Can Stack Overflow Posts **Capture** Library Features?

The 19th Belgium-Netherlands Software Evolution Workshop

Camilo Velázquez-Rodríguez, Eleni Constantinou and Coen De Roover

December 4th, 2020

VRIJE UNIVERSITEIT BRUSSEL

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

SECO-Assist

@cvelazquezr

Introduction

Software products that evolve together in the same environment [1]

[1] - M. Lungu, “Towards reverse engineering software ecosystems,” in *2008 ICSM*. IEEE, 2008, pp. 428–431.

Introduction

Software products that evolve together in the same environment [1]

[1] - M. Lungu, “Towards reverse engineering software ecosystems,” in *2008 ICSM*. IEEE, 2008, pp. 428–431.

Introduction

Software products that evolve together in the same environment [1]

I/O Utilities (17)

HTTP Clients (81)

Dependency Injection (51)

Mocking (41)

XML Processing (34)

JSON Libraries (74)

Collections (72)

Reflection Libraries (47)

Command Line Parsers (45)

⋮

[1] - M. Lungu, “Towards reverse engineering software ecosystems,” in *2008 ICSM*. IEEE, 2008, pp. 428–431.

Introduction

Software products that evolve together in the same environment [1]

I/O Utilities (17)

HTTP Clients (81)

Dependency Injection (51)

Mocking (41)

XML Processing (34)

JSON Libraries (74)

Collections (72)

Reflection Libraries (47)

Command Line Parsers (45)

⋮

JCommander Library

Version	Repository	Usages	Date
1.78	Central	20	Aug, 2019
1.77	Central	0	Aug, 2019
1.75	Central	3	Aug, 2019
1.72	Central	193	Jun, 2017
1.69	Central	18	Apr, 2017
1.64	Central	23	Mar, 2017
1.60	Central	30	Jan, 2017
1.58	Central	21	Sep, 2016
1.48	Central	259	Apr, 2015
1.47	Central	24	Dec, 2014
1.35	Central	123	Feb, 2014
1.32	Central	55	Sep, 2013
1.30	Central	94	Oct, 2012

[1] - M. Lungu, “Towards reverse engineering software ecosystems,” in *2008 ICSM*. IEEE, 2008, pp. 428–431.

Introduction

Software products that evolve together in the same environment [1]

Library Selection Issues

I/O Utilities (17)

HTTP Clients (81)

Dependency Injection (51)

Mocking (41)

XML Processing (34)

JSON Libraries (74)

Collections (72)

Reflection Libraries (47)

Command Line Parsers (45)

⋮

JCommander Library

Version	Repository	Usages	Date
1.78	Central	20	Aug, 2019
1.77	Central	0	Aug, 2019
1.75	Central	3	Aug, 2019
1.72	Central	193	Jun, 2017
1.69	Central	18	Apr, 2017
1.64	Central	23	Mar, 2017
1.60	Central	30	Jan, 2017
1.58	Central	21	Sep, 2016
1.48	Central	259	Apr, 2015
1.47	Central	24	Dec, 2014
1.35	Central	123	Feb, 2014
1.32	Central	55	Sep, 2013
1.30	Central	94	Oct, 2012

[1] - M. Lungu, “Towards reverse engineering software ecosystems,” in *2008 ICSM*. IEEE, 2008, pp. 428–431.

Introduction

Guava

> But trawling through Javadoc isn't always the most effective way to learn how to make best use of a library. Here, we try to provide readable and pleasant explanations of **some of the most popular and most powerful features of Guava**.

https://github.com/google/guava/wiki

Introduction

Guava

But trawling through Javadoc isn't always the most effective way to learn how to make best use of a library. Here, we try to provide readable and pleasant explanations of **some of the most popular and most powerful features of Guava**.

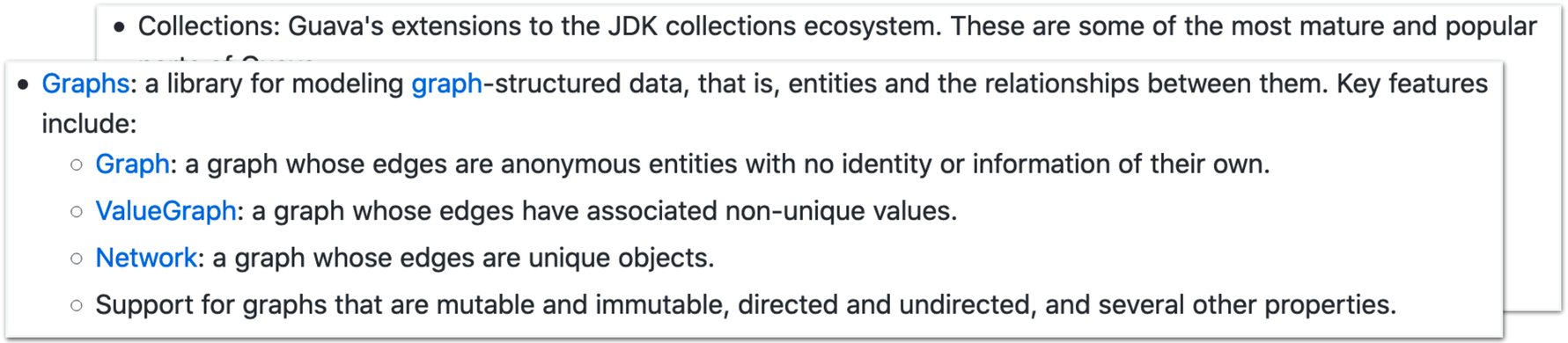
- Collections: Guava's extensions to the JDK collections ecosystem. These are some of the most mature and popular parts of Guava.
  - Immutable collections, for defensive programming, constant collections, and improved efficiency.
  - New collection types, for use cases that the JDK collections don't address as well as they could: multisets, multimaps, tables, bidirectional maps, and more.
  - Powerful collection utilities, for common operations not provided in `java.util.Collections`.
  - Extension utilities: writing a `Collection` decorator? Implementing `Iterator`? We can make that easier.

https://github.com/google/guava/wiki

Introduction

Guava

But trawling through Javadoc isn't always the most effective way to learn how to make best use of a library. Here, we try to provide readable and pleasant explanations of **some of the most popular and most powerful features of Guava**.

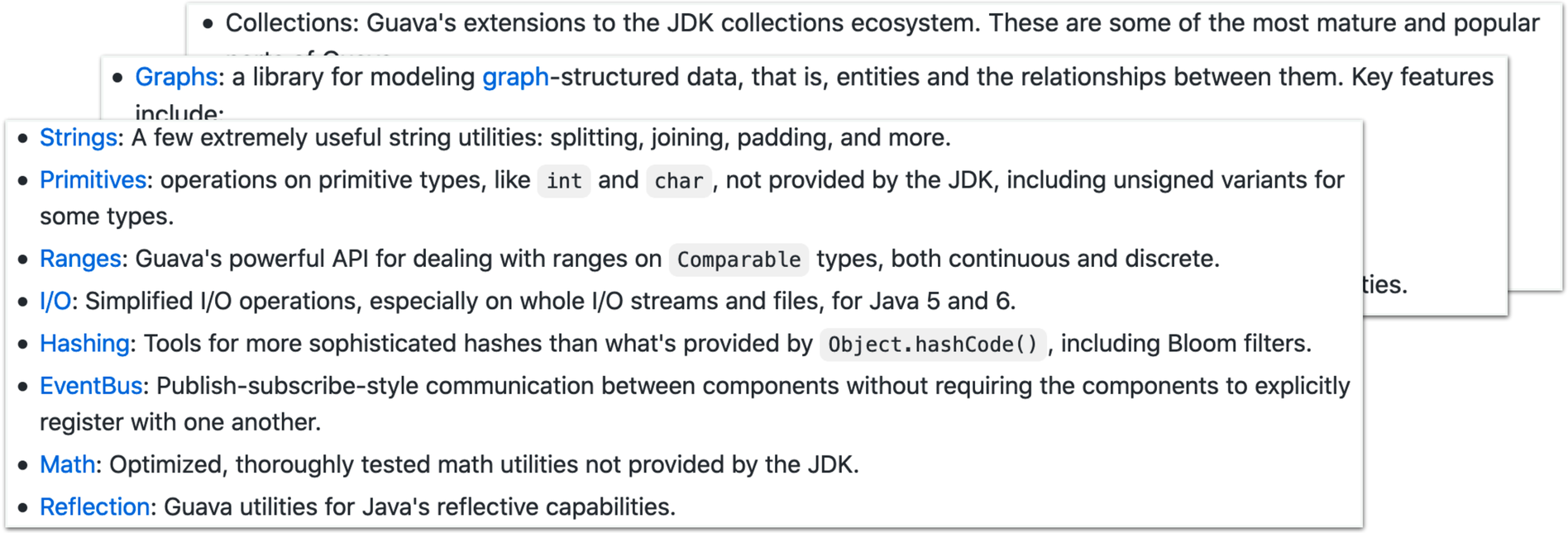


https://github.com/google/guava/wiki

Introduction

Guava

But trawling through Javadoc isn't always the most effective way to learn how to make best use of a library. Here, we try to provide readable and pleasant explanations of **some of the most popular and most powerful features of Guava**.



https://github.com/google/guava/wiki

Introduction

Stack Overflow Q&A

How do I convert a String to an int in Java?

