HBase Dependency Extraction

Team5Star

•••

Daniel McVicar - 213027479
Rafay Sheikh - 213033451
Daniel Fortunato - 216796443
Adham El Shafie - 212951018
Yahya Ismail- 213235403

Overview

1. Introduction (About Dependency Extraction)
2. Understand
3. SRCML
4. DependencyExtractor
5. Comparison Process
6. Qualitative Analysis
7. Risks and Limitations
8. Conclusion (+Lessons Learned)

Introduction

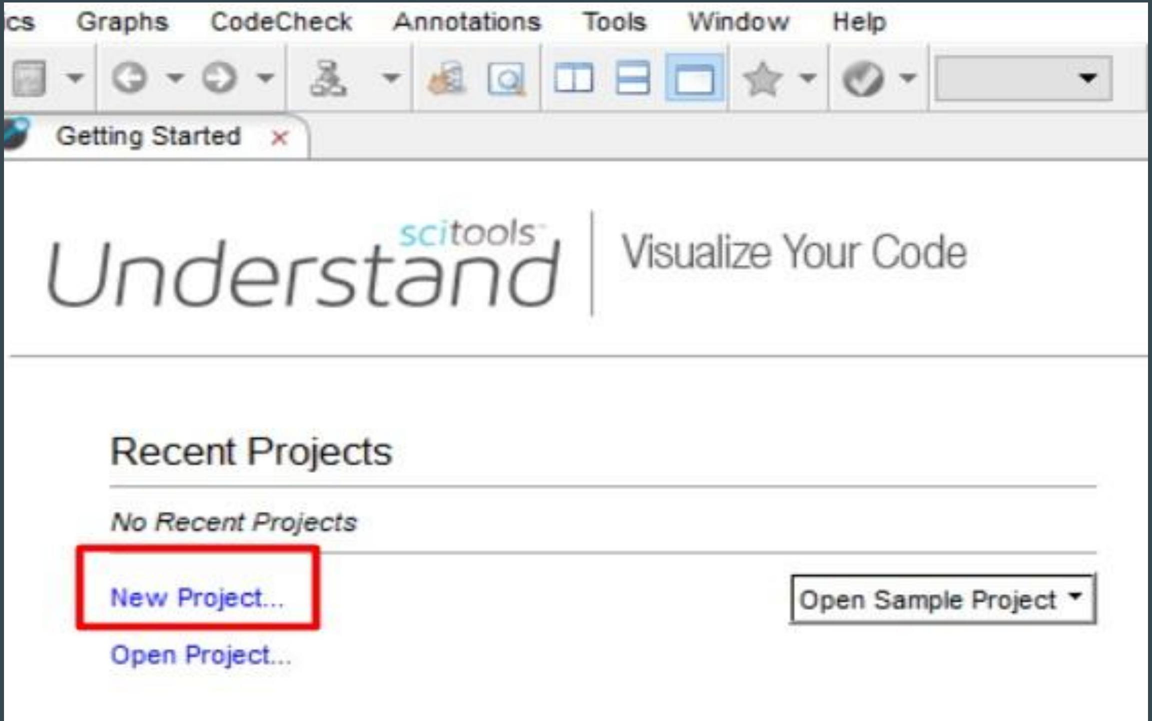
Extraction of file level dependencies from source code (directory)

- File dependencies help show how a software works/functions
  - Helps engineers better understand the software to maintain, update etc
- Dependency extraction is a step in the architecture recovery pipeline
- Can be used to understand code with poor documentation
- Analyze code architecture for optimization purposes and code metrics
- 3 approaches for dependency extraction that we used:
  - Understand
  - srcML
  - DependencyExtractor

What is Understand

- IDE to visualize and analyze static code
- Developed to aid software developers to understand and document source code
- Also helps new developers to comprehend legacy code passed down to them
- Supports a variety of source code languages

Extracting Dependencies

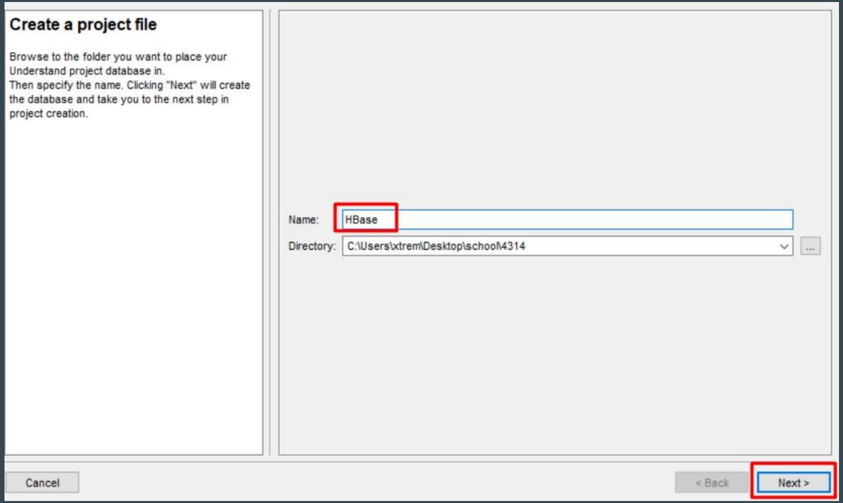


Recent Projects

No Recent Projects

Once understand is installed, opening it up results in the figure shown

Extracting Dependencies



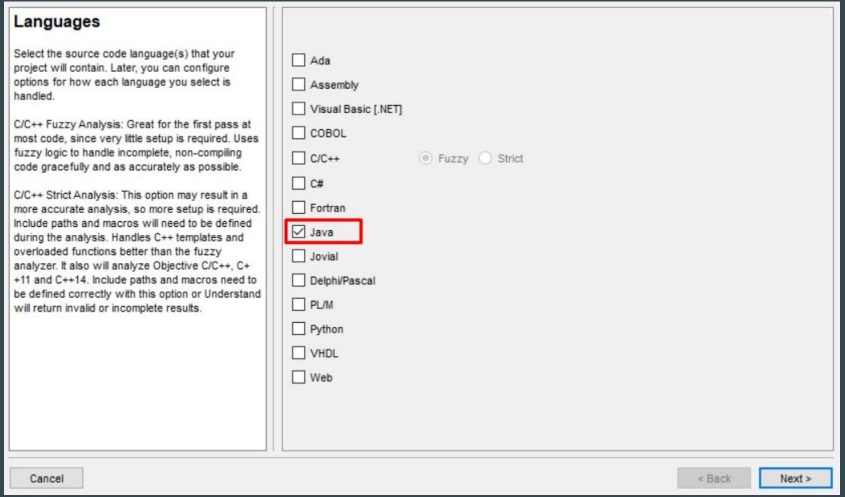
Create a project file

Browse to the folder you want to place your Understand project database in.
Then specify the name. Clicking "Next" will create the database and take you to the next step in project creation.

Name: HBase

Directory: C:\Users\xtrem\Desktop\school\4314

Rename the current project to preferred name



Languages

Select the source code language(s) that your project will contain. Later, you can configure options for how each language you select is handled.

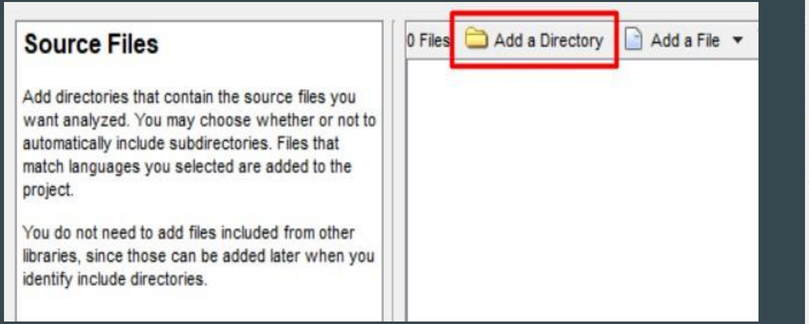
C/C++ Fuzzy Analysis: Great for the first pass at most code, since very little setup is required. Uses fuzzy logic to handle incomplete, non-compiling code gracefully and as accurately as possible.

C/C++ Strict Analysis: This option may result in a more accurate analysis, so more setup is required. Include paths and macros will need to be defined during the analysis. Handles C++ templates and overloaded functions better than the fuzzy analyzer. It also will analyze Objective C/C++, C++11 and C++14. Include paths and macros need to be defined correctly with this option or Understand will return invalid or incomplete results.

- □ Ada
- □ Assembly
- □ Visual Basic [.NET]
- □ COBOL
- □ C/C++ ● Fuzzy ○ Strict
- □ C#
- □ Fortran
- ☑ Java
- □ Jovial
- □ Delphi/Pascal
- □ PL/M
- □ Python
- □ VHDL
- □ Web

Supported languages for source code

Extracting Dependencies

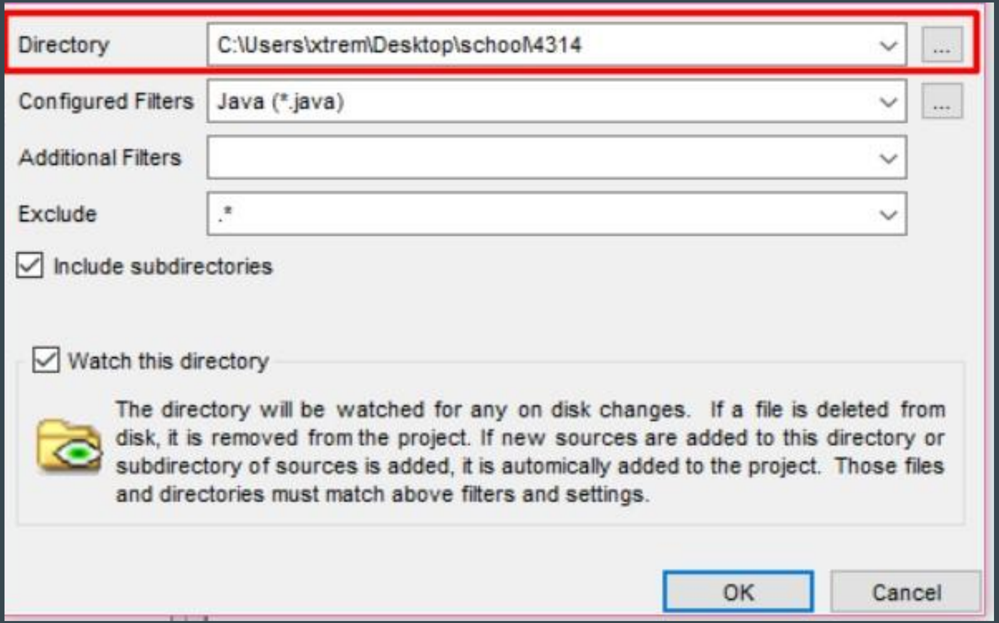


Source Files

Add directories that contain the source files you want analyzed. You may choose whether or not to automatically include subdirectories. Files that match languages you selected are added to the project.

