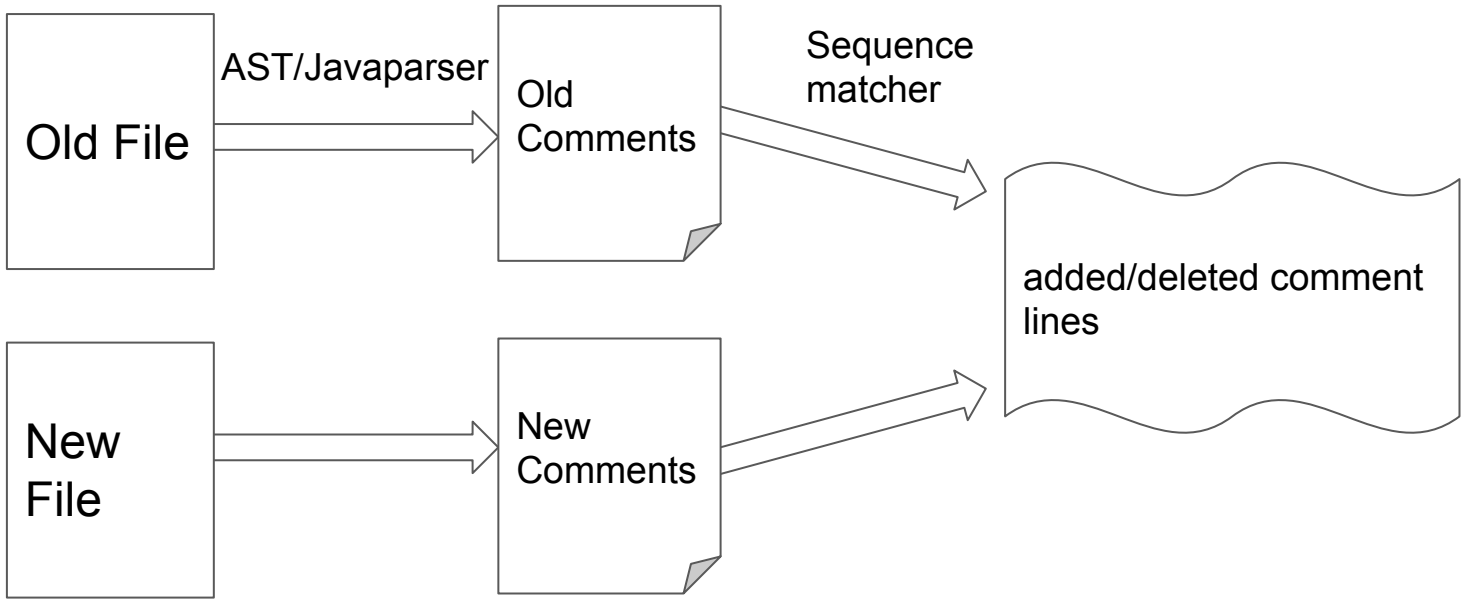
```
331  348                 if not self.query.standard_ordering:
332  349                     field = field.copy()
333  350                     field.reverse_ordering()
334      -               if isinstance(field.expression, F) and (
335      -                   annotation := self.query.annotation_select.get(
336      -                       field.expression.name
337      -                   )
     351 +               select_ref = selected_exprs.get(field.expression)
     352 +               if select_ref or (
     353 +                   isinstance(field.expression, F)
     354 +                   and (select_ref := selected_exprs.get(field.expression.name))
338  355                 ):
339      -                   field.expression = Ref(field.expression.name, annotation)
340      -               yield field, isinstance(field.expression, Ref)
     356 +                   # Emulation of NULLS (FIRST|LAST) cannot be combined with
     357 +                   # the usage of ordering by position.
     358 +                   if (
     359 +                       field.nulls_first is None and field.nulls_last is None
     360 +                   ) or self.connection.features.supports_order_by_nulls_modifier:
     361 +                       field.expression = select_ref
     362 +                   # Alias collisions are not possible when dealing with
     363 +                   # combined queries so fallback to it if emulation of NULLS
     364 +                   # handling is required.
     365 +                   elif self.query.combinator:
     366 +                       field.expression = Ref(select_ref.refs, select_ref.source)
     367 +               yield field, select_ref is not None
341  368                 continue
342  369             if field == "?":  # random
343  370                 yield OrderBy(Random()), False
```