Guava equivalent for IOUtils.toString(InputStream)

How do I compare strings in Java?

Remove last character of a StringBuilder?

Guava: Splitter and considering Escaping?

String utilities

initializing a Guava ImmutableMap

Flattening an Iterable<Iterable<T>> in Guava

How to directly initialize a HashMap (in a literal way)?

Google Guava isNullOrEmpty for collections

Google Guava “zip” two lists

Collection utilities

Merge ranges with guava

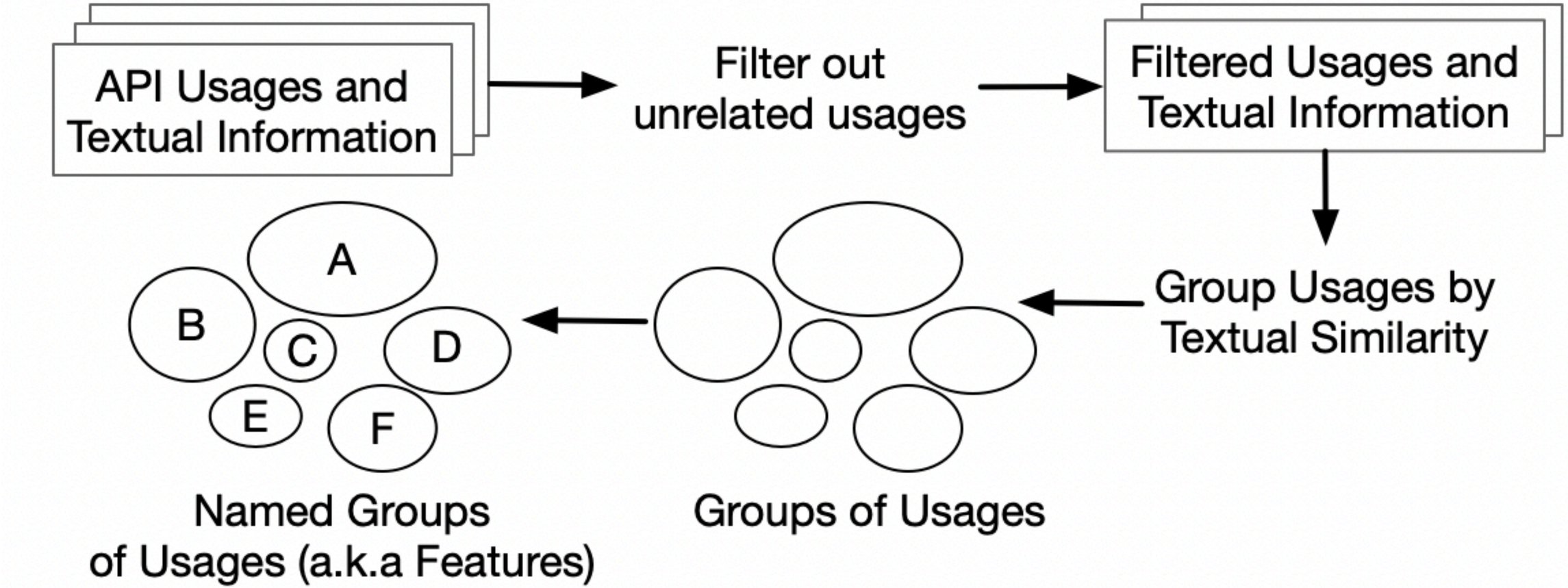
Ranges

What is the simplest way to read a file into String?

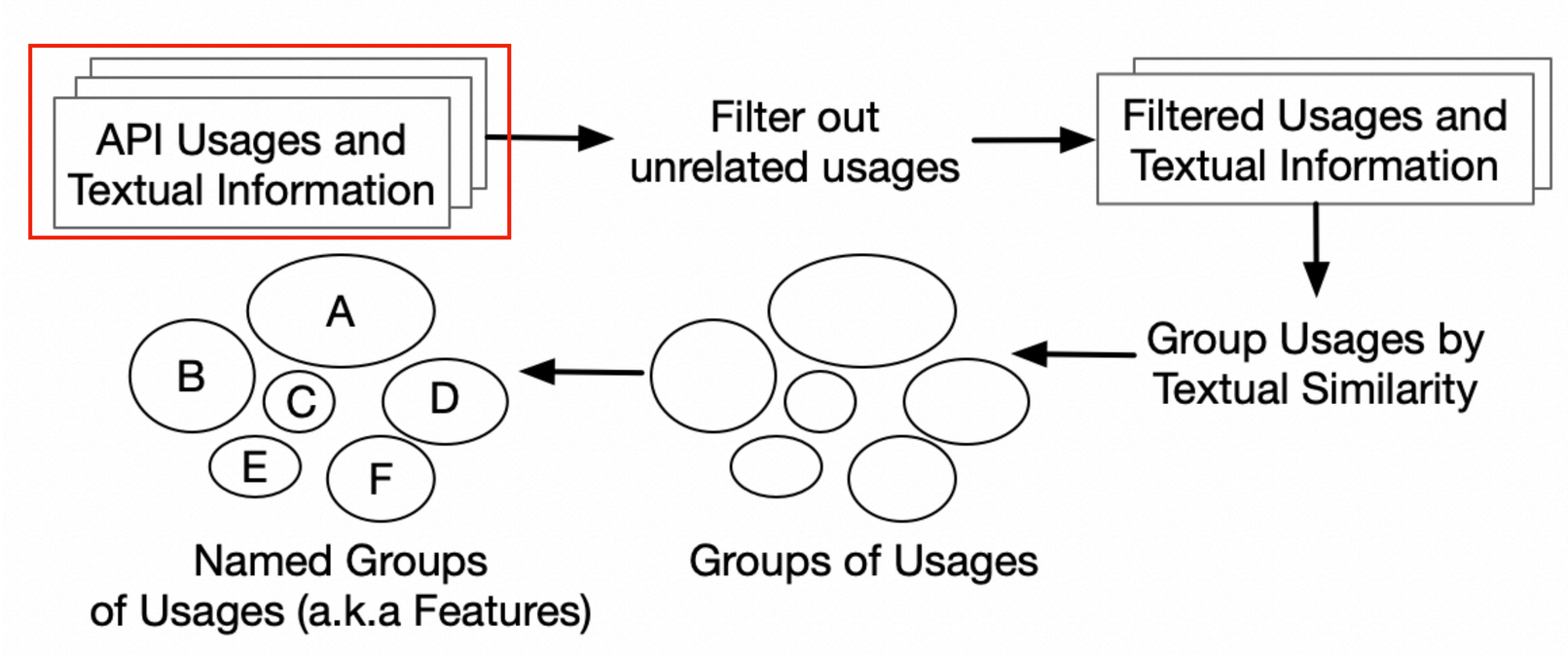
Utils to read resource text file to String (Java)

I/O

Pipeline


API Usages and Textual Information
Filter out unrelated usages
Filtered Usages and Textual Information
Group Usages by Textual Similarity
A
B
C
D
E
F
Named Groups of Usages (a.k.a Features)
Groups of Usages

Extracting Usages

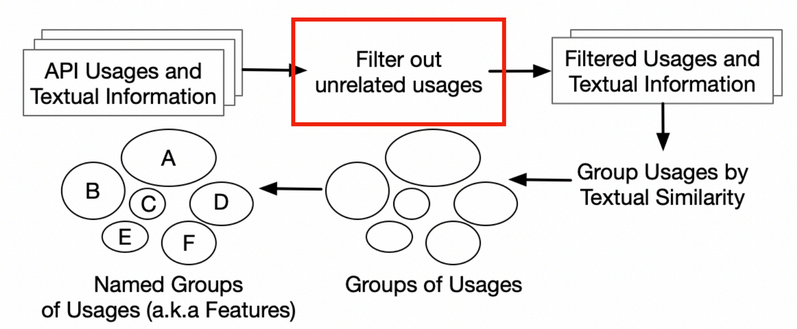



```
Gson gson = new Gson();
PrintWriter out = new PrintWriter(;
String json = gson.toJson("{}")
out.print(json)
out.close();
```

```
[com.google.Gson: 0.91
 org.apache.http.NameValuePair: 0.04]
Gson gson = new Gson();
[java.io.PrintStream: 0.4
 java.io.BufferedReader: 0.4]
PrintWriter out = new PrintWriter(;
[com.google.Gson: 0.98
 org.apache.http.NameValuePair: 0.01]
String json = gson.toJson("{}")
```

```
[java.io.PrintStream: 0.95
 java.io.BufferedReader: 0.04]
out.print(json)
[java.io.PrintStream: 0.96
 java.io.BufferedReader: 0.03]
out.close();
```

