Understanding Performance Concerns in the API Documentation of Data Science Libraries

Yida Tao
Shenzhen University
Guangdong, China

Jiefang Jiang
Shenzhen University
Guangdong, China

Yepang Liu
Southern University of Science and Technology
Guangdong, China

Zhiwu Xu
Shenzhen University
Guangdong, China

Shengchao Qin
Teesside University, UK
Shenzhen University, China

ASE. September 21–25, 2020, Virtual Event, Australia

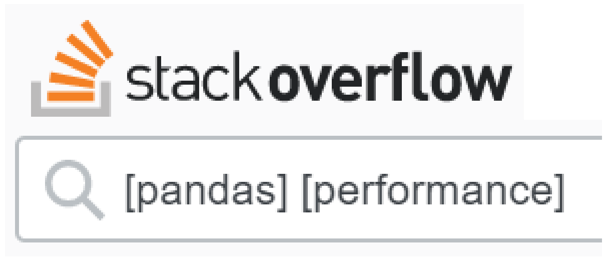
Motivation

- Data science is one of the most exciting emerging fields
- Performance issues are major bottlenecks for developing efficient data science applications
- The performance of popular data science libraries (e.g., pandas, numpy) is also vital for improving application efficiency and developer productivity

Painfully slow execution time and rapid memory exhaustion

Motivation

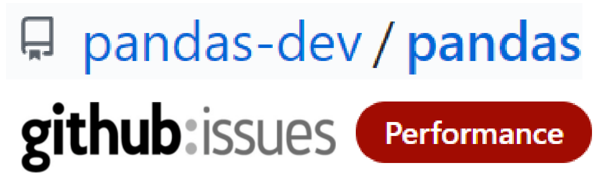
- Persistent and active discussions on the performance problems of data science libraries are observed
- Developers often suffer from long, recurring interruptions caused by performance problems



[pandas] [performance]

853 questions

58% answer acceptance



pandas-dev / pandas

Performance

1,113 issues

134 days max resolution time

Documentation to the Rescue?

RQ1. (**Prevalence**) How common are data science APIs documented in performance-related context?

RQ2. (**Knowledge**) What types of knowledge are provided by performance-related documentation?

RQ3. (**Consistency**) What are the difference between the official and crowd documentation in terms of performance-related content?

RQ4. (**Evolution**) How does performance-related documentation evolve over time?

Approach Overview

01
Performance Concerns Extraction

Extracting performance-related sentences from the documentation

02
Knowledge Classification

Knowledge types of performance-related documentation

03
Consistency Analysis

Consistency between the official and crowd documentation

04
Evolution Analysis

Evolution patterns of performance-related documentation

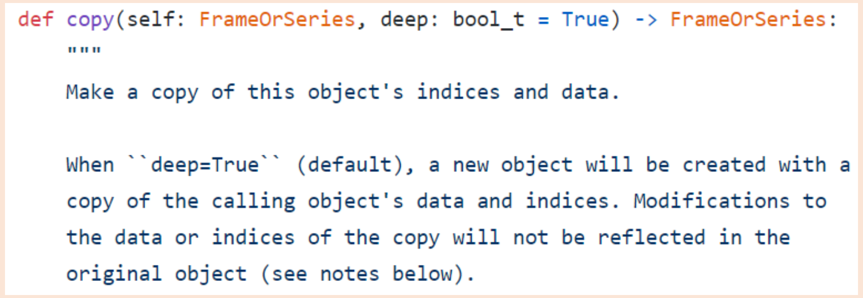
Data Collection

Libraries

- NumPy
- Pandas
- SciPy
- Scikit-learn
- TensorFlow
- Gensim

Official Documentation

- API docstring



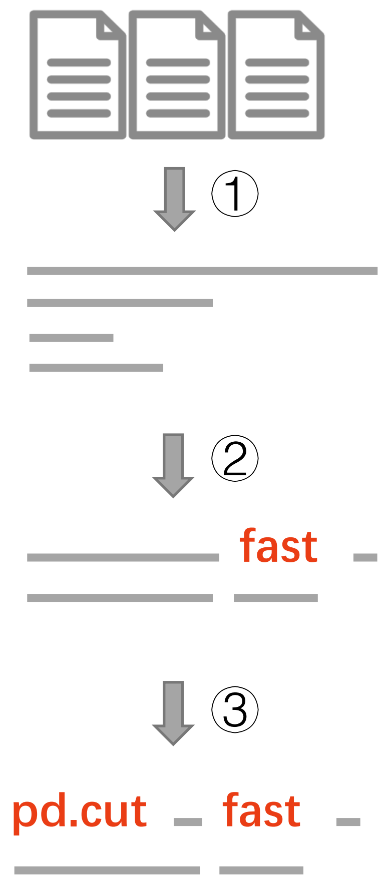
```
def copy(self: FrameOrSeries, deep: bool_t = True) -> FrameOrSeries:
    """
    Make a copy of this object's indices and data.

    When ``deep=True`` (default), a new object will be created with a
    copy of the calling object's data and indices. Modifications to
    the data or indices of the copy will not be reflected in the
    original object (see notes below).
```

- User guide
  - Markdown (.md)
  - reStructuredText (.rst)
  - Jupyter notebook (.ipynb)

Crowd Documentation

- Stack Overflow
  - Threads of the target libraries
- GitHub issues
  - Threads of the target libraries