Added and deleted coding lines

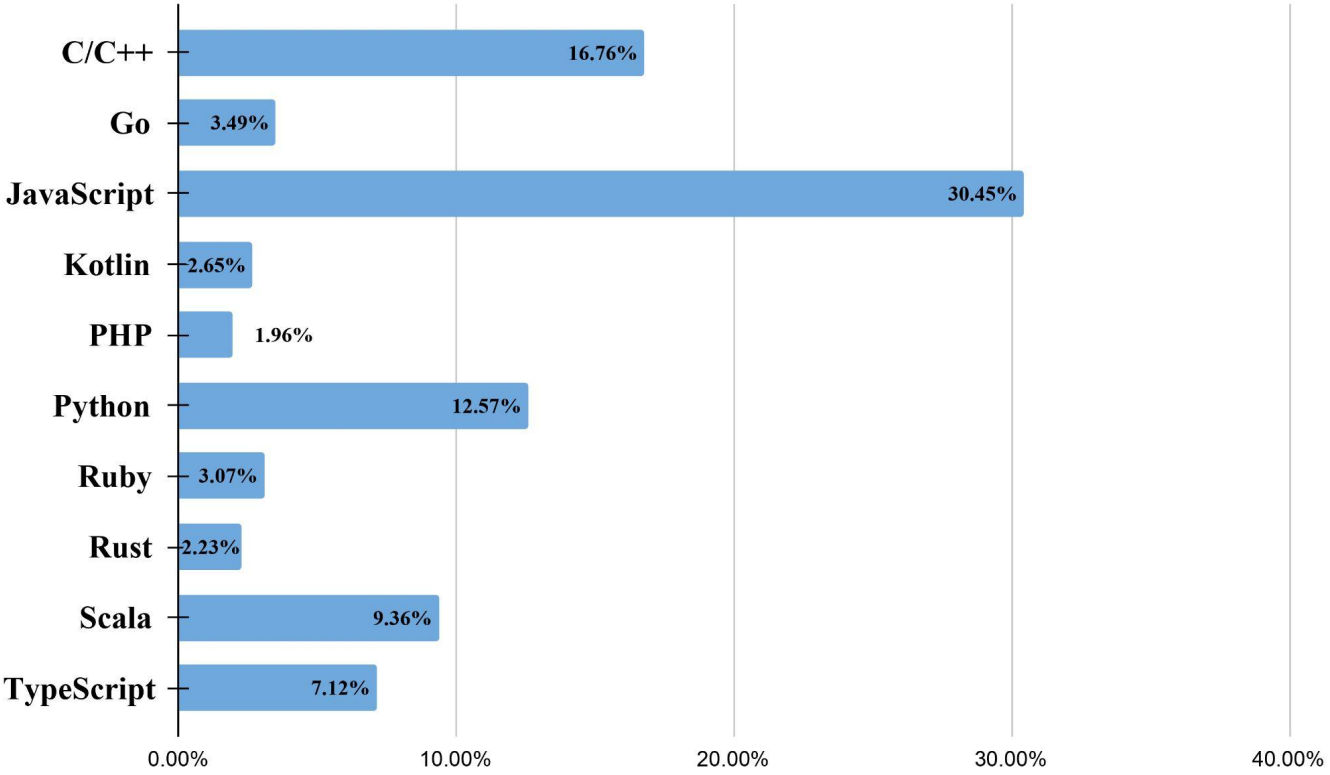
Added and deleted comment lines

Changed Files

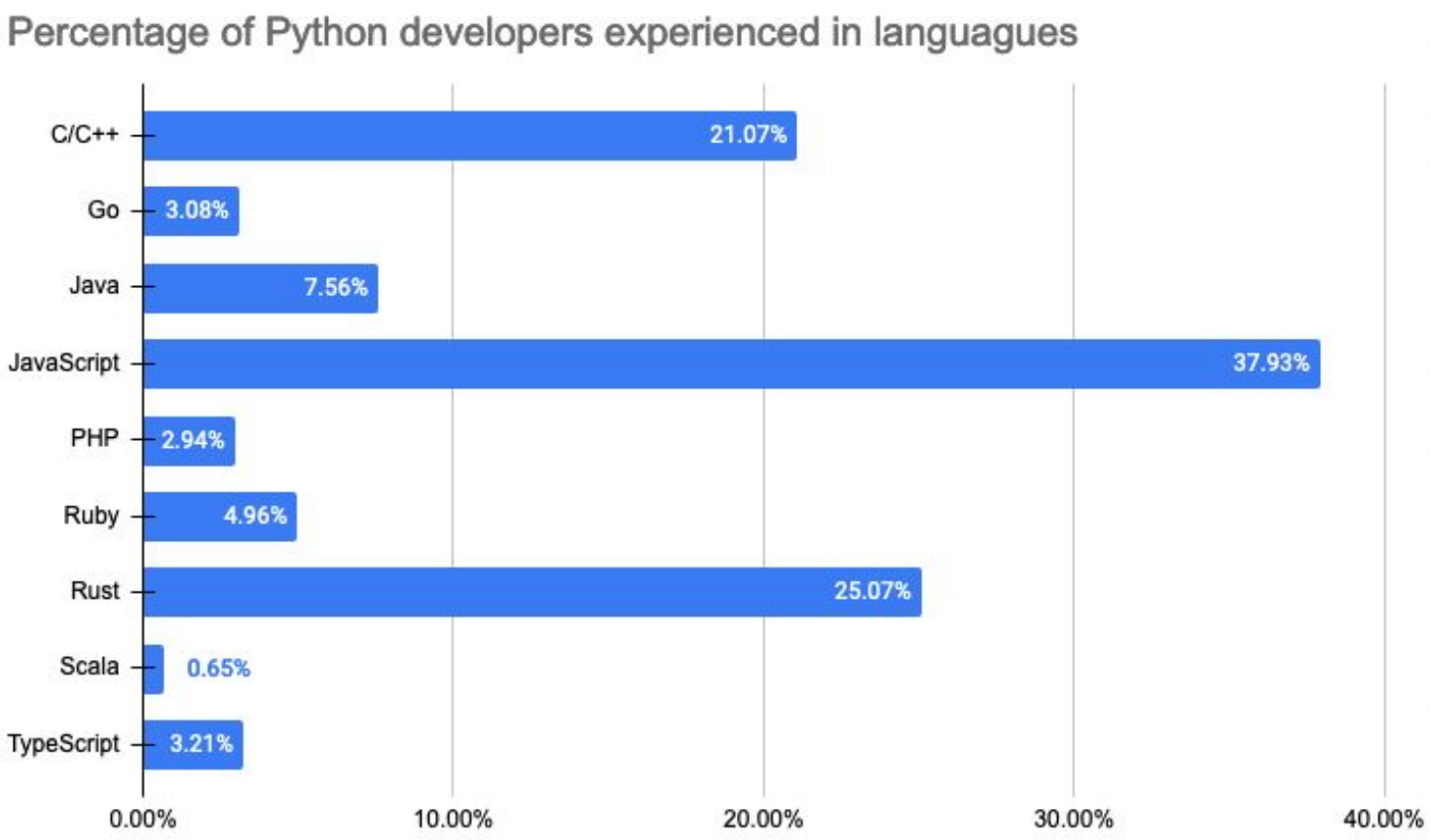
Added/deleted comment-lines



Polyglotism of Java developers



Polyglotism of Python developers



Top contributors of Java projects

	Guice	Guava	Spark	Vaadin	Eclipse	Hadoop
C/C++	0.1%	5.2%	0.1%	0.4%	5.5%	1.6%
Clojure	0.0%	0.0%	1.2%	0.0%	0.0%	0.0%
Go	0.2%	0.3%	0.5%	0.0%	0.6%	0.2%
Java	33.2%	25.0%	14.1%	30.9%	41.9%	31.6%
JavaScript	4.5%	1.6%	0.0%	6.7%	2.0%	2.7%
Kotlin	0.9%	0.7%	1.0%	0.0%	0.1%	0.0%
Other	59.0%	66.2%	4.6%	60.7%	49.0%	58.6%
Perl	0.0%	0.0%	7.5%	0.0%	0.0%	0.1%
PHP	0.2%	0.0%	0.1%	0.2%	0.1%	0.1%
Python	0.1%	0.2%	11.1%	0.1%	0.2%	1.4%
Ruby	0.1%	0.0%	5.6%	0.0%	0.2%	0.4%
Rust	0.0%	0.1%	35.3%	0.1%	0.1%	0.1%
Scala	0.4%	0.0%	2.5%	0.1%	0.0%	2.8%
Sql	0.0%	0.1%	15.3%	0.0%	0.0%	0.3%
TypeScript	1.0%	0.1%	0.9%	0.7%	0.2%	0.1%