You do not need to add files included from other libraries, since those can be added later when you identify include directories.

Can add directories or files related to source code



Directory: C:\Users\xtrem\Desktop\school\4314

Configured Filters: Java (*.java)

Additional Filters:

Exclude: .*

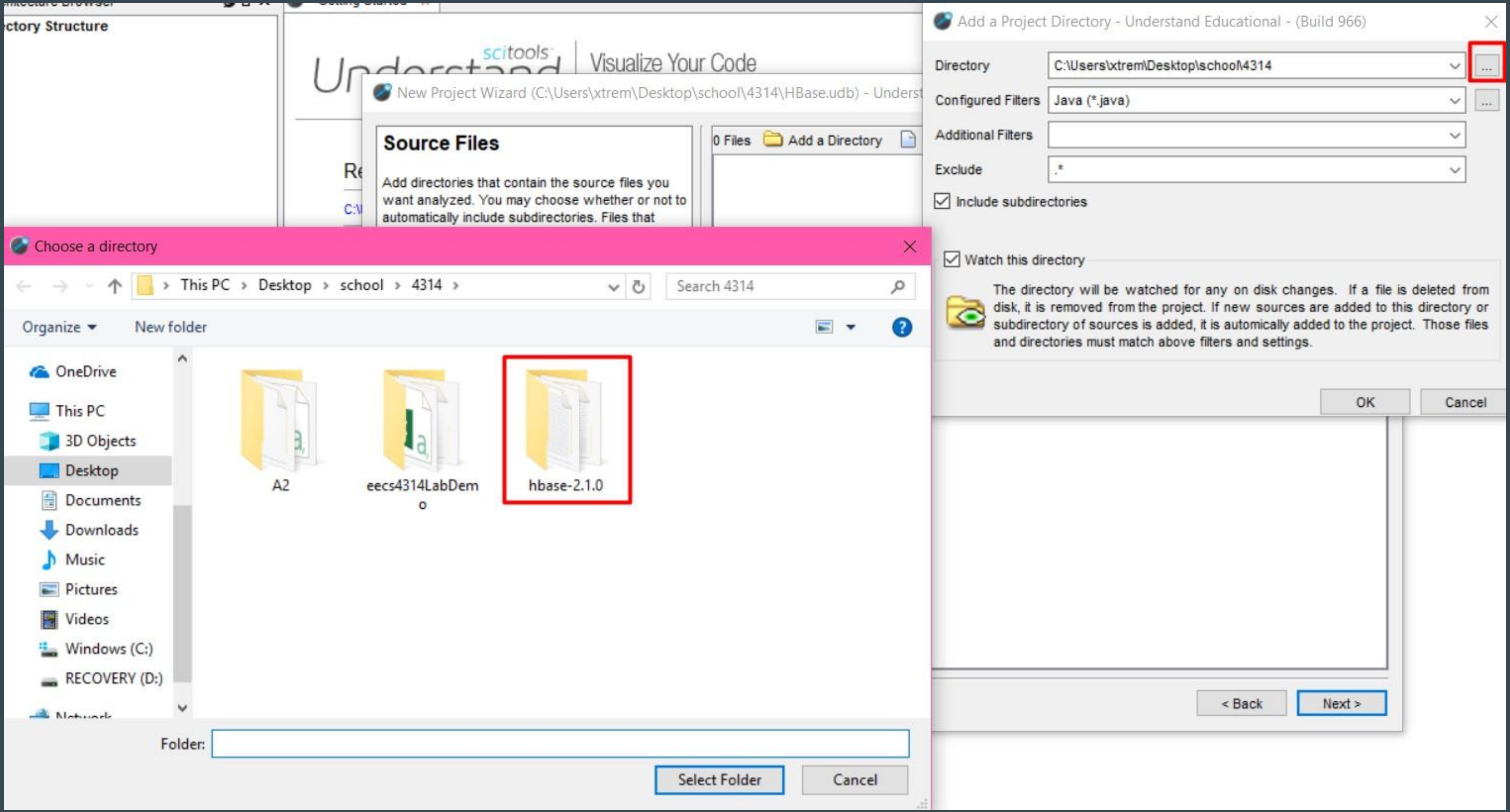
☑ Include subdirectories

☑ Watch this directory

The directory will be watched for any on disk changes. If a file is deleted from disk, it is removed from the project. If new sources are added to this directory or subdirectory of sources is added, it is automically added to the project. Those files and directories must match above filters and settings.

“Include subdirectories” and “watch this directory” are checked

Extracting Dependencies



New Project Wizard (C:\Users\xtrem\Desktop\school\4314\HBase.udb) - Underst

Source Files

Add directories that contain the source files you want analyzed. You may choose whether or not to automatically include subdirectories. Files that

Add a Project Directory - Understand Educational - (Build 966)

Directory: C:\Users\xtrem\Desktop\school\4314

Configured Filters: Java (*.java)

Additional Filters:

Exclude: .*

☑ Include subdirectories

☑ Watch this directory

The directory will be watched for any on disk changes. If a file is deleted from disk, it is removed from the project. If new sources are added to this directory or subdirectory of sources is added, it is automically added to the project. Those files and directories must match above filters and settings.

Choose a directory

This PC › Desktop › school › 4314 ›

A2

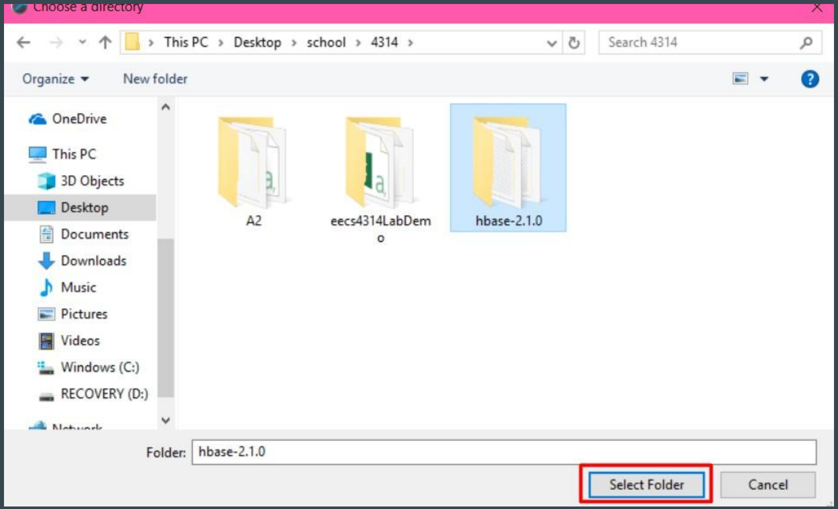
eecs4314LabDemo

hbase-2.1.0

Folder:

Navigate to the required source code root directory

Extracting Dependencies

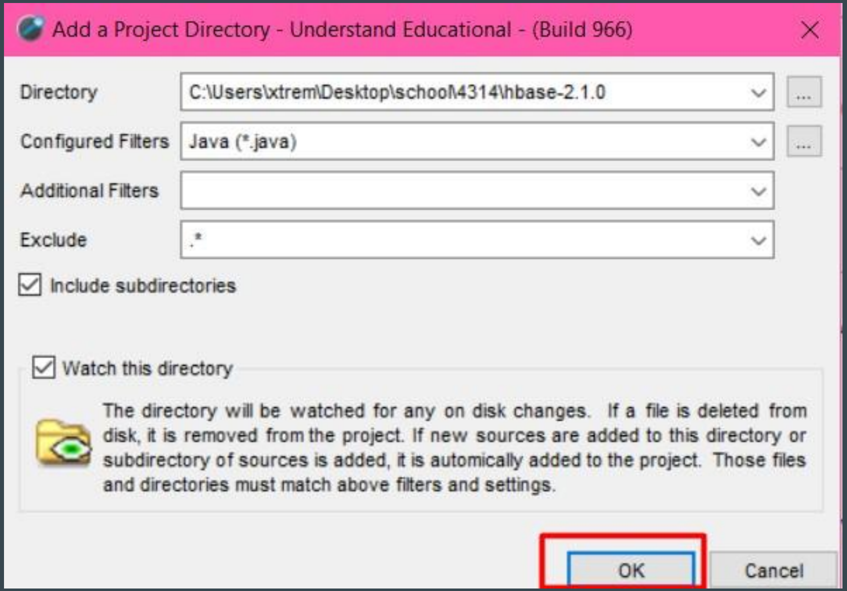


Choose a directory

This PC › Desktop › school › 4314 ›

Folder: hbase-2.1.0

Confirm root directory



Add a Project Directory - Understand Educational - (Build 966)