I. Extracting Performance-related Documentation



① Sentence Segmentation

② Matching performance-related keywords

- *Fast*, *slow*, *expensive*, *performance*, *speedup*, *efficient*, etc.
- Inflections of the above keywords (e.g., *efficiency*)

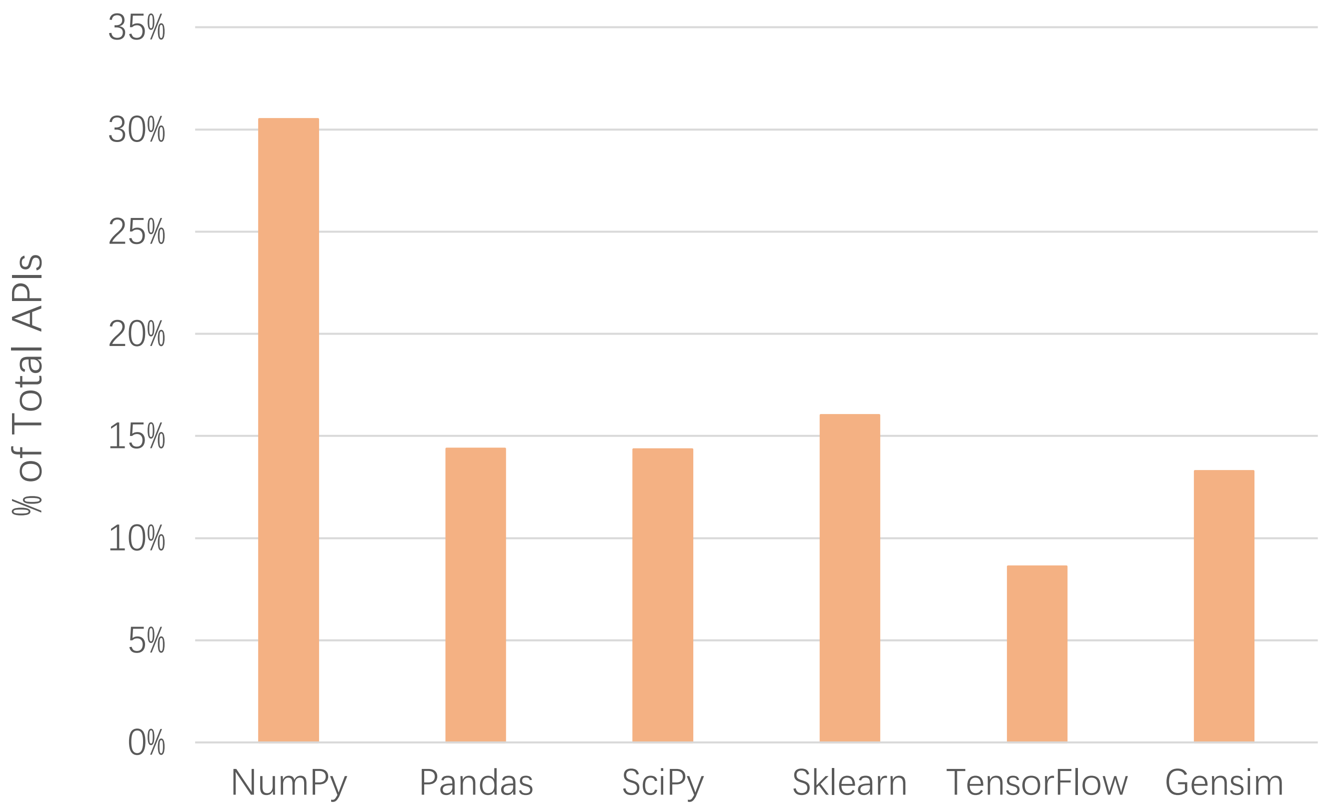
③ Identifying the APIs discussed in each sentence

- Use *declarations*, *hyperlinks*, and *regular expressions* to identify code entities in natural-language sentences
- Use *AST parsing* and *naming* heuristics to resolve APIs

④ Manual validation

- Whether the sentence truly discuss performance concerns
- Whether the API resolution is correct and complete