Top contributors of Python projects

	Mailpile	Requests	Pipenv	iPython	Pandas	Django	Pytorch
Basic	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%
C/C++	9.6%	9.8%	3.2%	5.9%	6.1%	7.3%	22.0%
Erlang	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%
fml	0.0%	0.1%	0.0%	0.2%	0.1%	0.1%	0.2%
Fortran	0.0%	0.0%	0.0%	0.2%	0.4%	0.1%	0.1%
Go	8.5%	1.6%	0.7%	0.3%	0.6%	1.1%	0.8%
Java	4.8%	8.9%	0.8%	3.8%	1.0%	1.0%	1.2%
JavaScript	19.7%	10.1%	8.2%	9.9%	14.2%	11.3%	5.7%
Kotlin	0.1%	0.3%	0.0%	0.0%	0.0%	0.0%	0.0%
Lisp	0.1%	0.0%	0.1%	0.1%	0.4%	0.5%	0.2%
Other	45.9%	57.6%	70.0%	63.4%	56.8%	57.9%	52.3%
Perl	0.2%	0.0%	0.0%	0.1%	0.1%	0.3%	0.1%
PHP	4.8%	2.2%	0.5%	0.2%	0.8%	0.6%	0.2%
Python	3.1%	4.9%	12.0%	11.9%	16.6%	17.0%	13.4%
R	0.1%	0.0%	0.0%	0.1%	0.2%	0.0%	0.1%
Ruby	0.6%	1.9%	0.7%	0.3%	0.4%	0.6%	1.0%
Rust	1.1%	0.6%	1.5%	2.7%	1.6%	1.2%	1.0%
Scala	0.1%	1.0%	0.4%	0.2%	0.4%	0.1%	0.5%
Sql	0.1%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%
Swift	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%
TypeScript	1.1%	0.9%	1.6%	0.4%	0.3%	0.5%	0.4%

Heatmap for top commenters in Java

	Guice	Guava	Spark	Vaadin	Eclipse	Hadoop
C/C++	0.5%	2.2%	3.0%	0.2%	4.5%	1.2%
Go	0.0%	0.3%	1.1%	0.0%	0.9%	0.3%
Java	30.4%	30.2%	14.8%	23.4%	45.7%	29.7%
JavaScript	5.2%	2.7%	4.0%	6.1%	2.3%	1.1%
Kotlin	0.2%	0.7%	0.0%	0.0%	0.1%	0.0%
Other	63.3%	62.6%	49.6%	69.5%	45.5%	65.6%
PHP	0.0%	0.0%	0.2%	0.1%	0.2%	0.1%
Python	0.1%	0.4%	5.0%	0.0%	0.3%	0.5%
Ruby	0.0%	0.1%	0.4%	0.0%	0.0%	0.5%
Rust	0.0%	0.1%	0.4%	0.0%	0.0%	0.0%
Scala	0.0%	0.4%	19.0%	0.0%	0.0%	0.9%
TypeScript	0.1%	0.1%	0.3%	0.6%	0.3%	0.1%

Heatmap for top commenters in Python

	Mailpile	Requests	Pipenv	iPython	Pandas	Django	Pytorch
C/C++	7.5%	3.2%	5.5%	4.5%	7.0%	7.3%	18.8%
fml	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%	0.2%
Fortran	0.0%	0.0%	0.0%	0.2%	0.3%	0.1%	0.1%
Go	6.6%	3.5%	3.5%	0.1%	0.7%	1.2%	0.6%
Java	3.7%	0.1%	1.6%	1.1%	1.5%	1.5%	1.7%
JavaScript	22.7%	15.7%	15.7%	14.6%	14.3%	14.0%	6.4%
Julia	0.0%	0.0%	0.0%	0.3%	0.1%	0.0%	0.1%
Kotlin	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%
Lisp	0.1%	0.1%	0.2%	0.2%	0.3%	0.4%	0.1%
Lua	0.0%	0.1%	0.1%	0.1%	0.0%	0.0%	0.2%
other	49.3%	57.1%	48.6%	60.7%	59.5%	56.2%	59.6%
Perl	0.1%	0.1%	0.0%	0.0%	0.1%	0.3%	0.1%
PHP	3.9%	3.0%	2.4%	1.1%	0.7%	1.1%	0.3%
Python	3.2%	12.9%	14.7%	13.3%	11.7%	14.4%	8.5%
R	0.1%	0.0%	0.0%	0.2%	0.2%	0.0%	0.1%
Ruby	0.7%	0.5%	1.3%	0.6%	1.0%	0.8%	0.9%
Rust	1.0%	1.7%	3.8%	2.0%	1.3%	1.2%	0.8%
Scala	0.1%	0.5%	1.0%	0.0%	0.4%	0.1%	0.4%
Sql	0.1%	0.3%	0.0%	0.0%	0.1%	0.1%	0.0%
Swift	0.0%	0.1%	0.1%	0.0%	0.0%	0.1%	0.3%
TypeScript	0.9%	0.9%	0.7%	0.9%	0.5%	0.5%	0.3%

Conclusion and future work

Python developers tend to be more polyglot

Conclusion and future work

Python developers tend to be more polyglot

Next steps:

- ❏ Choosing the language to compare

Conclusion and future work

Python developers tend to be more polyglot

Next steps:

- ❏ Choosing the language to compare
- ❏ Extracting comments

Conclusion and future work

Python developers tend to be more polyglot

Next steps:

- ❏ Choosing the language to compare
- ❏ Extracting comments
- ❏ Analyze developer commenting practices