Filtering



Compute Covariance Matrix in Java

Here is a short example, how you can create it with Apache Commons Math (3.5):

10

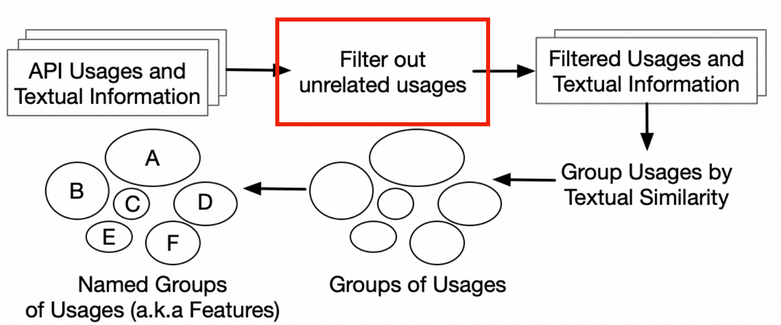
```
RealMatrix mx = MatrixUtils.createRealMatrix(new double[][]{
  {1, 2, 3},
  {2, 4, 6}
});
RealMatrix cov = new Covariance(mx).getCovarianceMatrix();
```

How to average BigDecimals using Streams?

3

```
MutableDoubleList doubles = DoubleLists.mutable.with(1.0, 2.0, 3.0, 4.0);
List<BigDecimal> bigDecimals = doubles.collect(BigDecimal::new);
BigDecimal average =
        bigDecimals.stream()
                .collect(Collectors2.summarizingBigDecimal(e -> e))
                .getAverage(MathContext.DECIMAL32);

Assert.assertEquals(BigDecimal.valueOf(2.5), average);
```

Filtering



Compute Covariance Matrix in Java

10

Here is a short example, how you can create it with Apache Commons Math (3.5):

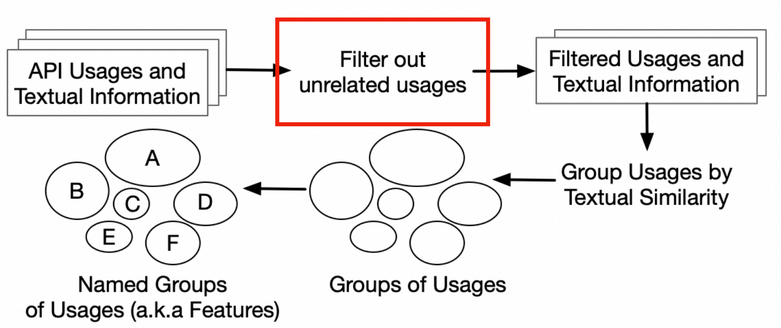
```
RealMatrix mx = MatrixUtils.createRealMatrix(new double[][]{
   {1, 2, 3},
   {2, 4, 6}
});
RealMatrix cov = new Covariance(mx).getCovarianceMatrix();
```

How to average BigDecimals using Streams?

3

```
MutableDoubleList doubles = DoubleLists.mutable.with(1.0, 2.0, 3.0, 4.0);
List<BigDecimal> bigDecimals = doubles.collect(BigDecimal::new);
BigDecimal average =
        bigDecimals.stream()
                .collect(Collectors2.summarizingBigDecimal(e -> e))
                .getAverage(MathContext.DECIMAL32);

Assert.assertEquals(BigDecimal.valueOf(2.5), average);
```

Filtering



1. Process post titles.
2. Process API usage names.
3. Load prebuilt embeddings model.
4. Compute semantic similarity.

Compute Covariance Matrix in Java

10

Here is a short example, how you can create it with Apache Commons Math (3.5):

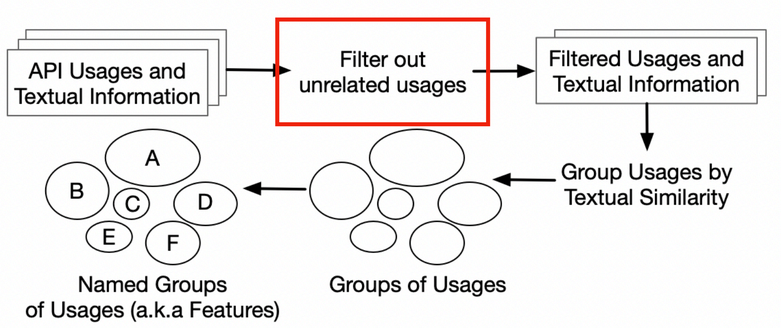
```
RealMatrix mx = MatrixUtils.createRealMatrix(new double[][]{
   {1, 2, 3},
   {2, 4, 6}
});
RealMatrix cov = new Covariance(mx).getCovarianceMatrix();
```

How to average BigDecimals using Streams?

3

```
MutableDoubleList doubles = DoubleLists.mutable.with(1.0, 2.0, 3.0, 4.0);
List<BigDecimal> bigDecimals = doubles.collect(BigDecimal::new);
BigDecimal average =
        bigDecimals.stream()
                .collect(Collectors2.summarizingBigDecimal(e -> e))
                .getAverage(MathContext.DECIMAL32);

Assert.assertEquals(BigDecimal.valueOf(2.5), average);
```

Filtering



1. Process post titles.
2. Process API usage names.
3. Load prebuilt embeddings model.
4. Compute semantic similarity.

0.86

10

Compute Covariance Matrix in Java

Here is a short example, how you can create it with Apache Commons Math (3.5):

```
RealMatrix mx = MatrixUtils.createRealMatrix(new double[][]{
   {1, 2, 3},
   {2, 4, 6}
});
RealMatrix cov = new Covariance(mx).getCovarianceMatrix();
```

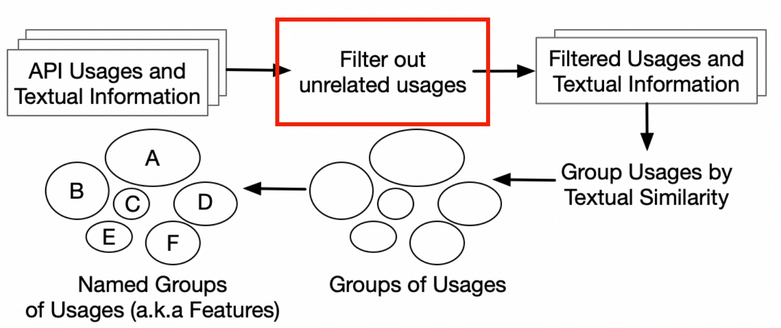
0.71

3

How to average BigDecimals using Streams?

```
MutableDoubleList doubles = DoubleLists.mutable.with(1.0, 2.0, 3.0, 4.0);
List<BigDecimal> bigDecimals = doubles.collect(BigDecimal::new);
BigDecimal average =
        bigDecimals.stream()
                .collect(Collectors2.summarizingBigDecimal(e -> e))
                .getAverage(MathContext.DECIMAL32);

Assert.assertEquals(BigDecimal.valueOf(2.5), average);
```

Filtering



1. Process post titles.
2. Process API usage names.
3. Load prebuilt embeddings model.
4. Compute semantic similarity.

0.86

10

Compute Covariance Matrix in Java

Here is a short example, how you can create it with Apache Commons Math (3.5):

```
RealMatrix mx = MatrixUtils.createRealMatrix(new double[][]{
   {1, 2, 3},
   {2, 4, 6}
});
RealMatrix cov = new Covariance(mx).getCovarianceMatrix();
```

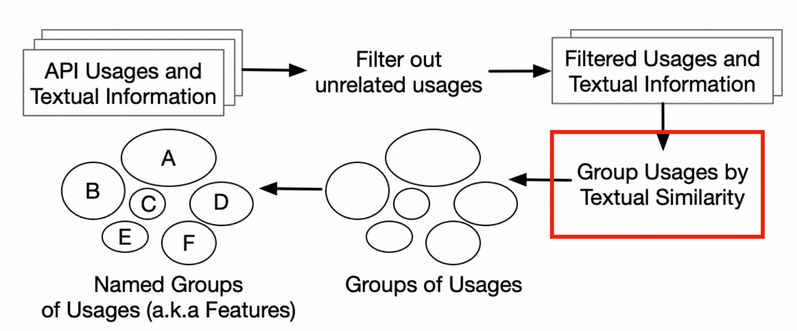
0.03

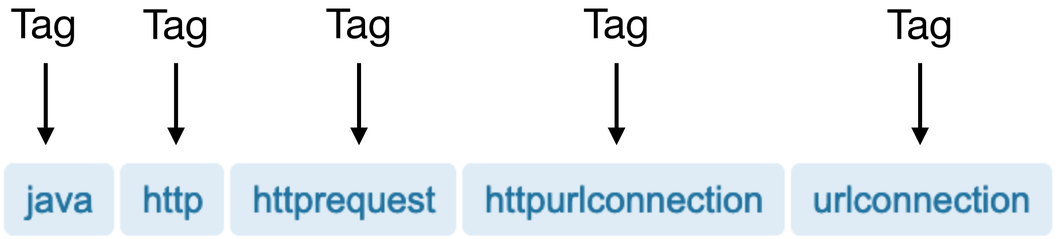
3

How to average BigDecimals using Streams?