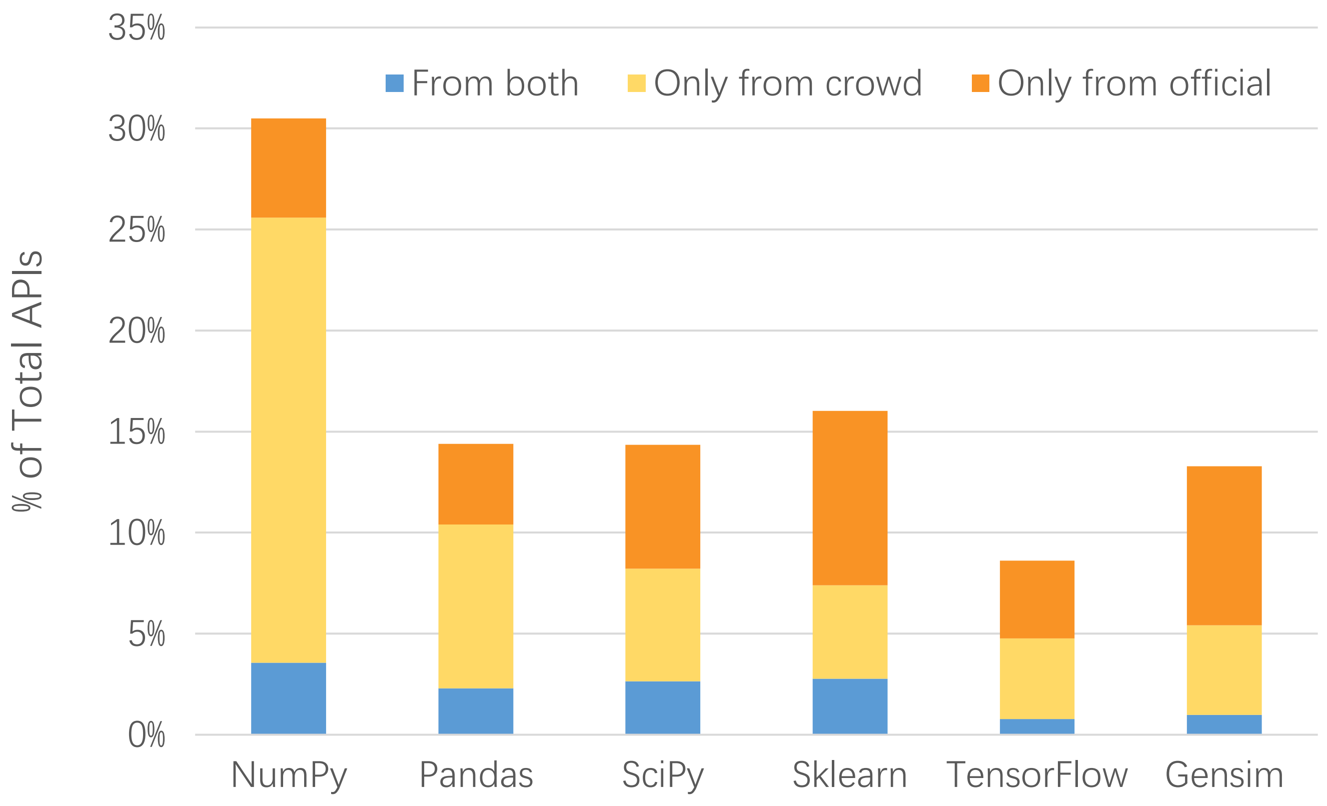
Prevalence



All libraries have nontrivial proportion of APIs being documented in performance-related context.

- NumPy has **30%** of its APIs being documented in performance-related context
- Other libraries have **10%-15%** of such APIs

Prevalence



Performance concerns from official documentation and crowd documentation cover a different set of APIs

II. Knowledge Classification

- Maalej and Robillard proposed 12 knowledge types for general API documentation [1]

- We conducted *inductive coding* to adjust the taxonomy to performance-specific documentation
  - 11 knowledges types: 6 from [1] and 5 are newly emerged

[1] Patterns of knowledge in API reference documentation. Walid Maalej and Martin P. Robillard. TSE. 2013

Knowledge Type

Functionality
"The `Series.align` method is the fastest way to simultaneously align two objects"
(`pandas.Series.align`)

Alternatives
"Mini-batch sparse PCA `MiniBatchSparsePCA` is a variant of `SparsePCA` that is faster but less accurate"
(`sklearn.decomposition.MiniBatchSparsePCA`)

Usage Practice
"To construct a matrix efficiently, make sure the items are pre-sorted by index, per row"
(`scipy.sparse.lil_matrix`)

Knowledge Type

- **Functionality** is the most common knowledge type (33%) in API docstrings, yet is less discussed in crowd documentation (8%)

- **Alternatives** (27%) and **Usage Practice** (25%) types of knowledge are more prevalent in crowd documentation

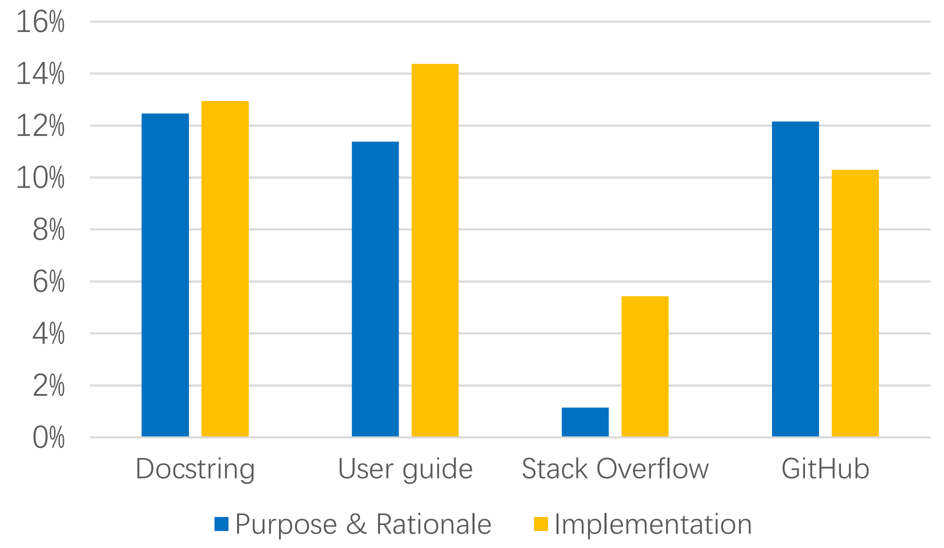
Functionality
"The `Series.align` method is the fastest way to simultaneously align two objects" (`pandas.Series.align`)

Alternatives
"Mini-batch sparse PCA `MiniBatchSparsePCA` is a variant of `SparsePCA` that is faster but less accurate" (`sklearn.decomposition.MiniBatchSparsePCA`)