Directory: C:\Users\xtrem\Desktop\school\4314\hbase-2.1.0

Configured Filters: Java (*.java)

Additional Filters:

Exclude: .*

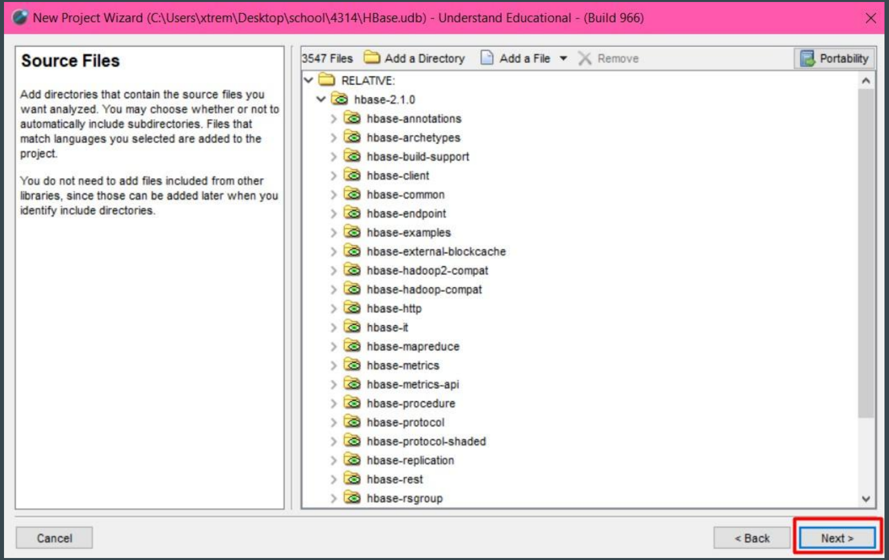
☑ Include subdirectories

☑ Watch this directory

The directory will be watched for any on disk changes. If a file is deleted from disk, it is removed from the project. If new sources are added to this directory or subdirectory of sources is added, it is automically added to the project. Those files and directories must match above filters and settings.

HBase is coded in java therefore it is in the "Configured Filters"

Extracting Dependencies



New Project Wizard (C:\Users\xtrem\Desktop\school\4314\HBase.udb) - Understand Educational - (Build 966)

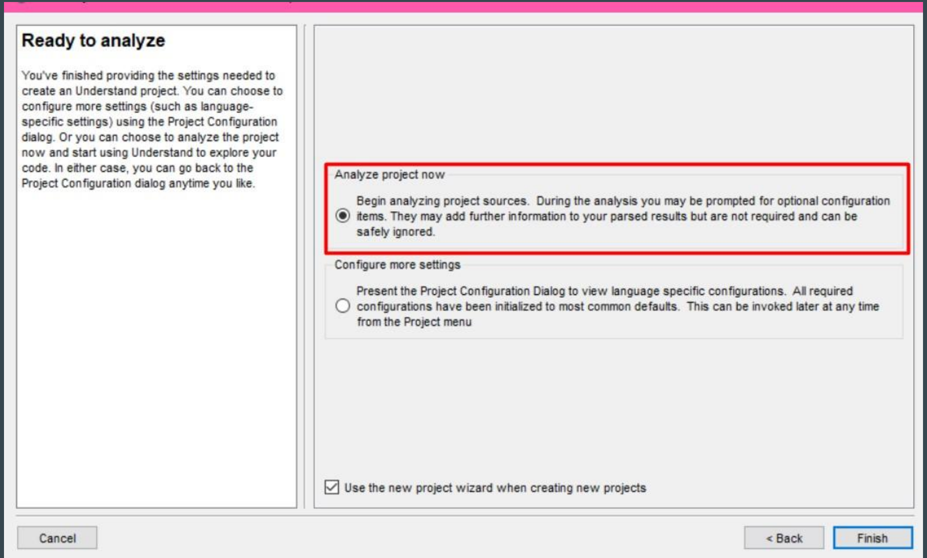
Source Files

Add directories that contain the source files you want analyzed. You may choose whether or not to automatically include subdirectories. Files that match languages you selected are added to the project.

You do not need to add files included from other libraries, since those can be added later when you identify include directories.

3547 Files

- RELATIVE:
  - hbase-2.1.0
    - hbase-annotations
    - hbase-archetypes
    - hbase-build-support
    - hbase-client
    - hbase-common
    - hbase-endpoint
    - hbase-examples
    - hbase-external-blockcache
    - hbase-hadoop2-compat
    - hbase-hadoop-compat
    - hbase-http
    - hbase-it
    - hbase-mapreduce
    - hbase-metrics
    - hbase-metrics-api
    - hbase-procedure
    - hbase-protocol
    - hbase-protocol-shaded
    - hbase-replication
    - hbase-rest
    - hbase-rsgroup



Ready to analyze

You've finished providing the settings needed to create an Understand project. You can choose to configure more settings (such as language-specific settings) using the Project Configuration dialog. Or you can choose to analyze the project now and start using Understand to explore your code. In either case, you can go back to the Project Configuration dialog anytime you like.

Analyze project now

● Begin analyzing project sources. During the analysis you may be prompted for optional configuration items. They may add further information to your parsed results but are not required and can be safely ignored.

Configure more settings

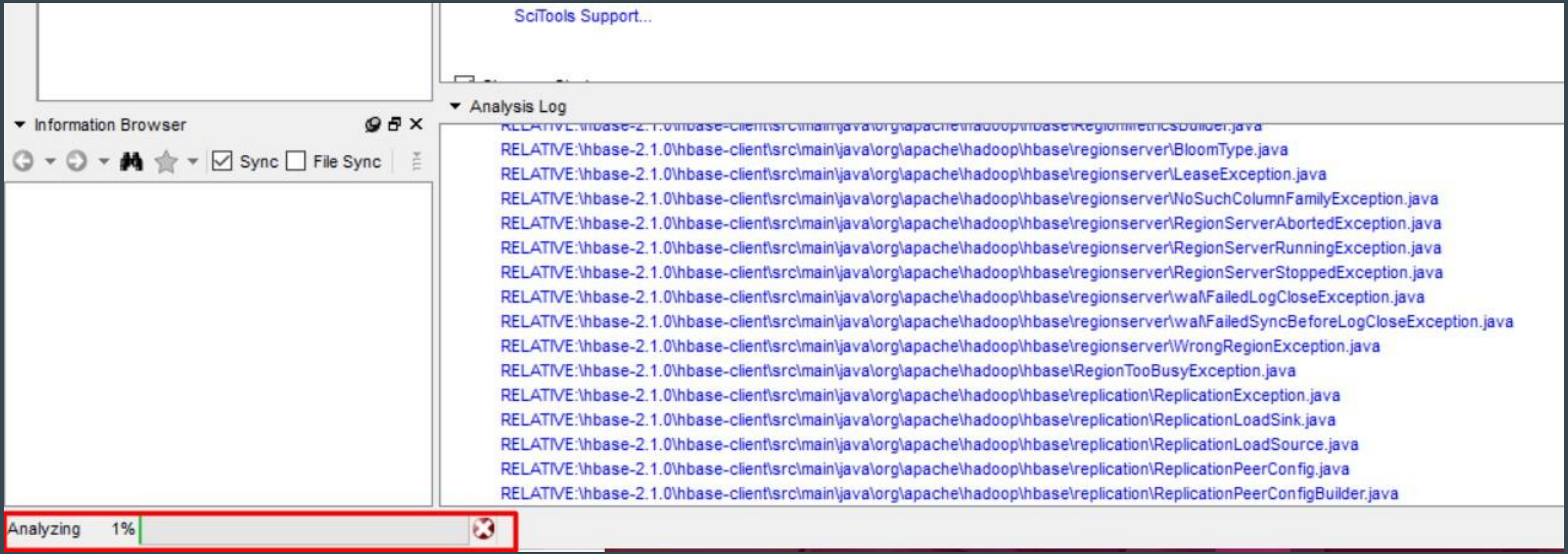
○ Present the Project Configuration Dialog to view language specific configurations. All required configurations have been initialized to most common defaults. This can be invoked later at any time from the Project menu

☑ Use the new project wizard when creating new projects

Subdirectories of HBase root directory shown

Can configure more settings if need be

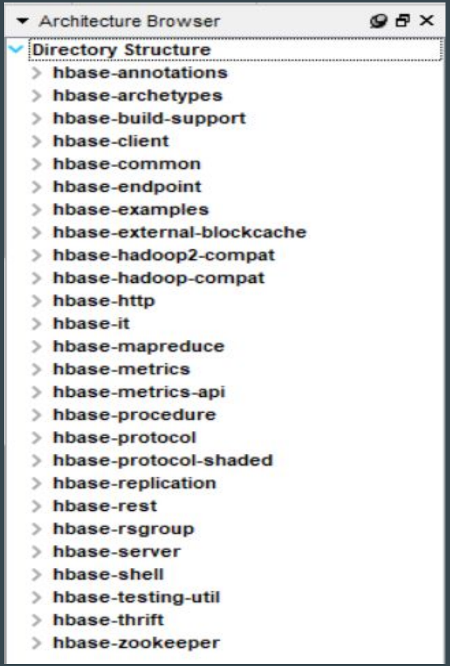
Extracting Dependencies



```
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\RegionMetricsBuilder.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\BloomType.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\LeaseException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\NoSuchColumnFamilyException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\RegionServerAbortedException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\RegionServerRunningException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\RegionServerStoppedException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\wal\FailedLogCloseException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\wal\FailedSyncBeforeLogCloseException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\regionserver\WrongRegionException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\RegionTooBusyException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\replication\ReplicationException.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\replication\ReplicationLoadSink.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\replication\ReplicationLoadSource.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\replication\ReplicationPeerConfig.java
RELATIVE:\hbase-2.1.0\hbase-client\src\main\java\org\apache\hadoop\hbase\replication\ReplicationPeerConfigBuilder.java
```