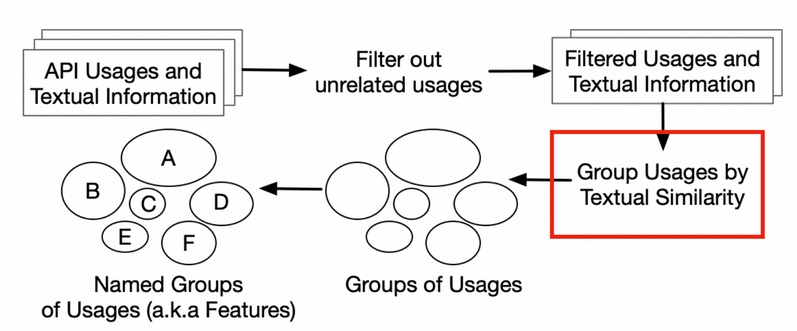
```
MutableDoubleList doubles = DoubleLists.mutable.with(1.0, 2.0, 3.0, 4.0);
List<BigDecimal> bigDecimals = doubles.collect(BigDecimal::new);
BigDecimal average =
        bigDecimals.stream()
                .collect(Collectors2.summarizingBigDecimal(e -> e))
                .getAverage(MathContext.DECIMAL32);

Assert.assertEquals(BigDecimal.valueOf(2.5), average);
```

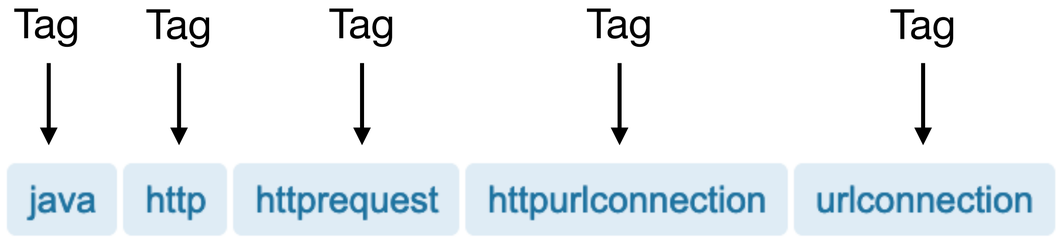
Grouping


API Usages and Textual Information
Filter out unrelated usages
Filtered Usages and Textual Information
Group Usages by Textual Similarity
A
B
C
D
E
F
Named Groups of Usages (a.k.a Features)
Groups of Usages