Usage Practice
"To construct a matrix efficiently, make sure the items are pre-sorted by index, per row" (`scipy.sparse.lil_matrix`)

Knowledge Type

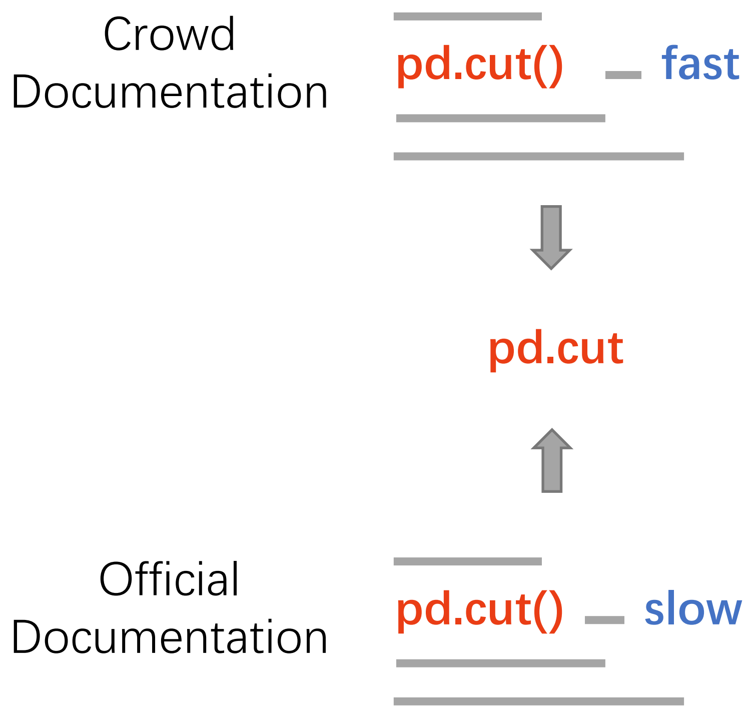
- Stack Overflow rarely provides explanatory knowledge types such as **Implementation** (5%) and **Purpose & Rationale** (1%)



Implementation
"Internally this version uses a much faster implementation that never constructs the indices and uses simple slicing. " `(numpy.fill_diagonal)`

Purpose & Rationale
"The vectorize function is provided primarily for convenience, not for performance."
`(numpy.vectorize)`

III. Consistency Analysis



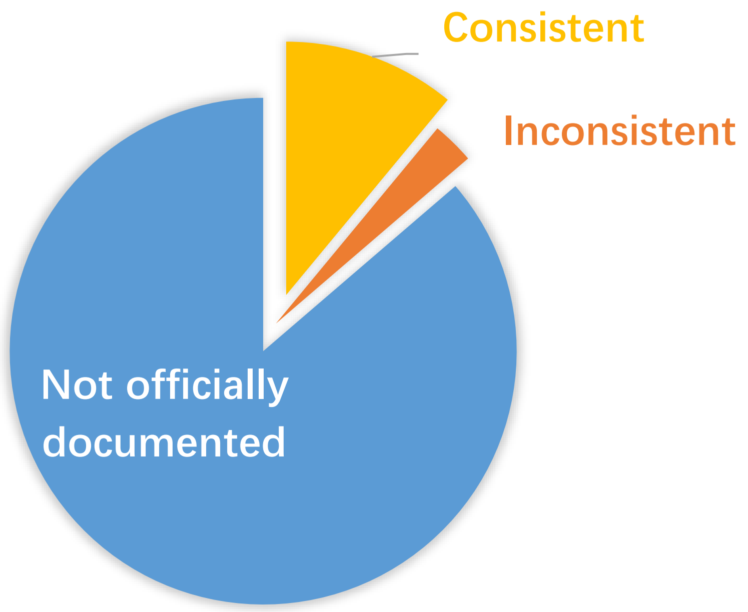
- Associate performance concerns from the crowd doc with the official doc that discuss the same subject APIs

- Classify each performance concern from crowd doc as
  - Consistent
  - Inconsistent
  - Not officially documented

Information Consistency

86% performance concerns from the crowd doc have not been found in the official doc

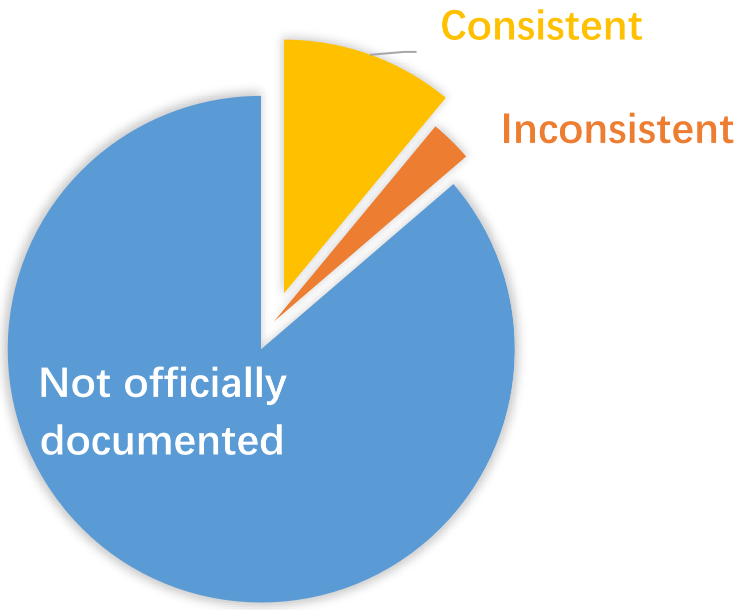
- Crowd documentation offer a large volume of new information on the performance of data science libraries.