Understand begins analyzing the source code for errors and warnings

Extracting Dependencies

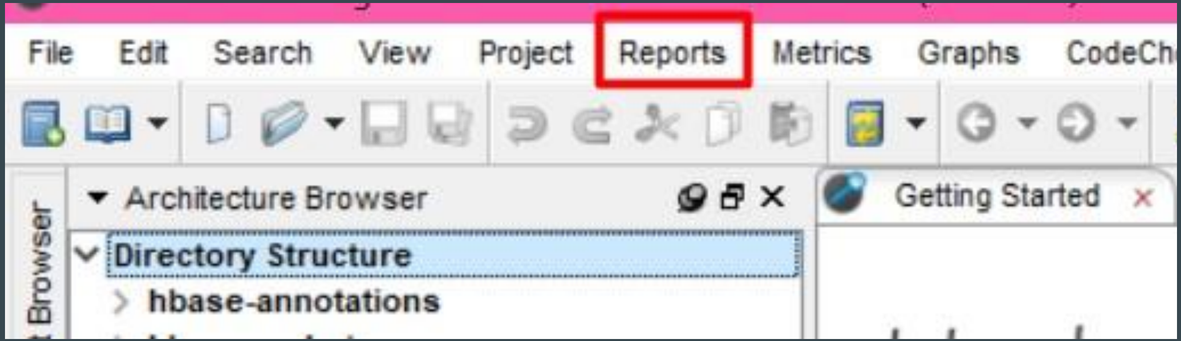


Architecture Browser

Directory Structure

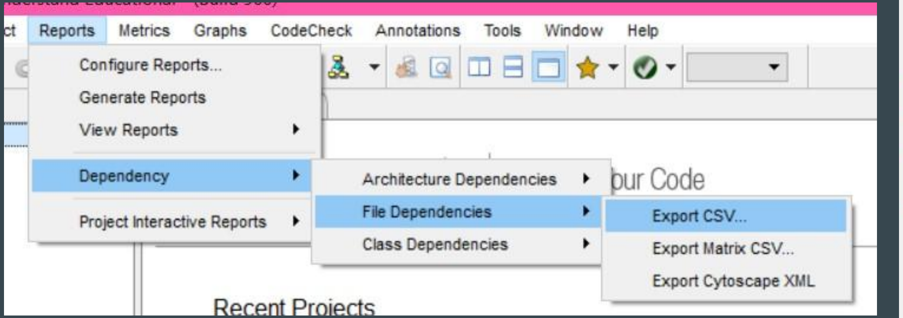
- hbase-annotations
- hbase-archetypes
- hbase-build-support
- hbase-client
- hbase-common
- hbase-endpoint
- hbase-examples
- hbase-external-blockcache
- hbase-hadoop2-compat
- hbase-hadoop-compat
- hbase-http
- hbase-it
- hbase-mapreduce
- hbase-metrics
- hbase-metrics-api
- hbase-procedure
- hbase-protocol
- hbase-protocol-shaded
- hbase-replication
- hbase-rest
- hbase-rsgroup
- hbase-server
- hbase-shell
- hbase-testing-util
- hbase-thrift
- hbase-zookeeper

Analyzed project shown in "Architecture Browser"

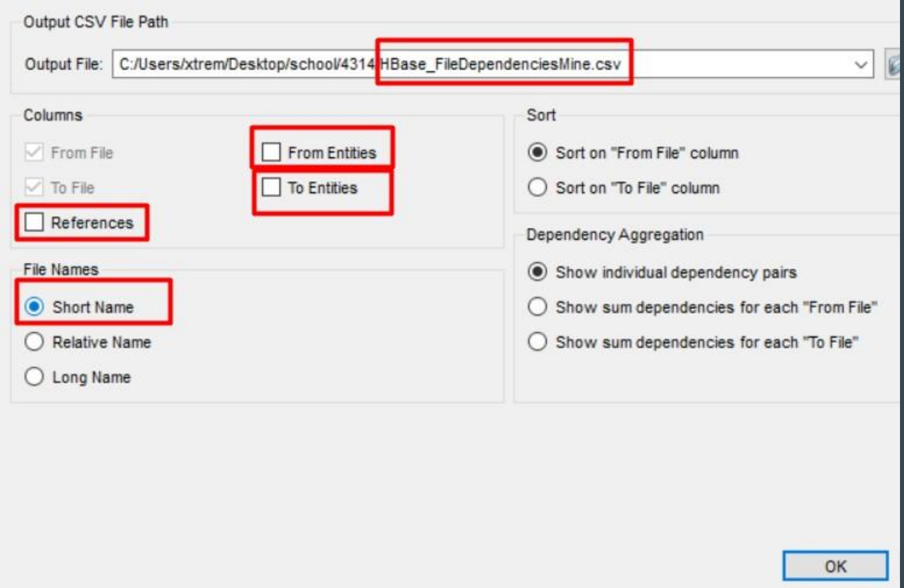


Highlight project directory and export csv file of dependencies in HBase

Extracting Dependencies



Export a .csv (comma separated values) of file dependencies


Output CSV File Path

Output File: C:/Users/xtrem/Desktop/school/4314/HBase_FileDependenciesMine.csv

Columns

☑ From File
☑ To File
☐ References
☐ From Entities
☐ To Entities

File Names

● Short Name
○ Relative Name
○ Long Name

Sort

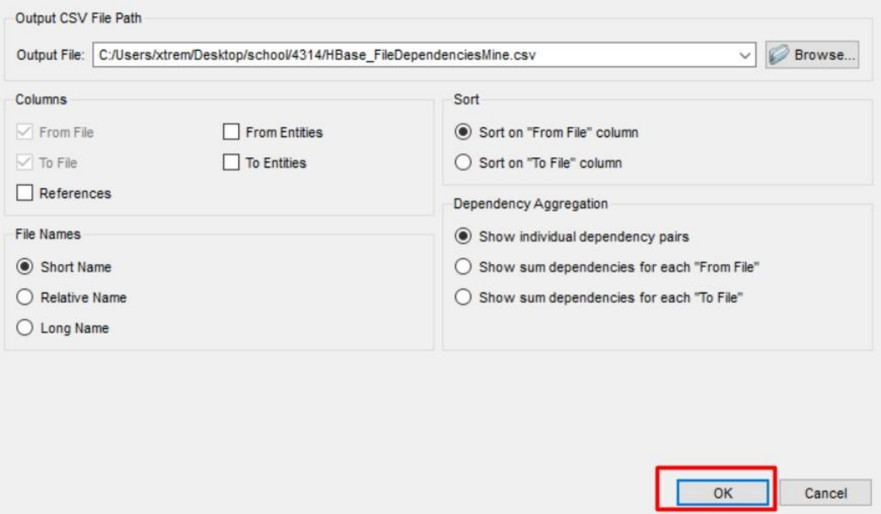
● Sort on "From File" column
○ Sort on "To File" column

Dependency Aggregation

● Show individual dependency pairs
○ Show sum dependencies for each "From File"
○ Show sum dependencies for each "To File"

For comparison, needed short name (file name), from file and to file only

Extracting Dependencies



Output CSV File Path

Output File: C:/Users/xtrem/Desktop/school/4314/HBase_FileDependenciesMine.csv

Columns

- ☑ From File
- ☑ To File
- ☐ References
- ☐ From Entities
- ☐ To Entities

Sort

- ● Sort on "From File" column
- ○ Sort on "To File" column

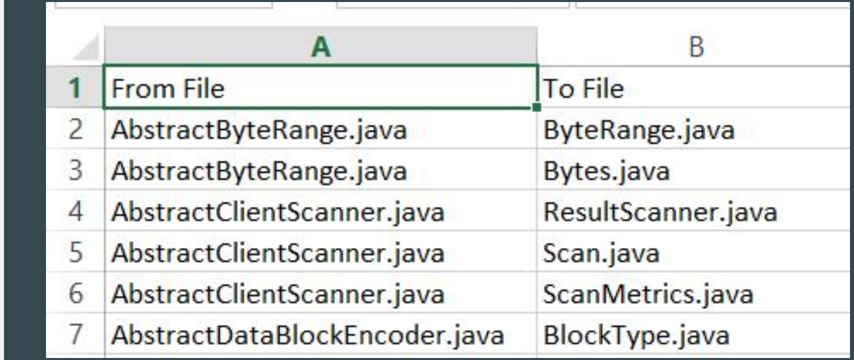
File Names

- ● Short Name
- ○ Relative Name
- ○ Long Name

Dependency Aggregation

- ● Show individual dependency pairs
- ○ Show sum dependencies for each "From File"
- ○ Show sum dependencies for each "To File"

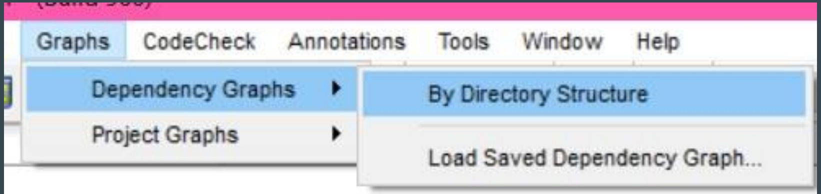
Settings for extracted dependencies are done and confirmed