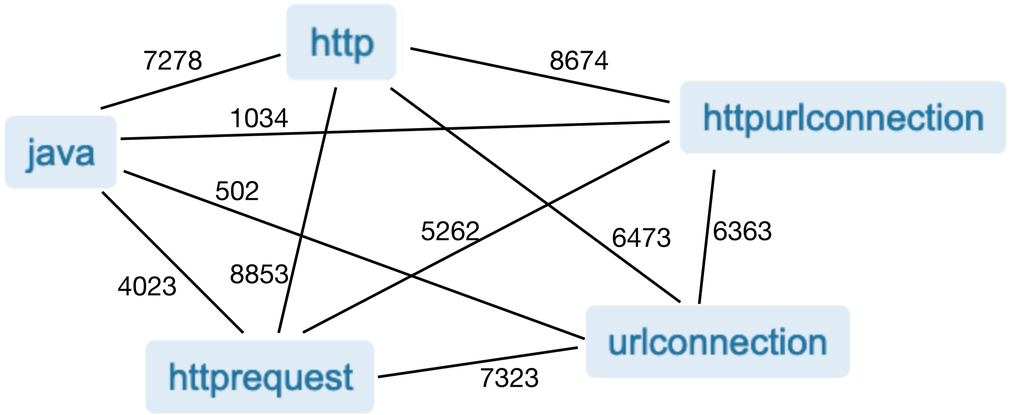

Tag
Tag
Tag
Tag
Tag
java
http
httprequest
httpurlconnection
urlconnection

Grouping

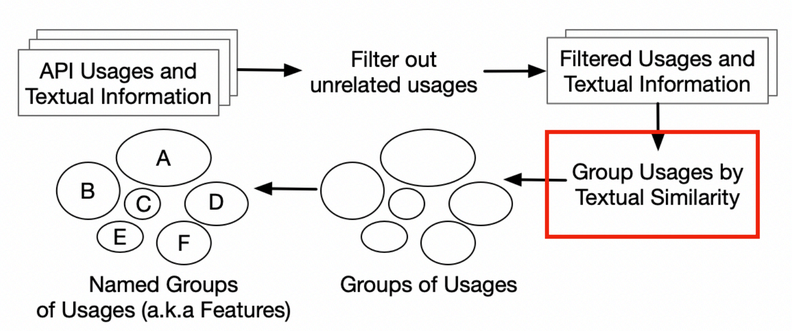


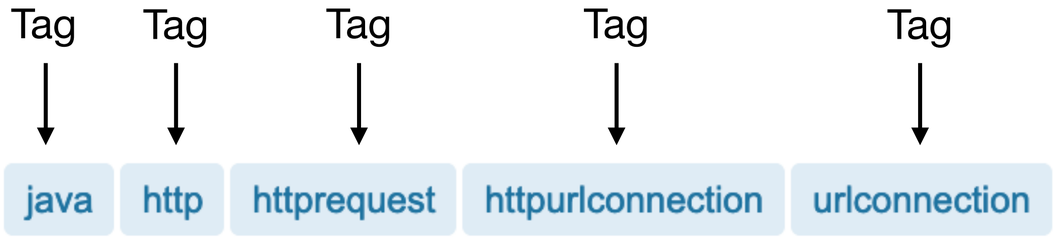


Relation between Tags

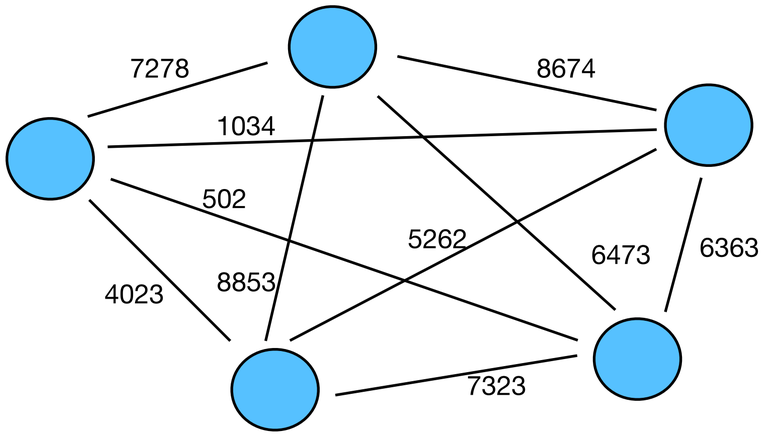


Grouping

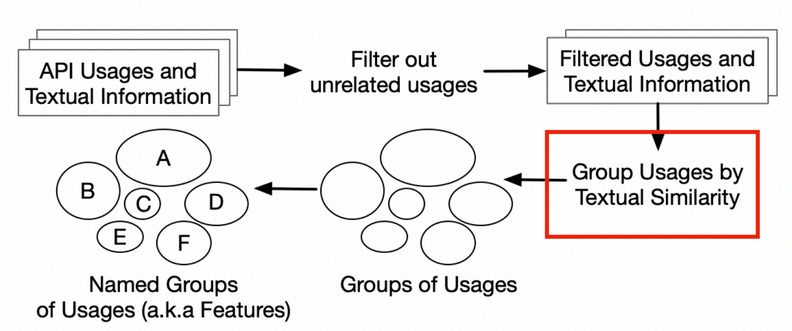


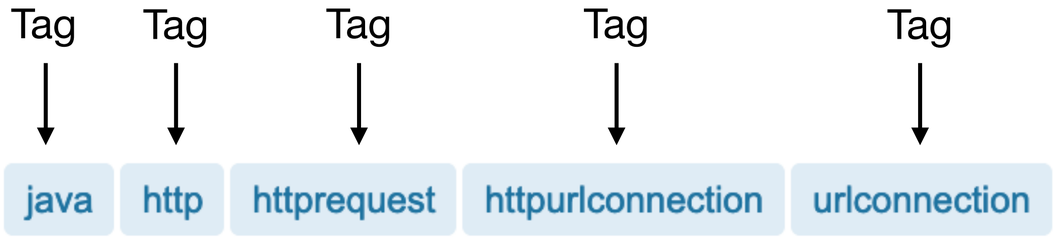


Relation between Tags

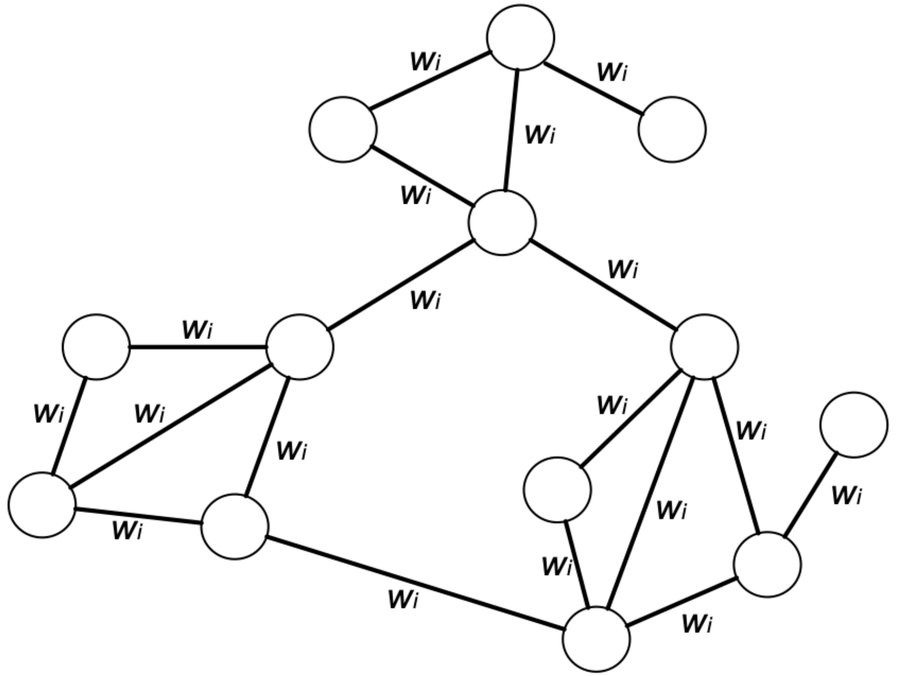


Grouping

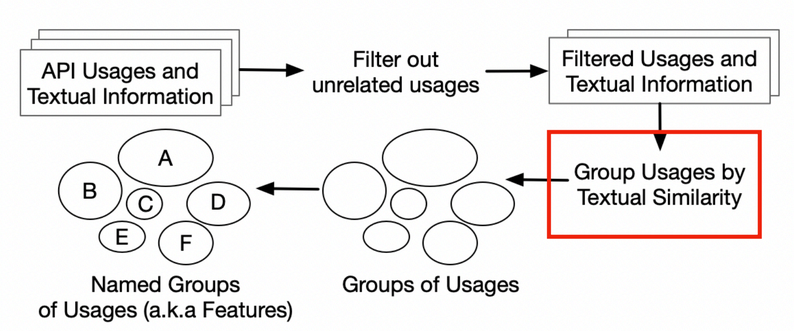


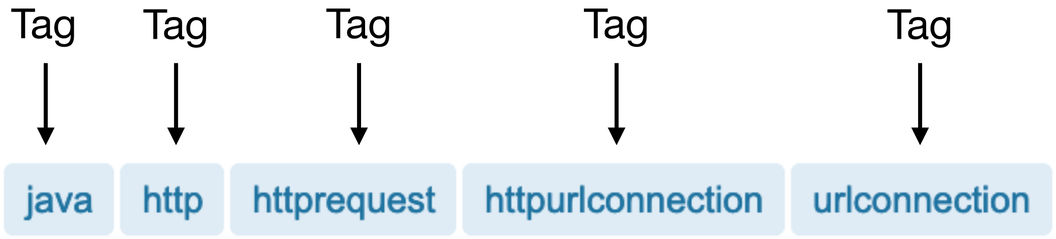


Relation between Tags

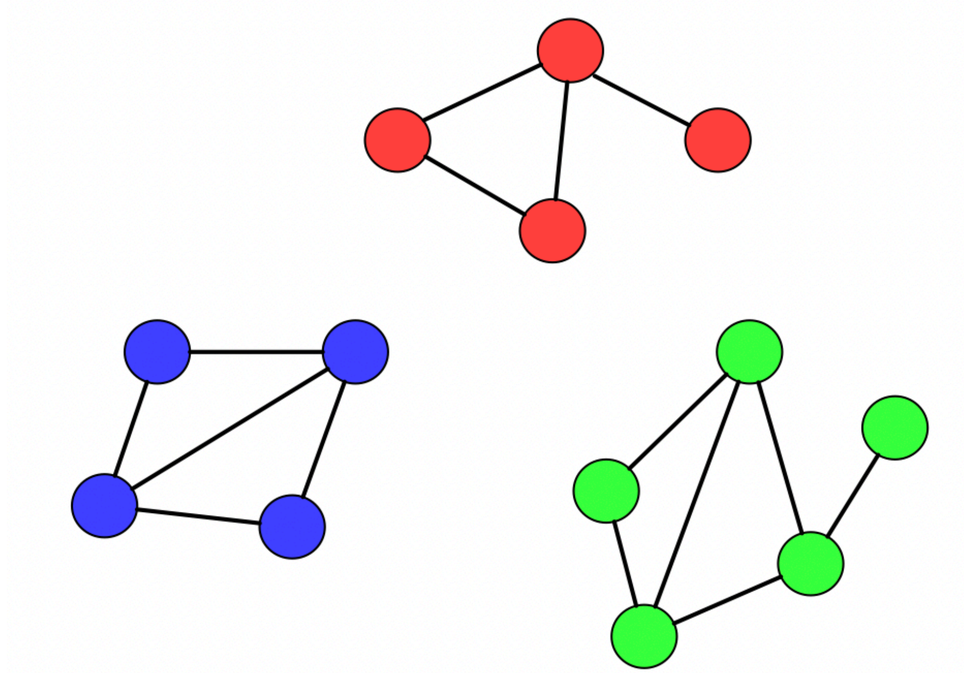


Grouping



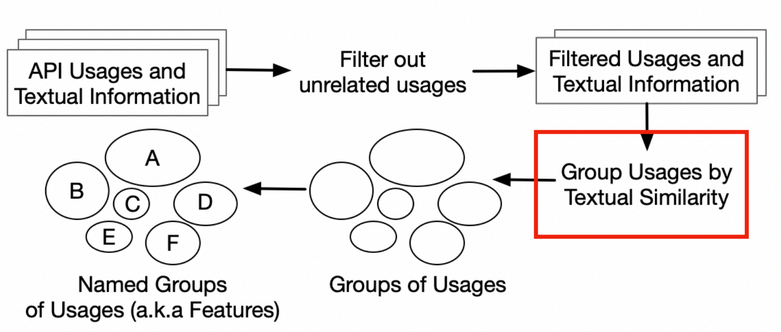


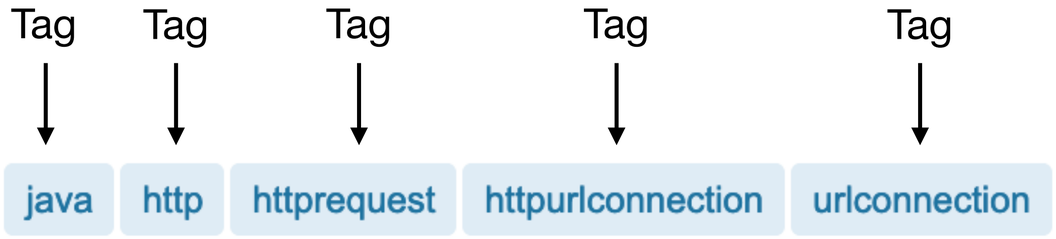
Relation between Tags



https://github.com/taynaud/python-louvain

Grouping





Identifying Features in Forks

Shurui Zhou
Carnegie Mellon University

Ştefan Stănciulescu
IT University of Copenhagen

Olaf Leßenich
University of Passau

Yingfei Xiong
Peking University

Andrzej Wąsowski
IT University of Copenhagen

Christian Kästner
Carnegie Mellon University

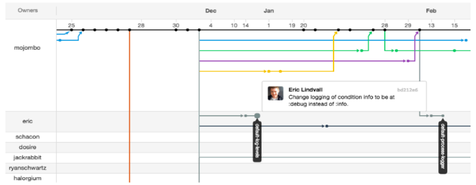
ABSTRACT

Fork-based development has been widely used both in open source communities and in industry, because it gives developers flexibility to modify their own fork without affecting others. Unfortunately, this mechanism has downsides: When the number of forks becomes large, it is difficult for developers to get or maintain an overview of activities in the forks. Current tools provide little help. We introduce INFOX, an approach to automatically identify non-merged features in forks and to generate an overview of active forks in a project. The approach clusters cohesive code fragments using code and network-analysis techniques and uses information-retrieval techniques to label clusters with keywords. The clustering is effective, with 90 % accuracy on a set of known features. In addition, a human-subject evaluation shows that INFOX can provide actionable insight for developers of forks.