Information Consistency

86% performance concerns from the crowd doc have not been found in the official doc

- Crowd documentation offer a large volume of new information on the performance of data science libraries.



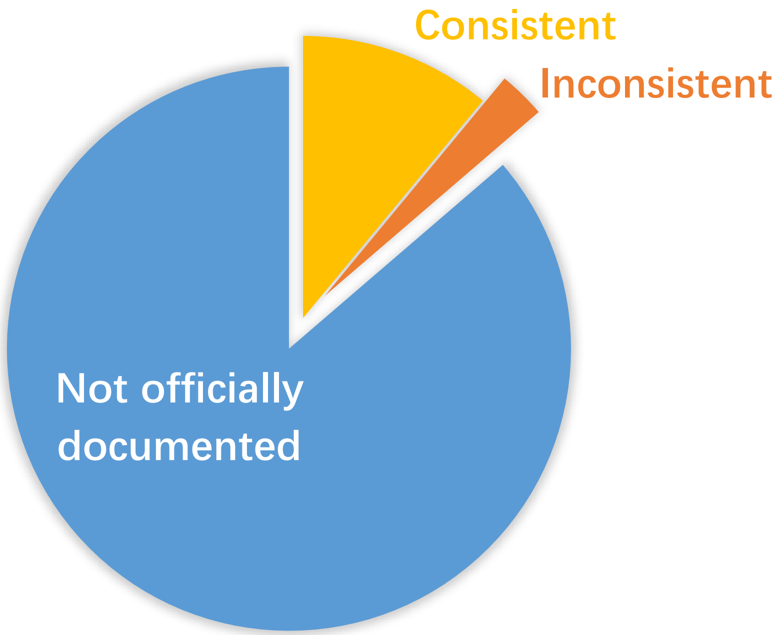
11% performance concerns from the crowd doc are **consistent** with official documentation

GitHub
"`pd.eval('x // y', engine='python')` is 1000 times slower than the same operation in actual Python"

User guide
"`pandas.eval` is many orders of magnitude slower for smaller expressions/objects than plain ol' Python"

Information Consistency

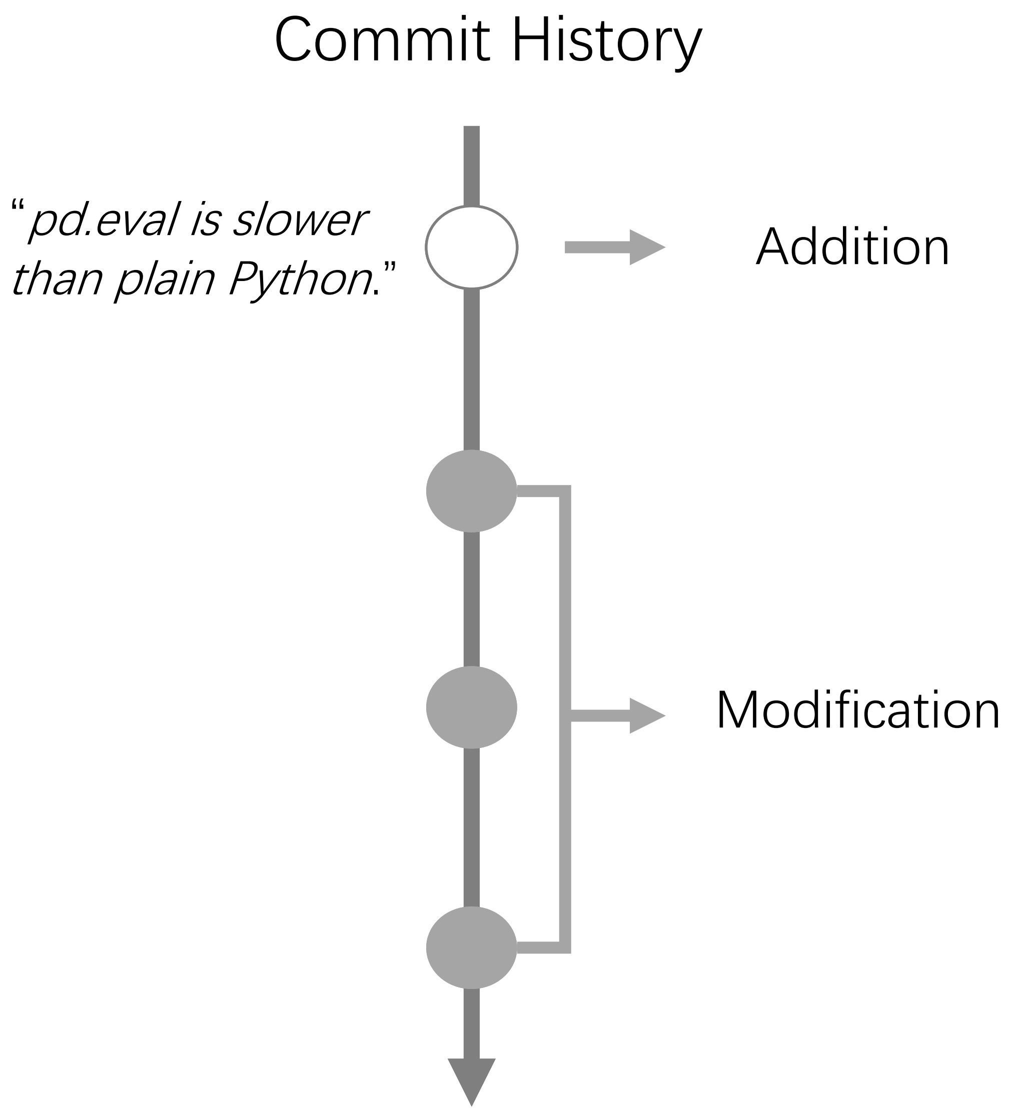
3% performance concerns from the crowd doc are **inconsistent** with official doc