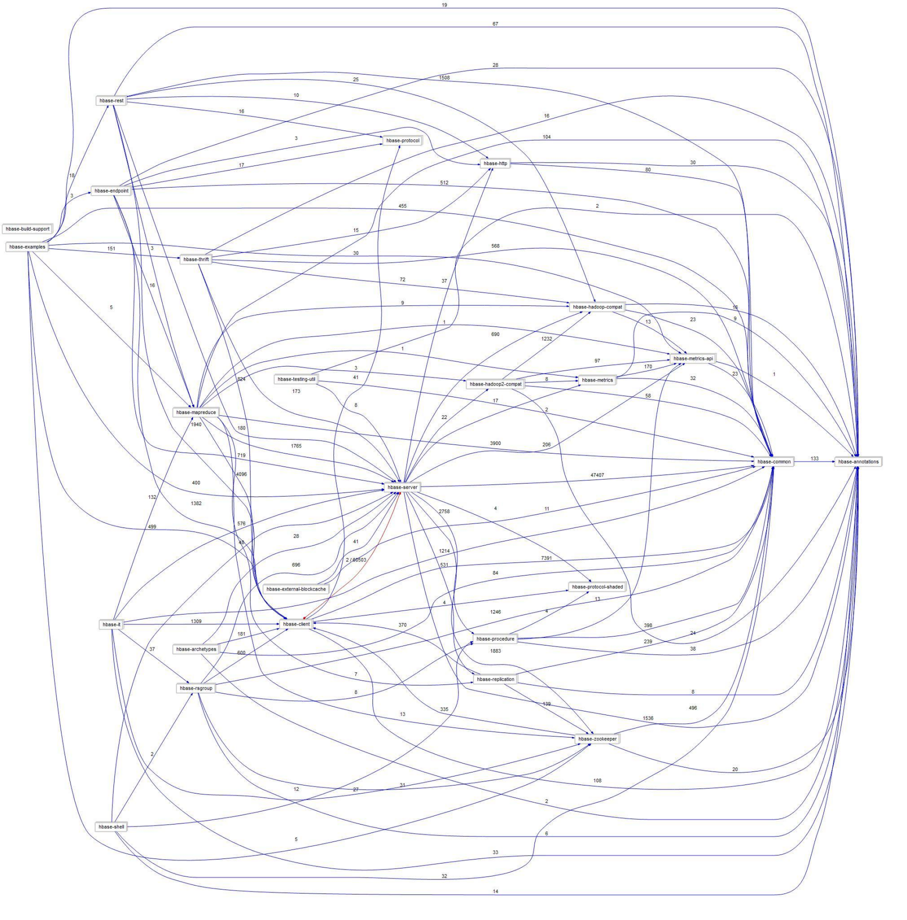
	A	B
1	From File	To File
2	AbstractByteRange.java	ByteRange.java
3	AbstractByteRange.java	Bytes.java
4	AbstractClientScanner.java	ResultScanner.java
5	AbstractClientScanner.java	Scan.java
6	AbstractClientScanner.java	ScanMetrics.java
7	AbstractDataBlockEncoder.java	BlockType.java

Sample of extracted .csv file shown in Excel

Representing Dependencies



Understand has a ‘graph representation of dependencies’ option



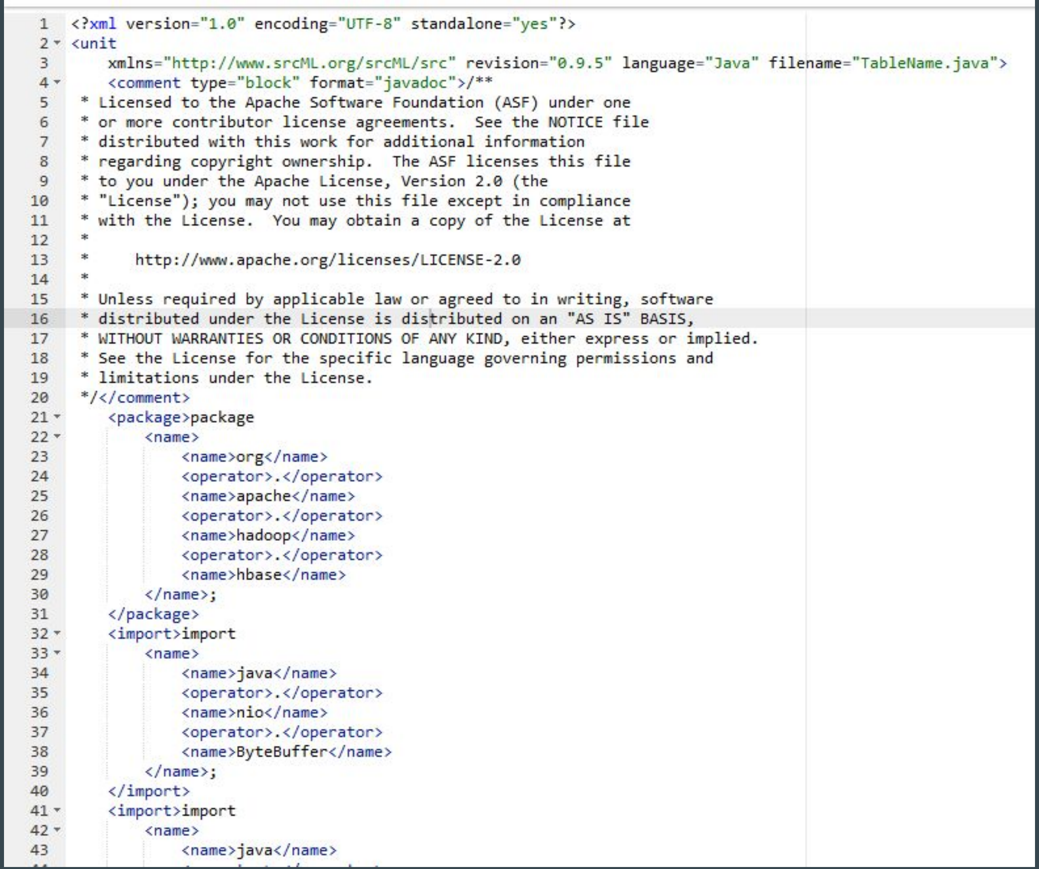
Generated graph dependency of HBase between first-level subdirectories

Advantages and Disadvantages of Scitools: Understand

Advantages	**Disadvantages**
Easy visualization of source code (through treemaps and dependency graphs)	The languages of the source code need to be known
Produces warnings and errors where there might be vulnerabilities in source code	Only has a ‘set of rules’ to detect these vulnerabilities but there may be more
Supports a variety of languages for the legacy/source code	A lot to take in the first time through

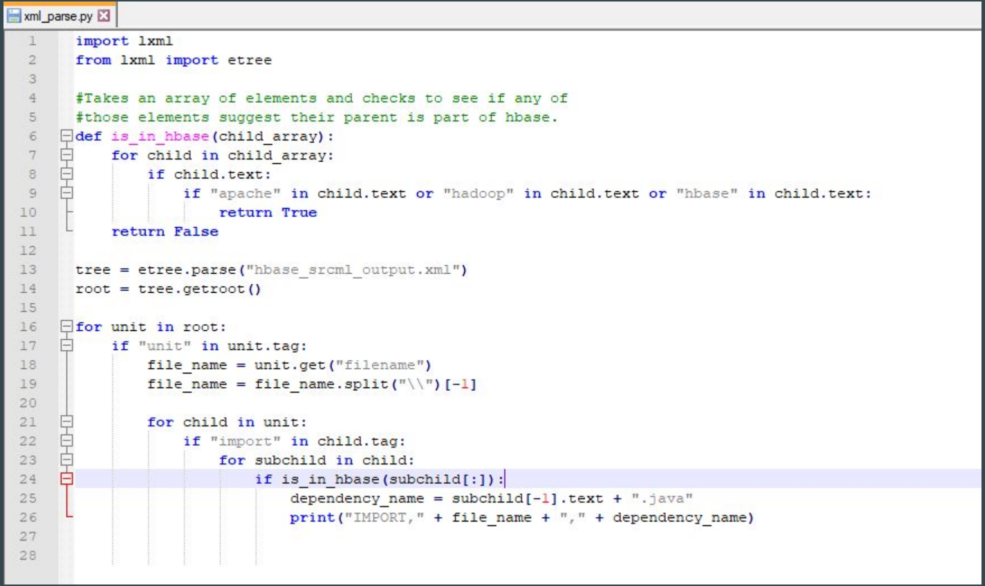
What is srcML

- A tool for analysing and exploring source code
- Takes source code and transforms it to XML format
- Produced a 150MB text file when run on HBase