ACM Reference Format:
Shurui Zhou, Ştefan Stănciulescu, Olaf Leßenich, Yingfei Xiong, Andrzej Wąsowski, and Christian Kästner. 2018. Identifying Features in Forks. In *ICSE '18: 40th International Conference on Software Engineering , May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3180155.3180205

1 INTRODUCTION

Forking is a lightweight and easy mechanism that allows developers, both in open source and in industry, to start development from

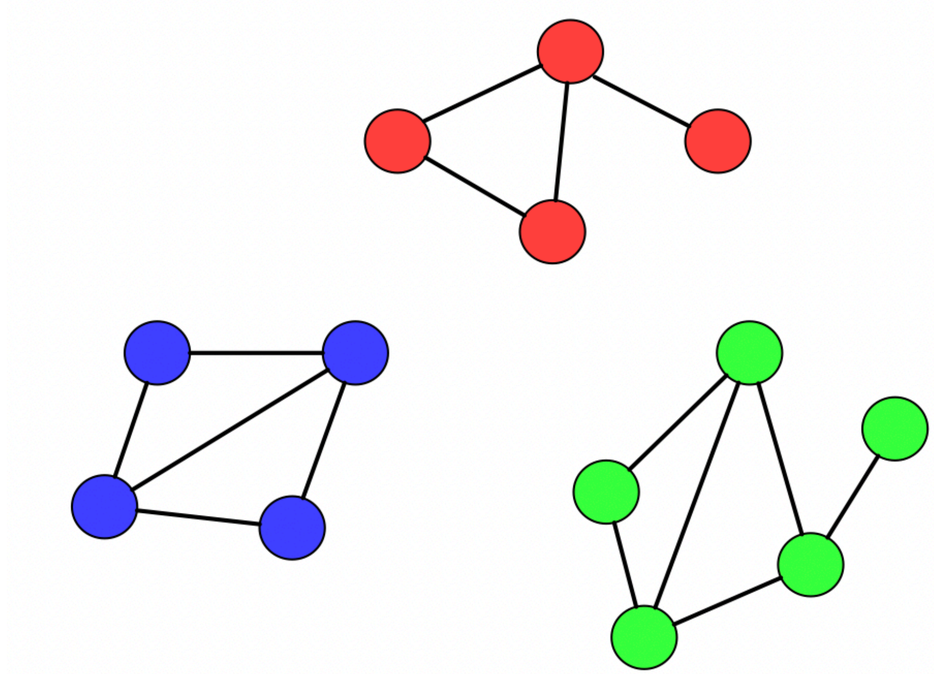


(a) GitHub network graph

(b) INFOX overview

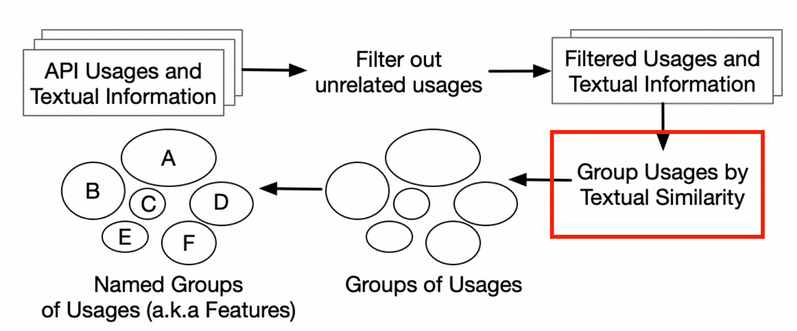
Figure 1: GitHub's network graph shows commits across known forks, but is difficult to use to gain an overview of activities in projects with many forks. INFOX's overview summarizes features in active forks.

Relation between Tags



https://github.com/taynaud/python-louvain

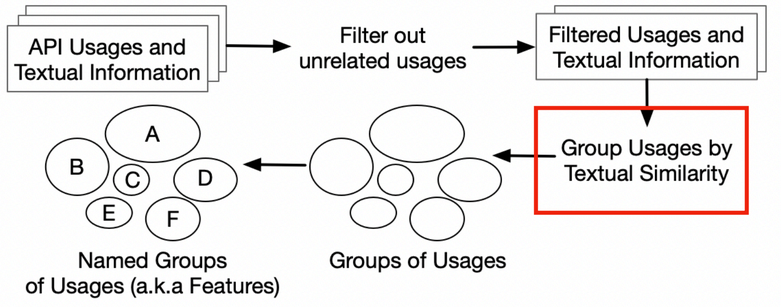
Groups



Library: org.apache.commons.math

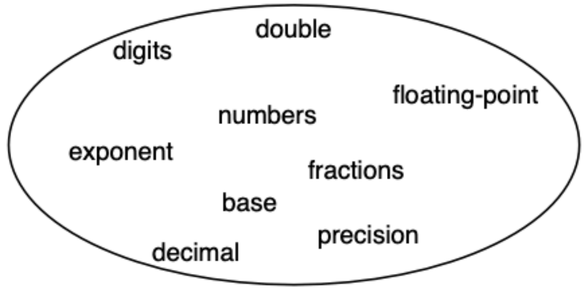
Mathematics Library from the Apache Foundation

Groups



Library: org.apache.commons.math

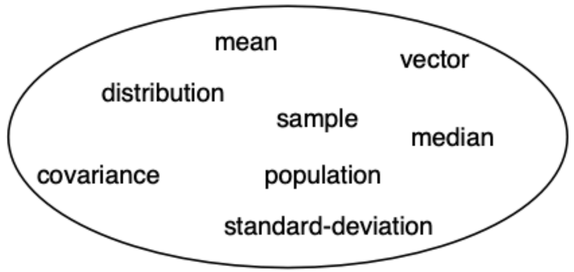
Mathematics Library from the Apache Foundation




```
ComplexFormat.parse
Complex.getReal
Complex.getImaginary
```

```
BigFraction.getNumerator
BigFraction.getDenominator
```

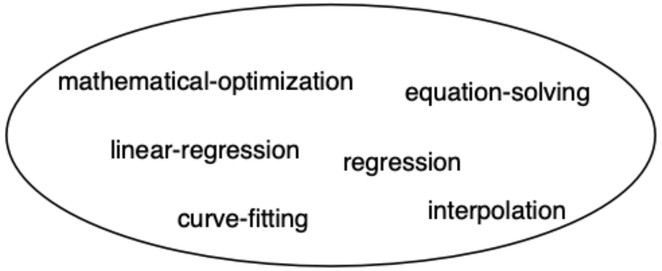
```
Precision.round
```




```
SummaryStatistics.getMean
SummaryStatistics.addValue
SummaryStatistics.getStandardDeviation
```

```
NormalDistribution.cumulativeProbability
NormalDistribution.inverseCumulativeProbability
```

```
LogNormalDistribution.sample
```

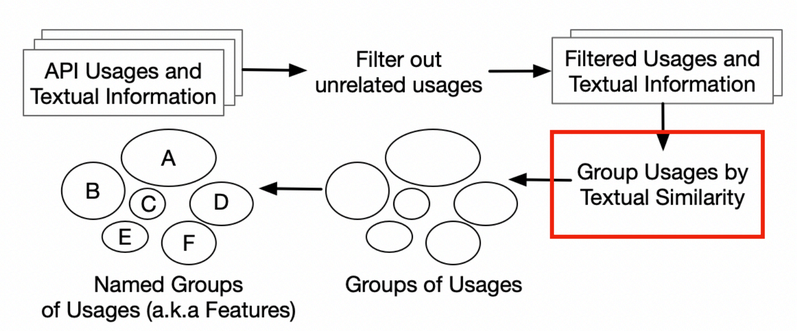



```
PolynomialSplineFunction.getPolynomials
PolynomialSplineFunction.value
LinearInterpolator.interpolate
```

```
PointValuePair.getPoint
SimplexSolver.optimize
```

```
UnivariateOptimizer.optimize
UnivariateOptimizer.getMaxEvaluations
UnivariateOptimizer.getEvaluations
```