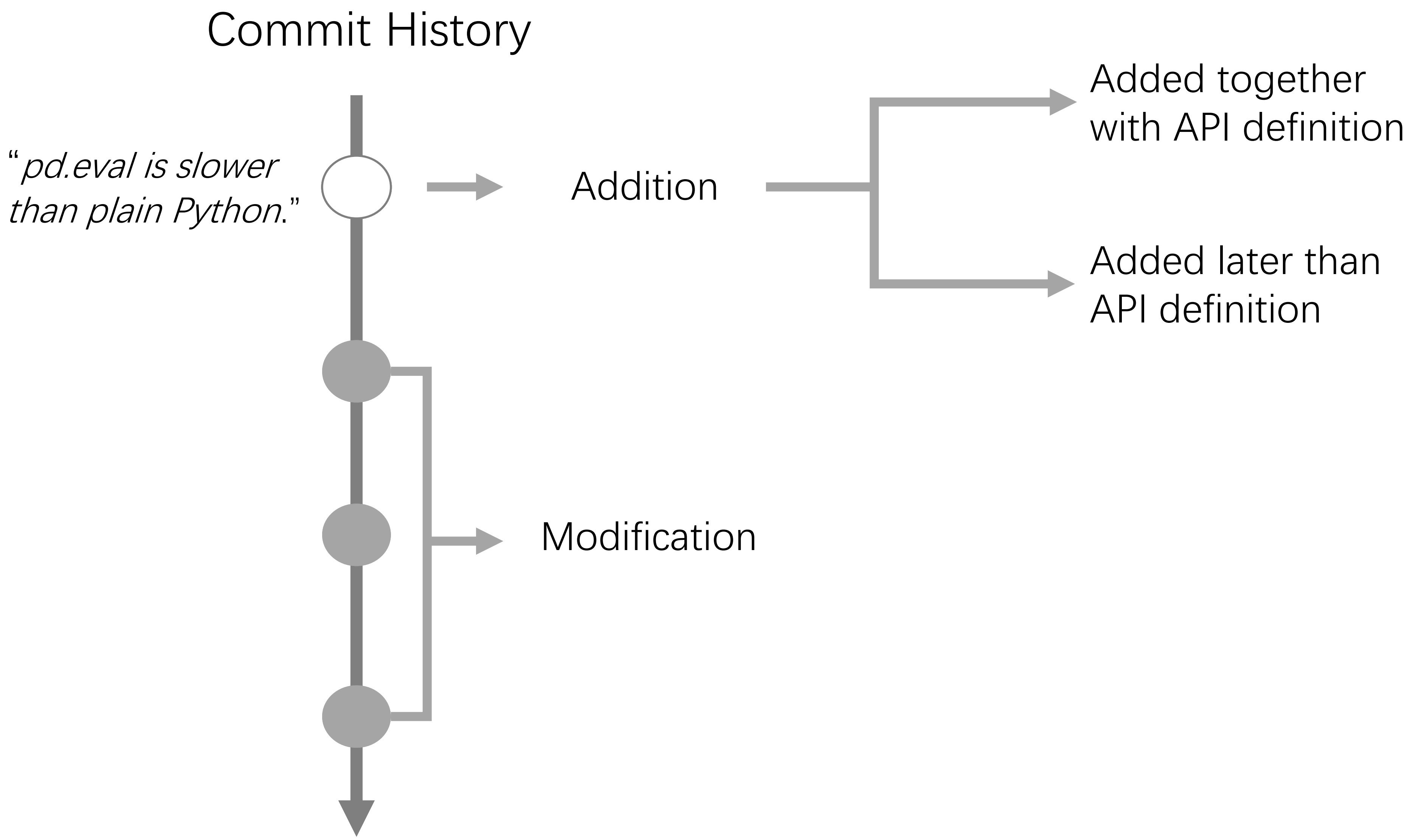
Stack Overflow
"The great thing about `CountVectorizer` is that . . . , which makes it very memory efficient, and should be able to solve any memory problems you're having."

User guide
"Have a look at the `Hashing Vectorizer` as a memory efficient alternative to `CountVectorizer`."