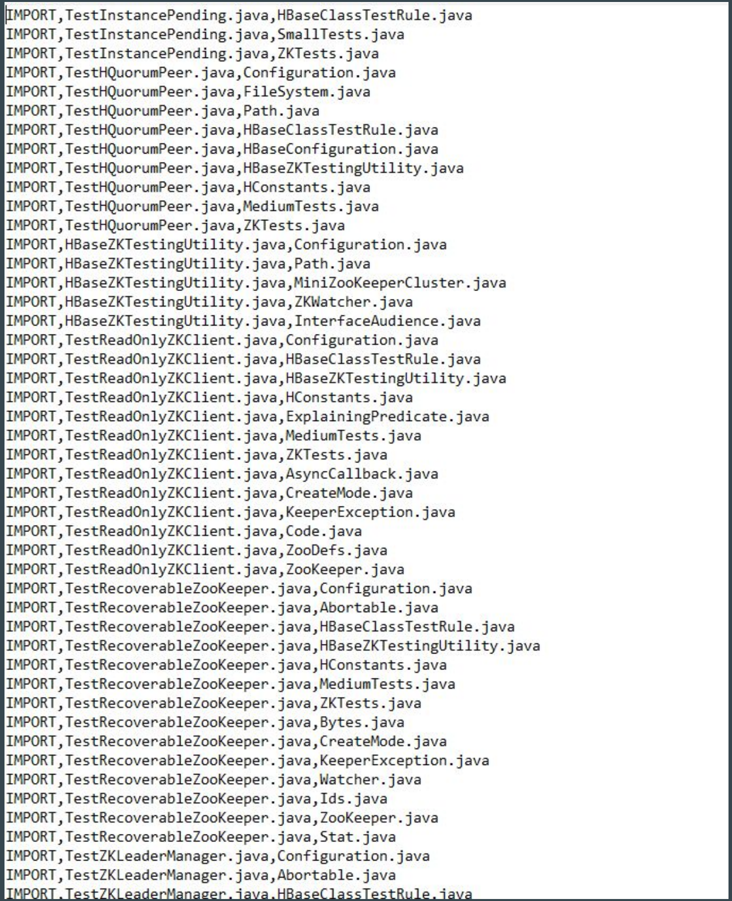


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<unit
    xmlns="http://www.srcML.org/srcML/src" revision="0.9.5" language="Java" filename="TableName.java">
    <comment type="block" format="javadoc">/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements.  See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership.  The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License.  You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */</comment>
    <package>package
        <name>
            <name>org</name>
            <operator>.</operator>
            <name>apache</name>
            <operator>.</operator>
            <name>hadoop</name>
            <operator>.</operator>
            <name>hbase</name>
        </name>;
    </package>
    <import>import
        <name>
            <name>java</name>
            <operator>.</operator>
            <name>nio</name>
            <operator>.</operator>
            <name>ByteBuffer</name>
        </name>;
    </import>
    <import>import
        <name>
            <name>java</name>
```

Extracting Dependencies



```python
import lxml
from lxml import etree

#Takes an array of elements and checks to see if any of
#those elements suggest their parent is part of hbase.
def is_in_hbase(child_array):
    for child in child_array:
        if child.text:
            if "apache" in child.text or "hadoop" in child.text or "hbase" in child.text:
                return True
    return False

tree = etree.parse("hbase_srcml_output.xml")
root = tree.getroot()

for unit in root:
    if "unit" in unit.tag:
        file_name = unit.get("filename")
        file_name = file_name.split("\\")[-1]