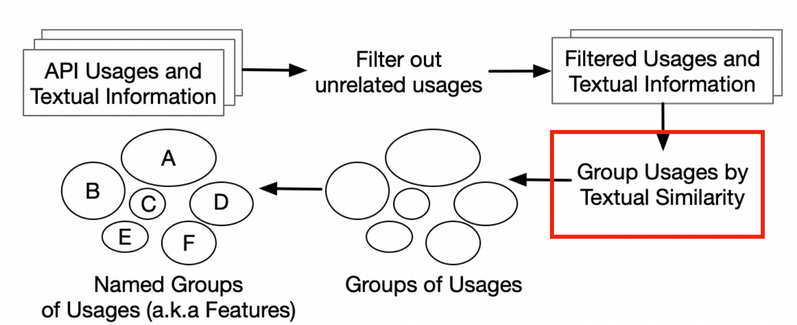
Groups



Library: com.itextpdf

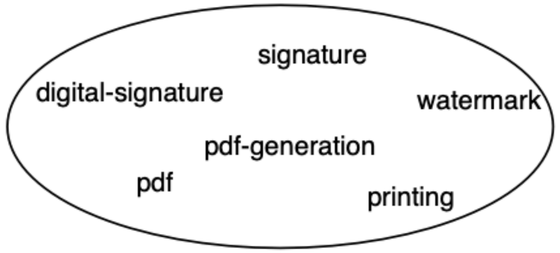
PDF creation and transformation

Groups



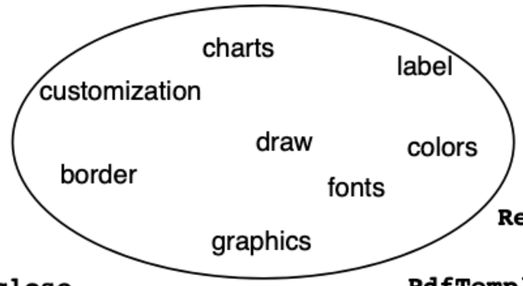
Library: com.itextpdf

PDF creation and transformation




```
PdfStamper.setEncryption    MakeSignature.signDeferred
PdfStamper.close
```

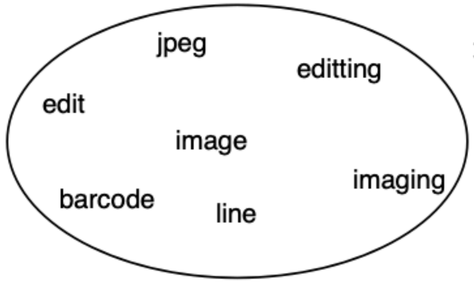
```
PdfCopy.addDocument
Document.open
Document.isOpen
Document.close
```




```
Rectangle.setBackgroundColor
```

```
Document.close
PdfContentByte.createGraphics
Document.open
DocumentException.getMessage
Font.createFont
PdfWriter.getInstance
```

```
PdfTemplate.fill
PdfTemplate.setColorFill
Image.getInstance
PdfTemplate.rectangle
```

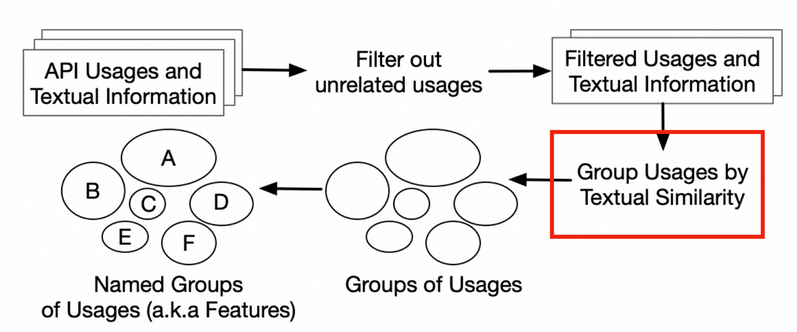



```
Image.getInstance
```

```
Image.getWidth
BarcodeQRCode.createAwtImage
```

```
PdfStamper.close
PdfStamper.getOverContent
ColumnText.showTextAligned
PdfReader.close
```

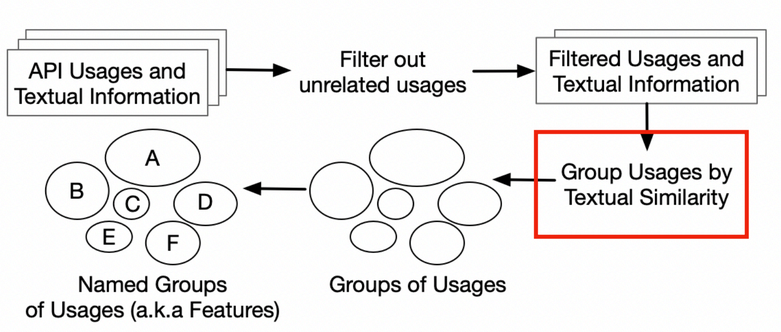
Groups



Library: nz.ac.waikato.cms.weka

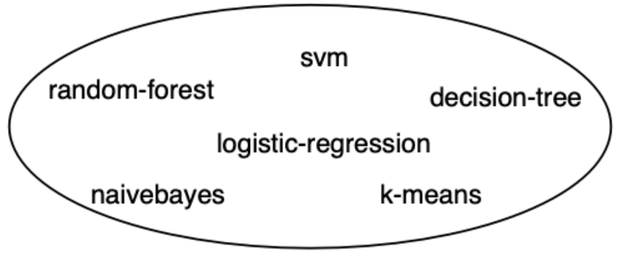
Data mining and Machine Learning Library

Groups

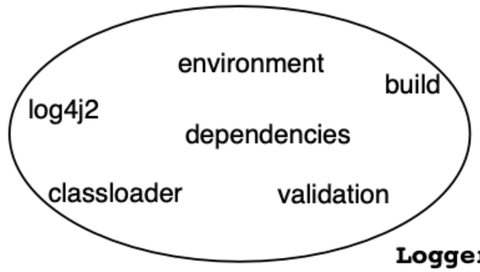


Library: nz.ac.waikato.cms.weka

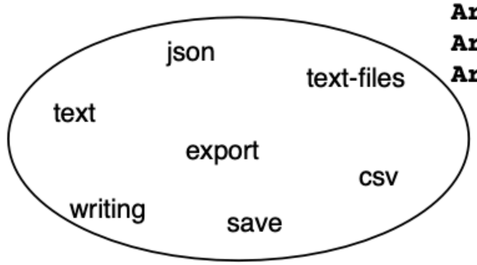
Data mining and Machine Learning Library




```
J48.distributionForInstance
DataSource.getDataSet
J48.classifyInstance
SimpleKMeans.setNumClusters
ClusterEvaluation.evaluateClusterer
ClusterEvaluation.setClusterer
SimpleKMeans.buildClusterer
ClusterEvaluation.clusterResultsToString
```

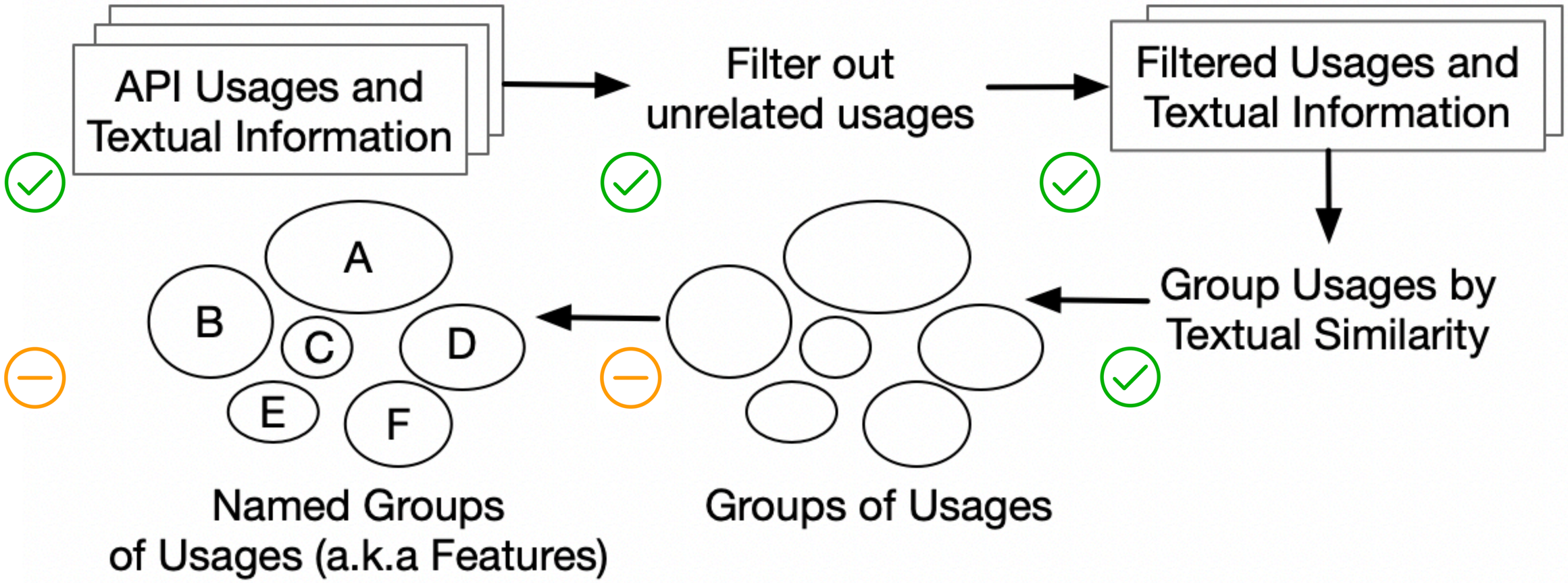



```
Logger.getLogger
Environment.getEnvironment
Evaluation.evaluateModel
Environment.getProperty
Classifier.buildClassifier
Evaluation.evaluateModel
Classifier.buildClassifier
```




```
ArffSaver.setDestination
ArffSaver.writeBatch
ArffSaver.setInstances
CSVLoader.getDataSet
CSVLoader.setSource
Instances.setClassIndex
Instances.numAttributes
Evaluation.evaluateModel
Evaluation.pctCorrect
Evaluation.toSummaryString
```