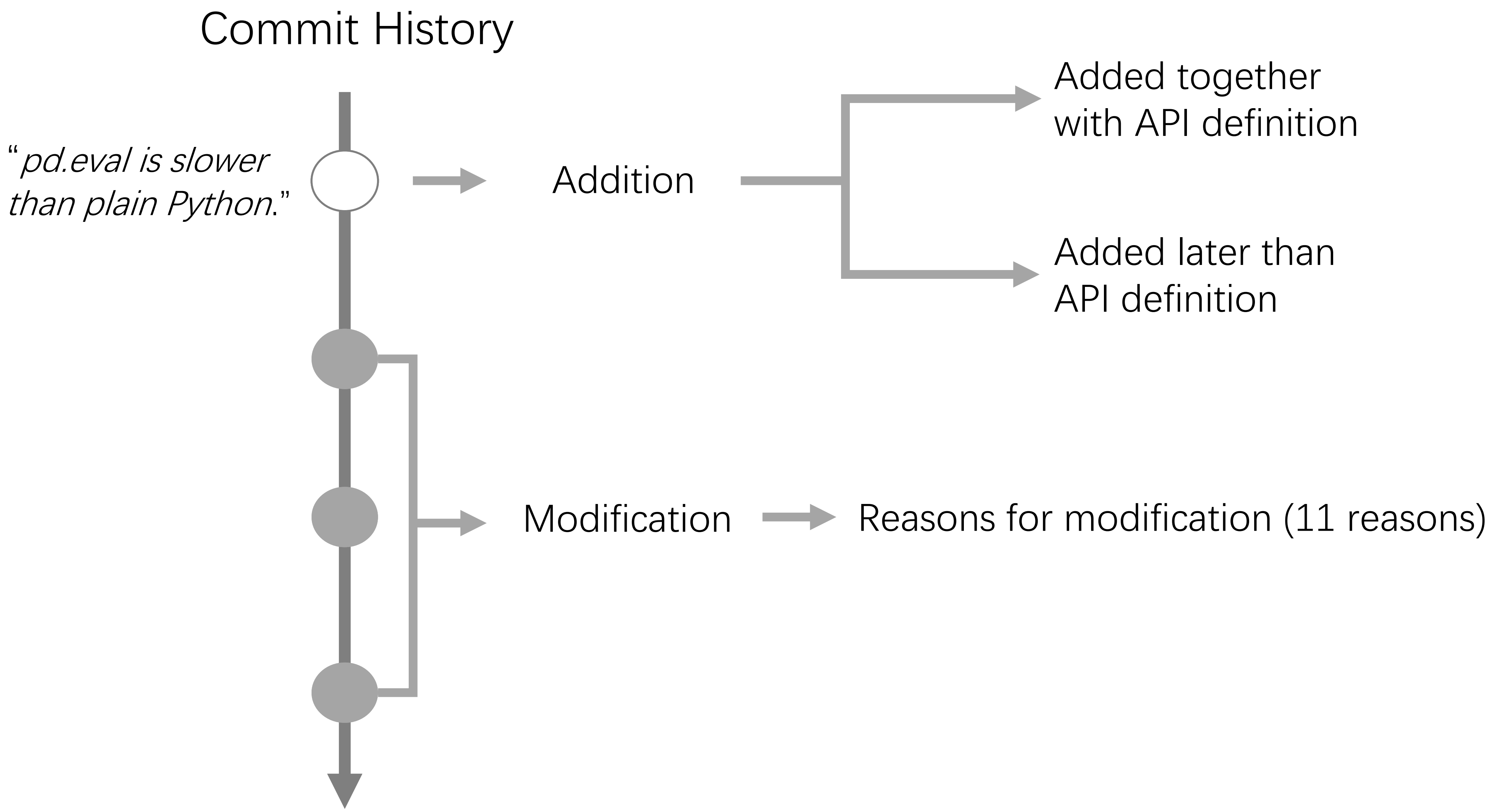
IV. Evolution Analysis



IV. Evolution Analysis


Commit History
"*pd.eval is slower than plain Python*."
Addition
Added together with API definition
Added later than API definition
Modification

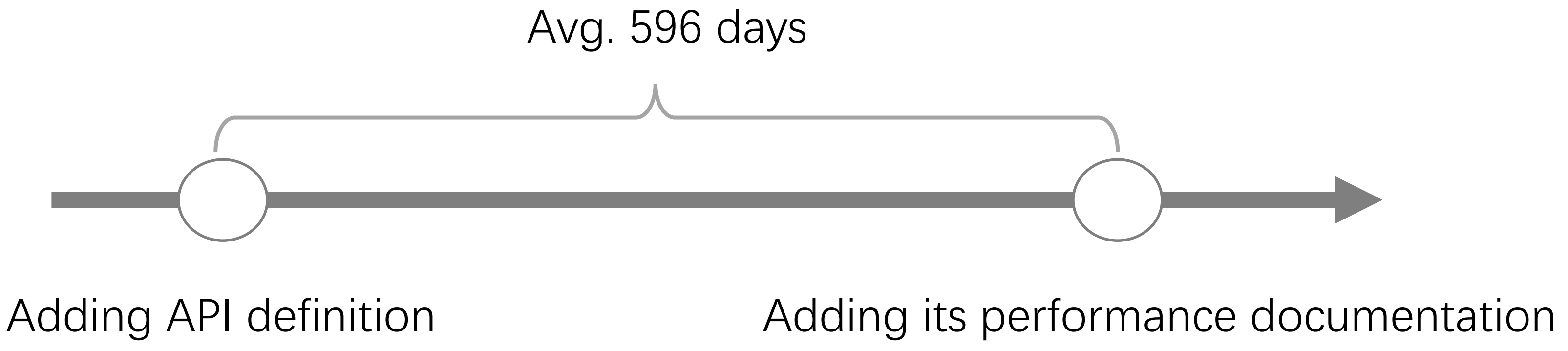
IV. Evolution Analysis


Commit History
"pd.eval is slower than plain Python."
Addition
Added together with API definition
Added later than API definition
Modification
Reasons for modification (11 reasons)

Evolution of Performance-related Doc

Developers tend to document performance concerns long after the addition of the subject API