        for child in unit:
            if "import" in child.tag:
                for subchild in child:
                    if is_in_hbase(subchild[:]):
                        dependency_name = subchild[-1].text + ".java"
                        print("IMPORT," + file_name + "," + dependency_name)
```



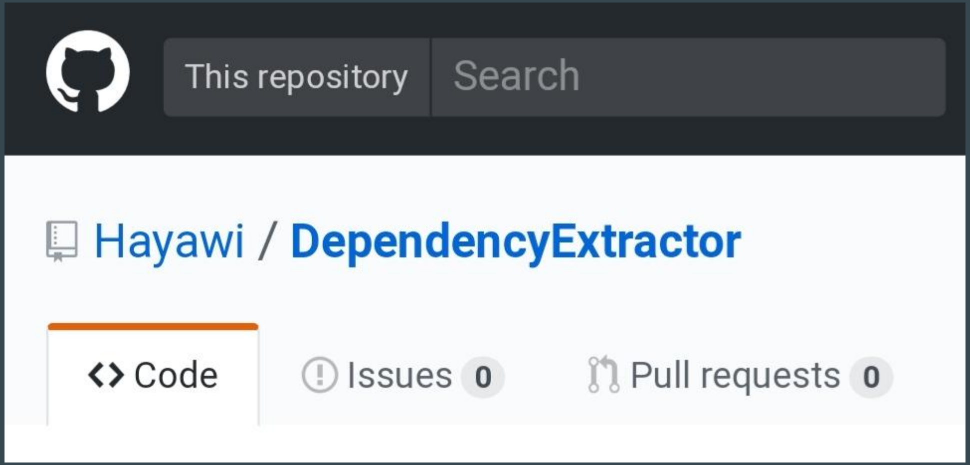
```
IMPORT,TestInstancePending.java,HBaseClassTestRule.java
IMPORT,TestInstancePending.java,SmallTests.java
IMPORT,TestInstancePending.java,ZKTests.java
IMPORT,TestHQuorumPeer.java,Configuration.java
IMPORT,TestHQuorumPeer.java,FileSystem.java
IMPORT,TestHQuorumPeer.java,Path.java
IMPORT,TestHQuorumPeer.java,HBaseClassTestRule.java
IMPORT,TestHQuorumPeer.java,HBaseConfiguration.java
IMPORT,TestHQuorumPeer.java,HBaseZKTestingUtility.java
IMPORT,TestHQuorumPeer.java,HConstants.java
IMPORT,TestHQuorumPeer.java,MediumTests.java
IMPORT,TestHQuorumPeer.java,ZKTests.java
IMPORT,HBaseZKTestingUtility.java,Configuration.java
IMPORT,HBaseZKTestingUtility.java,Path.java
IMPORT,HBaseZKTestingUtility.java,MiniZooKeeperCluster.java
IMPORT,HBaseZKTestingUtility.java,ZKWatcher.java
IMPORT,HBaseZKTestingUtility.java,InterfaceAudience.java
IMPORT,TestReadOnlyZKClient.java,Configuration.java
IMPORT,TestReadOnlyZKClient.java,HBaseClassTestRule.java
IMPORT,TestReadOnlyZKClient.java,HBaseZKTestingUtility.java
IMPORT,TestReadOnlyZKClient.java,HConstants.java
IMPORT,TestReadOnlyZKClient.java,ExplainingPredicate.java
IMPORT,TestReadOnlyZKClient.java,MediumTests.java
IMPORT,TestReadOnlyZKClient.java,ZKTests.java
IMPORT,TestReadOnlyZKClient.java,AsyncCallback.java
IMPORT,TestReadOnlyZKClient.java,CreateMode.java
IMPORT,TestReadOnlyZKClient.java,KeeperException.java
IMPORT,TestReadOnlyZKClient.java,Code.java
IMPORT,TestReadOnlyZKClient.java,ZooDefs.java
IMPORT,TestReadOnlyZKClient.java,ZooKeeper.java
IMPORT,TestRecoverableZooKeeper.java,Configuration.java
IMPORT,TestRecoverableZooKeeper.java,Abortable.java
IMPORT,TestRecoverableZooKeeper.java,HBaseClassTestRule.java
IMPORT,TestRecoverableZooKeeper.java,HBaseZKTestingUtility.java
IMPORT,TestRecoverableZooKeeper.java,HConstants.java
IMPORT,TestRecoverableZooKeeper.java,MediumTests.java
IMPORT,TestRecoverableZooKeeper.java,ZKTests.java
IMPORT,TestRecoverableZooKeeper.java,Bytes.java
IMPORT,TestRecoverableZooKeeper.java,CreateMode.java
IMPORT,TestRecoverableZooKeeper.java,KeeperException.java
IMPORT,TestRecoverableZooKeeper.java,Watcher.java
IMPORT,TestRecoverableZooKeeper.java,Ids.java
IMPORT,TestRecoverableZooKeeper.java,ZooKeeper.java
IMPORT,TestRecoverableZooKeeper.java,Stat.java
IMPORT,TestZKLeaderManager.java,Configuration.java
IMPORT,TestZKLeaderManager.java,Abortable.java
IMPORT,TestZKLeaderManager.java,HBaseClassTestRule.java
```

Advantages and Disadvantages of srcML

Advantages	**Disadvantages**
High level of detail for each class provides more information about interactions between classes	Large initial XML document is not human readable
XPath provides a robust search language for XML documents	Searching for detailed information requires extra programing to use XPath
Works with source code in any language	

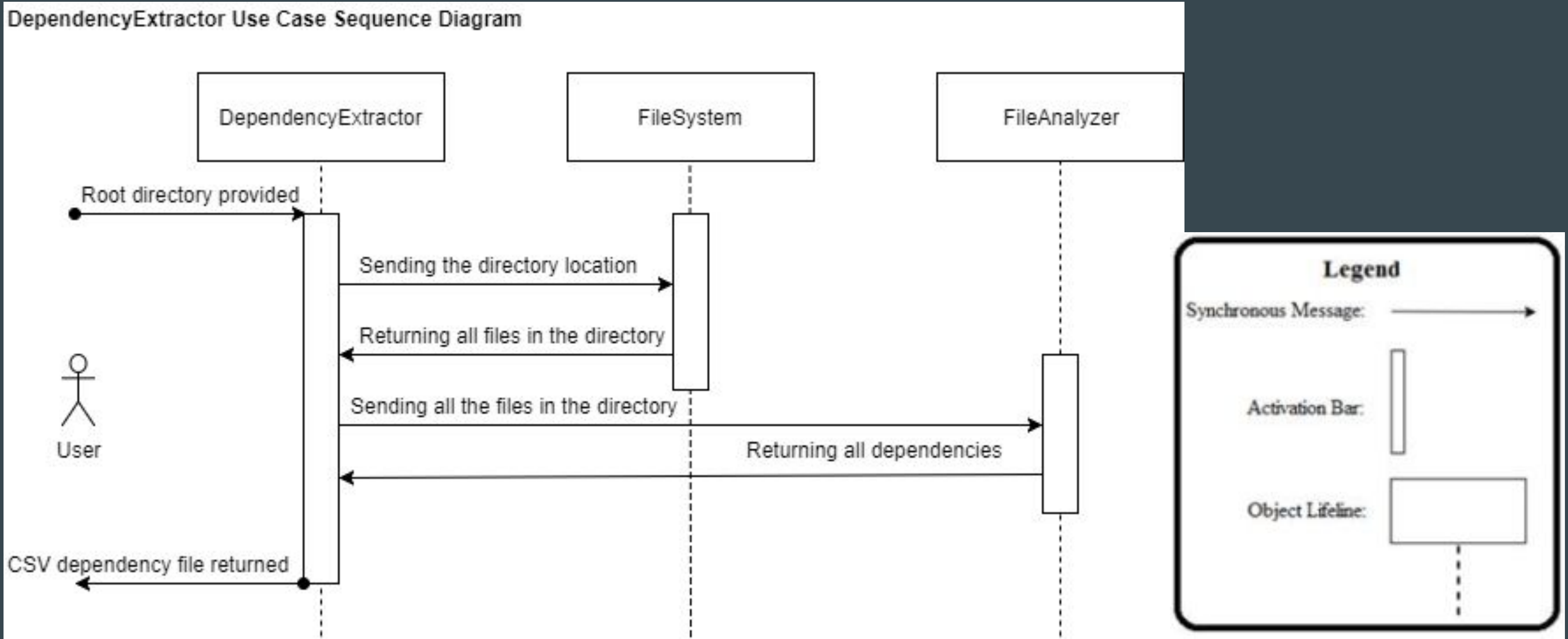
DependencyExtractor (Using Import Statements)

- Java based application that takes in a root directory and scrubs the import statements
- Powerful and limitless, i.e. can be made to do anything
- Extremely time consuming, especially as features expand
- Reinventing the wheel



Hayawi / DependencyExtractor

DependencyExtractor Usecase



DependencyExtractor Use Case Sequence Diagram

DependencyExtractor
FileSystem
FileAnalyzer
Root directory provided
Sending the directory location
Returning all files in the directory
Sending all the files in the directory
Returning all dependencies
User
CSV dependency file returned
Legend
Synchronous Message:
Activation Bar:
Object Lifeline:

Comparison Process - Overview

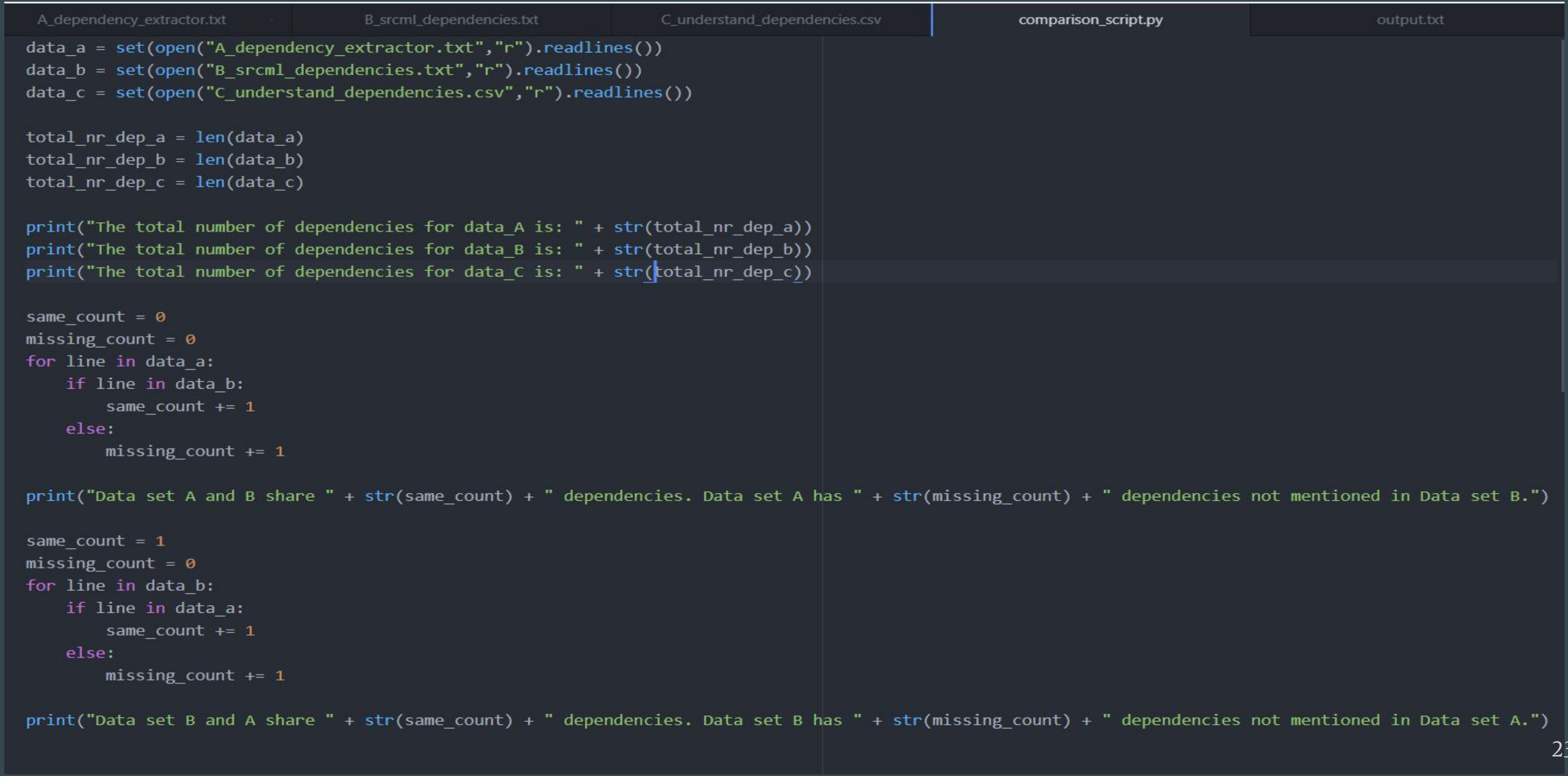
Quantitative Analysis

- Gather data from all dependency tools used
- Format data with excel
- Once all data is in the same format we can begin comparing
- Use script to compare the three sets of data gathered from the tools

Qualitative Analysis

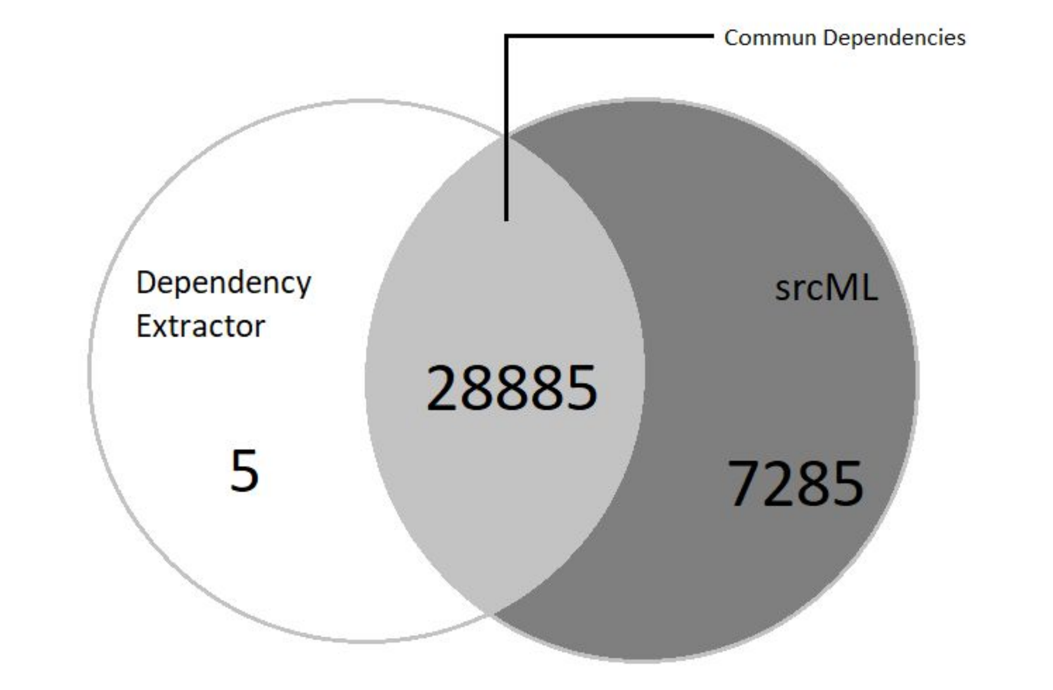
- Use sample size calculator
- Confidence level of: 95%, Confidence interval: 5, Population: 36175, indicating a required sample size of 380 random dependencies
- Use random number generator in excel to shuffle entries
- Take the first 380 samples
- Compare and contrast DependencyExtractor and srcML
- Use Understand as a control group to source conclusions on dependency relevance

Quantitative Analysis - The script



```
data_a = set(open("A_dependency_extractor.txt","r").readlines())
data_b = set(open("B_srcml_dependencies.txt","r").readlines())
data_c = set(open("C_understand_dependencies.csv","r").readlines())

total_nr_dep_a = len(data_a)
total_nr_dep_b = len(data_b)
total_nr_dep_c = len(data_c)

print("The total number of dependencies for data_A is: " + str(total_nr_dep_a))
print("The total number of dependencies for data_B is: " + str(total_nr_dep_b))
print("The total number of dependencies for data_C is: " + str(total_nr_dep_c))

same_count = 0
missing_count = 0
for line in data_a:
    if line in data_b:
        same_count += 1
    else:
        missing_count += 1

print("Data set A and B share " + str(same_count) + " dependencies. Data set A has " + str(missing_count) + " dependencies not mentioned in Data set B.")