Future Work


API Usages and Textual Information
Filter out unrelated usages
Filtered Usages and Textual Information
Group Usages by Textual Similarity
Groups of Usages
A
B
C
D
E
F
Named Groups of Usages (a.k.a Features)

Conclusion

Introduction

Software products that evolve together in the same environment [1]

Library Selection Issues

I/O Utilities (17)
HTTP Clients (81)
Dependency Injection (51)
Mocking (41)
XML Processing (34)
JSON Libraries (74)
Collections (72)
Reflection Libraries (47)
Command Line Parsers (45)
⋮

JCommander Library

Version	Repository	Usages	Date
1.78	Central	20	Aug, 2019
1.77	Central	2	Aug, 2019
1.75	Central	1	Aug, 2019
1.72	Central	193	Jun, 2017
1.69	Central	18	Apr, 2017
1.64	Central	23	Mar, 2017
1.60	Central	30	Jan, 2017
1.58	Central	21	Sep, 2016
1.48	Central	259	Apr, 2015
1.47	Central	24	Dec, 2014
1.35	Central	123	Feb, 2014
1.32	Central	10	Sep, 2013
1.30	Central	94	Oct, 2012

[1] - M. Lungu, "Towards reverse engineering software ecosystems," in *2008 ICSM*. IEEE, 2008, pp. 428–431.

1

Introduction

Selecting Third-Party Libraries: The Practitioners' Perspective

Enrique Larios Vargas
Software Improvement Group
The Netherlands
e.lariosvargas@sig.eu

Mauricio Aniche
Delft University of Technology
The Netherlands
M.FinavaroAniche@tudelft.nl

Christoph Treude
University of Adelaide
Australia
christoph.treude@adelaide.edu.au

Magiel Bruntink
Software Improvement Group
The Netherlands
m.bruntink@sig.eu

Georgios Gousios
Delft University of Technology
The Netherlands
G.Gousios@tudelft.nl

ABSTRACT

The selection of third-party libraries is an essential element of virtually any software development project. However, deciding which libraries to choose is a challenging practical problem. Selecting the wrong library can severely impact a software project in terms of cost, time, and development effort, with the severity of the impact depending on the role of the library in the software architecture, among others. Despite the importance of following a careful library selection process, in practice, the selection of third-party libraries is still conducted in an ad-hoc manner, where dozens of factors play an influential role in the decision.

In this paper, we study the factors that influence the selection process of libraries, as perceived by industry developers. To that aim, we perform a cross-sectional interview study with 16 developers from 11 different businesses and survey 115 developers that are involved in the selection of libraries. We systematically devised a comprehensive set of 26 technical, human, and economic factors that developers take into consideration when selecting a software library. Eight of these factors are new to the literature. We explain each of these factors and how they play a role in the decision. Finally,

Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20), November 8–13, 2020, Virtual Event, USA. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3368089.3409711

1 INTRODUCTION

The use of third-party libraries[1] is a recurrent practice among practitioners to speed up the development of software systems and, as a consequence, to reduce its costs [20]. Software libraries provide developers with customized functionality which is useful while developing software [15]. Developers do not need to "reinvent the wheel", but rather seek libraries that suit their purpose [22].

The selection of the right library to adopt is a demanding task for developers. The number of factors that should be taken into account during the selection process is significant. For example, the adoption of a library that goes against the existing architectural decisions of the software system might increase its adoption time; a library that is easy to use but not performant enough might have an impact on the quality of the system; a library that has good documentation and active community, but with which the team has had a bad experience, might not be the best possible choice. On top

Survey of 16 participants from different companies

1. Technical Factors
   - Functionality
   - Quality
   - Type of Project
   - Release Process
2. Human Factors
   - Stakeholders
   - Organization
   - Individual
   - Community
3. Economic Factors
   - Total cost of ownership
   - Risk

2

Introduction

Stack Overflow Q&A

How do I convert a String to an int in Java?
Guava equivalent for IOUtils.toString(InputStream)
How do I compare strings in Java?
Remove last character of a StringBuilder?
Guava: Splitter and considering Escaping?
String utilities

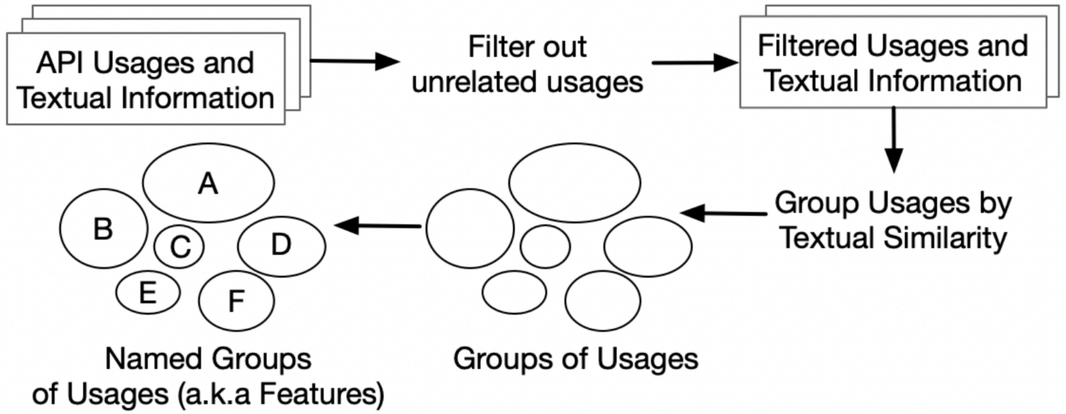
initializing a Guava ImmutableMap
Flattening an Iterable<Iterable<T>> in Guava
How to directly initialize a HashMap (in a literal way)?
Google Guava isNullOrEmpty for collections
Google Guava "zip" two lists
Collection utilities

Merge ranges with guava
Ranges

What is the simplest way to read a file into String?
Utils to read resource text file to String (Java)
I/O

4

Pipeline

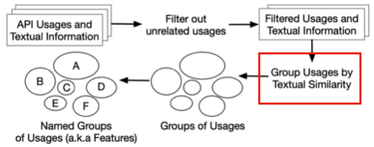


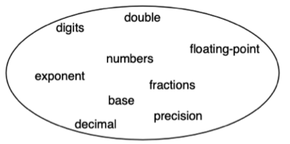
5

Groups

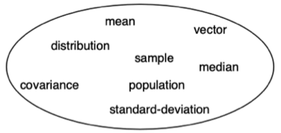
Library: org.apache.commons.math

Mathematics Library from the Apache Foundation

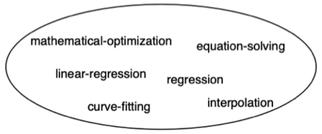





```
ComplexFormat.parse
Complex.getReal
Complex.getImaginary
BigFraction.getNumerator
BigFraction.getDenominator
Precision.round
```




```
SummaryStatistics.getMean
SummaryStatistics.addValue
SummaryStatistics.getStandardDeviation
NormalDistribution.cumulativeProbability
NormalDistribution.inverseCumulativeProbability
LogNormalDistribution.sample
```




```
PolynomialSplineFunction.getPolynomials
PolynomialSplineFunction.value
LinearInterpolator.interpolate
PointValuePair.getPoint
SimplexSolver.optimize
UnivariateOptimizer.optimize
UnivariateOptimizer.getMaxEvaluations
UnivariateOptimizer.getEvaluations
```

9