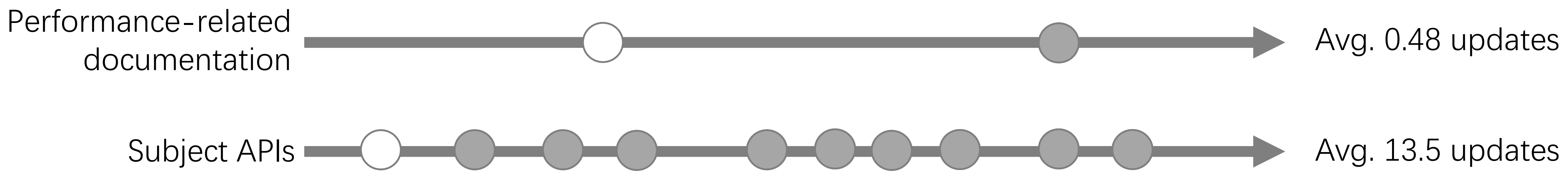
- 60.1% performance concerns are added later than the API definition
- Avg. 596 days between the addition of API definition and the addition of its performance concerns



Evolution of Performance-related Doc

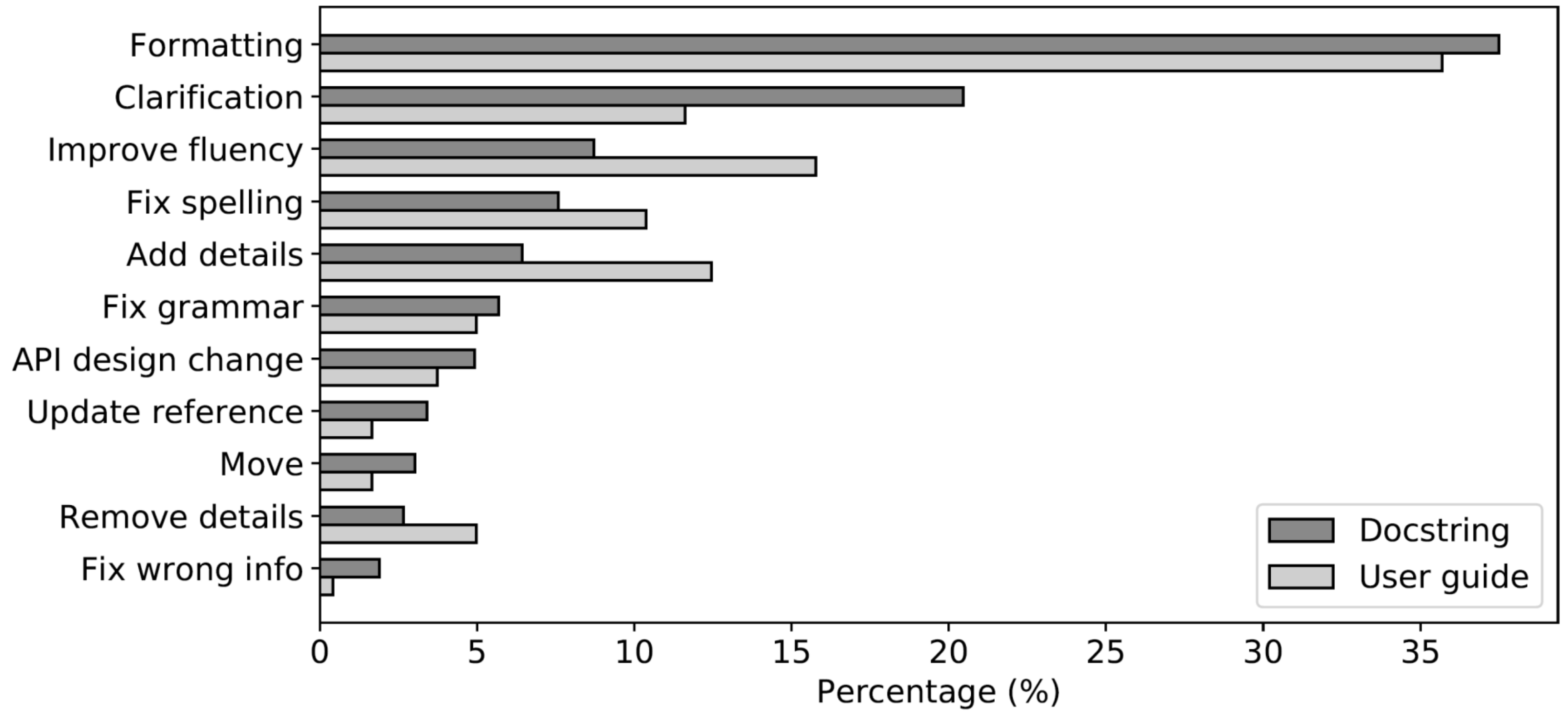
Performance concerns are not updated often, whereas their subject APIs have been updated much more frequently

- 73.1% performance concerns have been stayed the same since they were added
- 19.6% performance concerns have been updated just once



Evolution of Performance-related Doc

Developers typically apply trivial updates on performance-related documentation, without major semantic changes



Takeaways

- A nontrivial proportion of data science APIs was documented in performance-related context
- Crowd documentation is highly complementary to official documentation in terms of API coverage, knowledge types, and the specific information provided in performance-related context
- The maintenance on performance-related documentation is relatively plateauing and peripheral given the active evolution of the subject APIs
- The quality of performance-related documentation might be improved by leveraging the unofficial performance information from crowd platforms and monitoring the rarely-updated performance information from the official documentation