same_count = 1
missing_count = 0
for line in data_b:
    if line in data_a:
        same_count += 1
    else:
        missing_count += 1

print("Data set B and A share " + str(same_count) + " dependencies. Data set B has " + str(missing_count) + " dependencies not mentioned in Data set A.")
```

Quantitative Analysis - DependencyExtractor vs srcML



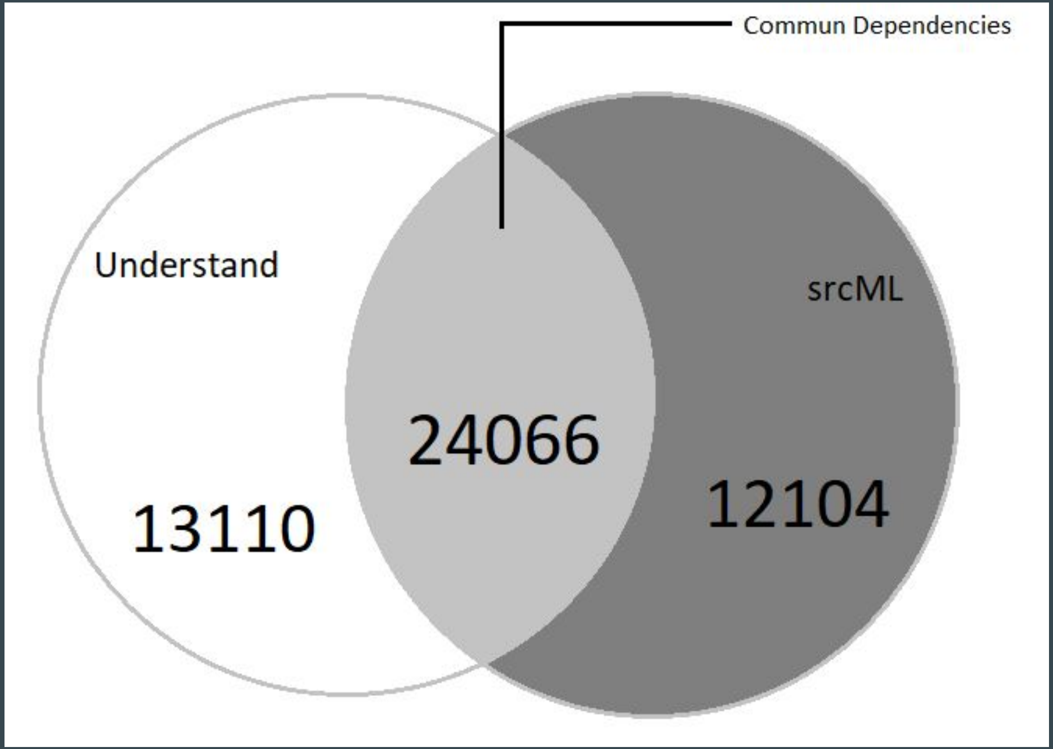
DependencyExtractor:

28890 total dependencies

srcML:

36170 total dependencies

Quantitative Analysis - Understand vs srcML



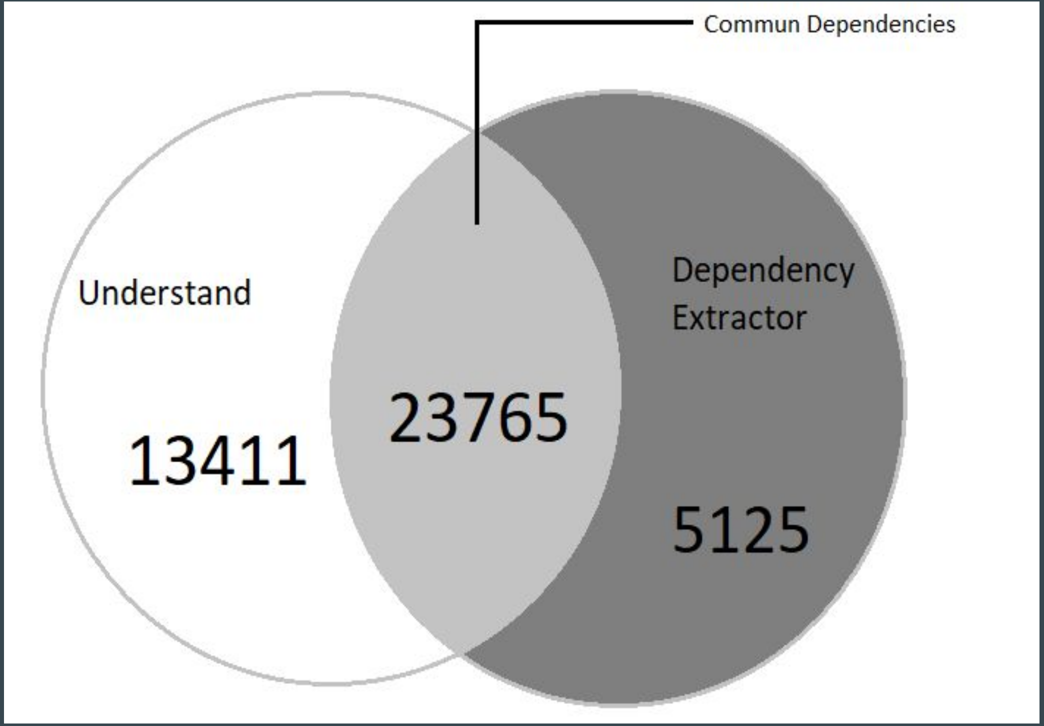
Understand:

37176 total dependencies

srcML:

36170 total dependencies

Quantitative Analysis - Understand vs DependencyExtractor



Understand:

37176 total dependencies

DependencyExtractor:

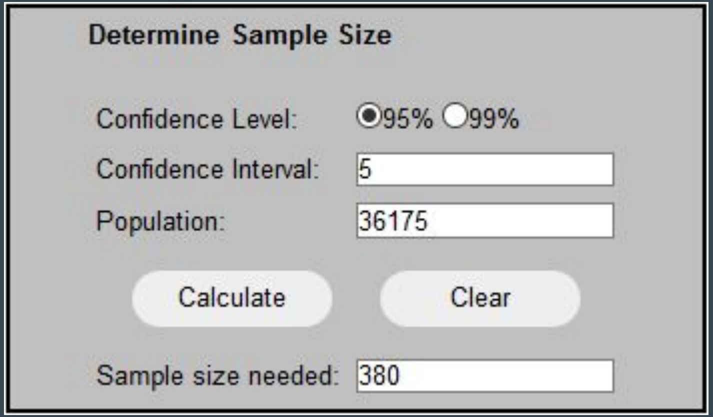
28890 total dependencies

Qualitative Analysis - DependencyExtractor vs srcML

1. Overlap: ~79%
2. DependencyExtractor: ~0.000138%
3. srcML: ~20%

- Large overlap between DependencyExtractor and srcML
- Very few dependencies in DependencyExtractor were unique, while 20% of srcML were unique.

Why? Is srcML returning false positives?



Determine Sample Size

Confidence Level: ● 95% ○ 99%

Confidence Interval: 5

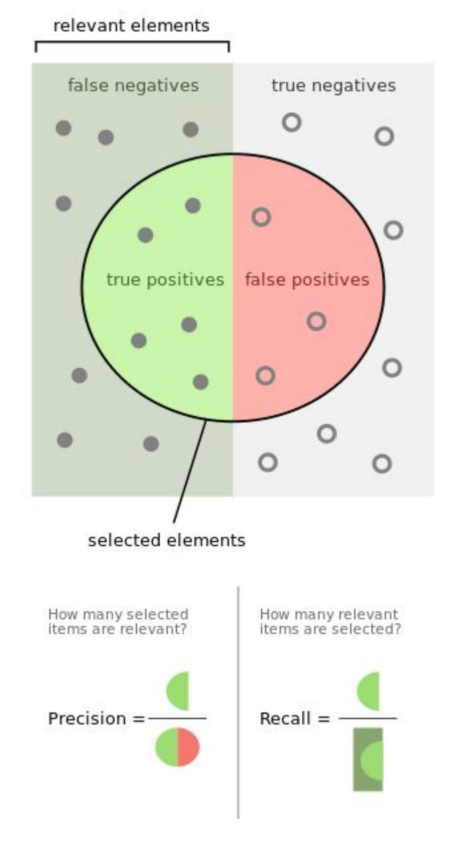
Population: 36175

Sample size needed: 380

Qualitative analysis - Precision and Recall

- Precision (Positive Predictive Value) is the fraction of retrieved instances that are relevant among the retrieved instances.

- Recall (Sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances.



Qualitative Analysis - srcML Precision and Recall

Using Understand as our control group we will calculate precision and recall.

1. Precision: ~66.5%
2. Recall: ~64.7%

- Mediocre Precision and Recall
- srcML seems to be inaccurately returning a large set of incorrect dependencies

Is DependencyExtractor any better?

Qualitative Analysis - DE Precision and Recall

Using Understand as our control group we will calculate precision and recall.

1. Precision: ~82.3%
2. Recall: ~63.9%

- Impressive Precision with mediocre Recall
- It seems DependencyExtractor is doing a better job at extracting accurate dependencies, which explains the odd statistics from before

srcML is retrieving a lot of irrelevant dependencies with it's shotgun approach, leading to an abysmal precision.

Risks and Limitations

- The Dependency Extractor is only as robust as the time spent extracting dependencies (very time consuming if we want to do a perfect job)
- The srcML generates of huge XML file which is not easily human readable and requires a second parsing step to extract desired information.
- Understand needs a properly configured and ordered source code directo so that the representation is accurate
- Since the list of dependencies is so big , some might have been lost when using the extraction softwares
- The sample taken for comparison may not be representative of the entire system

Conclusion (+Lessons Learned)

- Large scale and complex systems:
  - All 3 methods gave us different number of dependencies
  - Unlikely that different extractions methods will give similar results
  - Errors/mistakes possible
- Understand
  - Most comprehensive and inclusive
- srcMl
  - Limited number of dependencies (similar to our own program)
  - Requires parsing and need to format data
- DependencyExtractor
  - Many dependencies limited by inheritance, functions calls etc.
  - Software could have multiple coding languages so would need to account for that