Extracting Software Modules as Communities

Cezar Sas
c.a.sas@rug.nl

@CezarSas

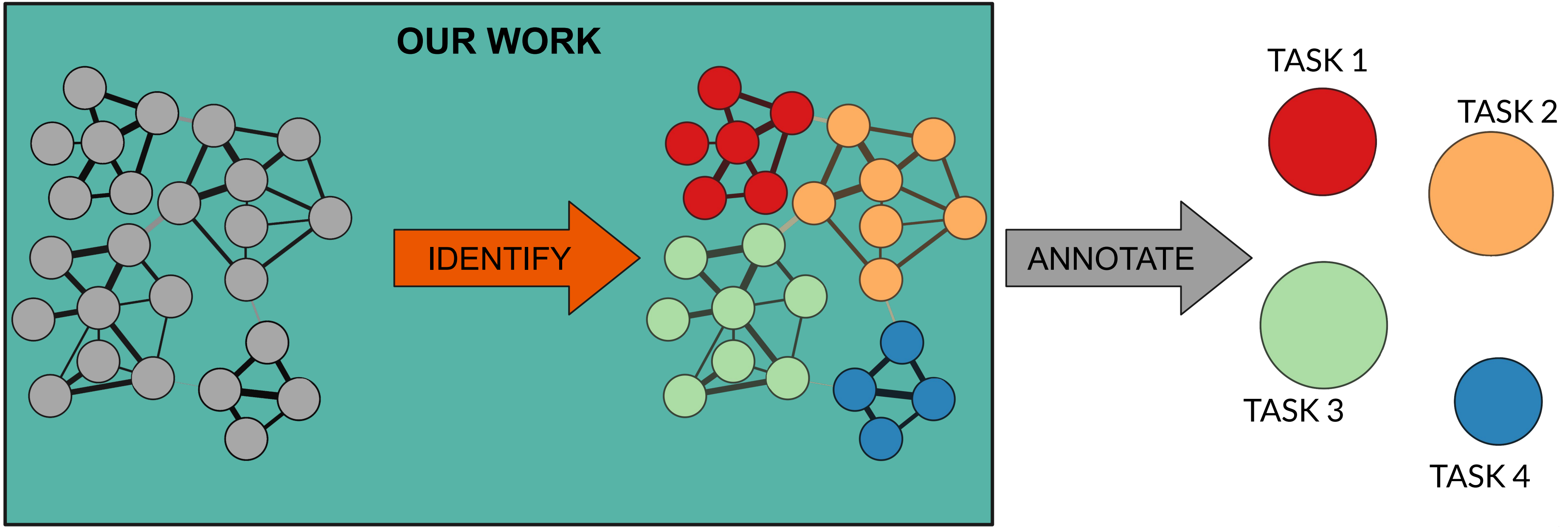
Andrea Capiluppi
a.capiluppi@rug.nl

@DrACapiluppi

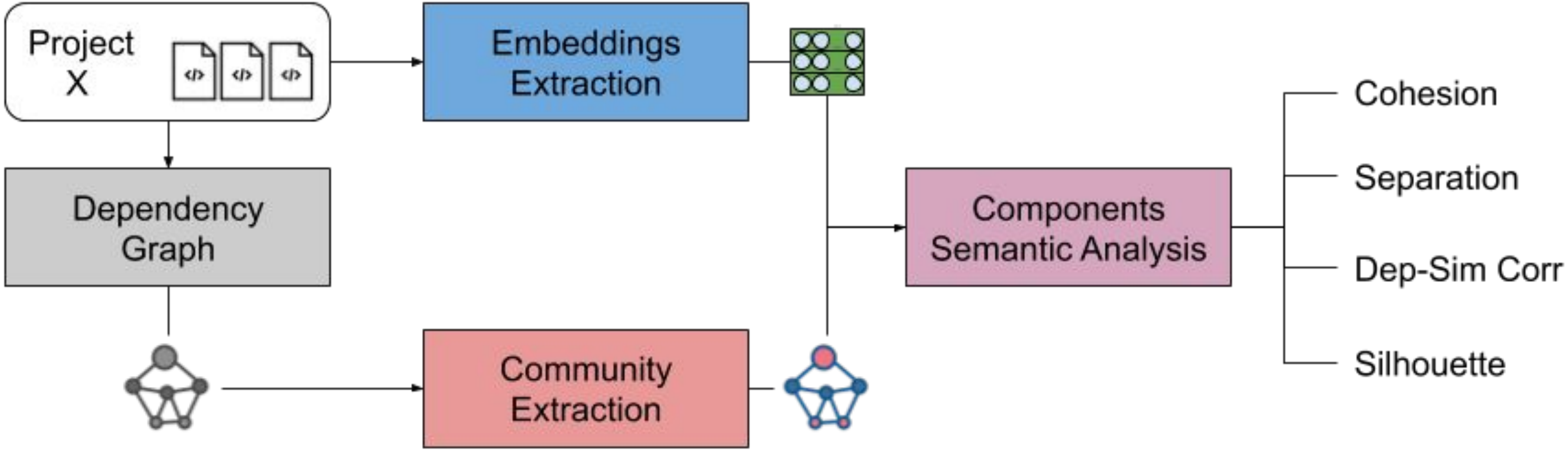
19TH
BELGIUM-NETHERLANDS
SOFTWARE EVOLUTION
WORKSHOP
(**BENEVOL**)

3-4 December 2020

Goal



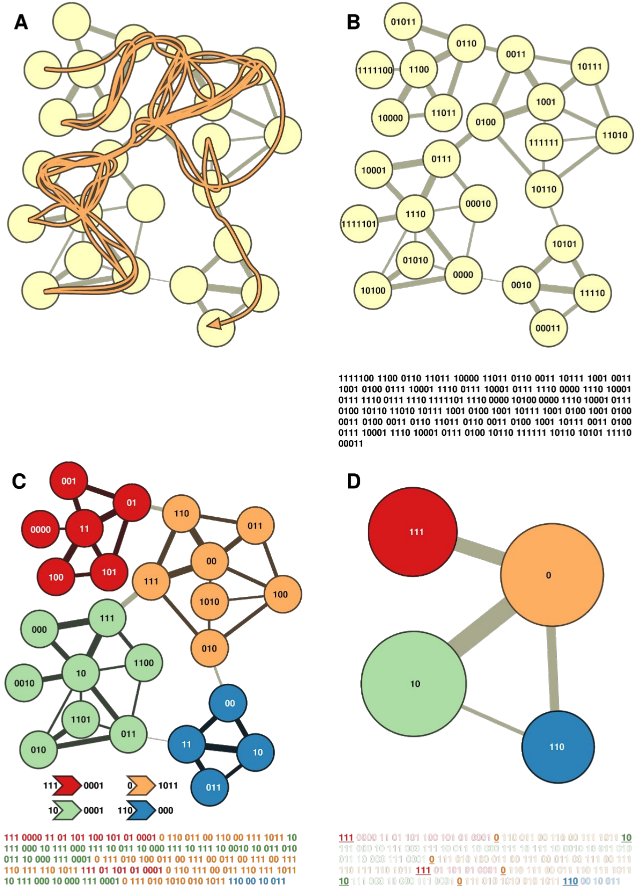
Pipeline


Project X
Embeddings Extraction
Dependency Graph
Community Extraction
Components Semantic Analysis
Cohesion
Separation
Dep-Sim Corr
Silhouette

Infomap

Optimizes **Information Flow**

1. Create random walk
2. Encodes it using Huffman coding
3. Optimize the encoding by using two level codebooks

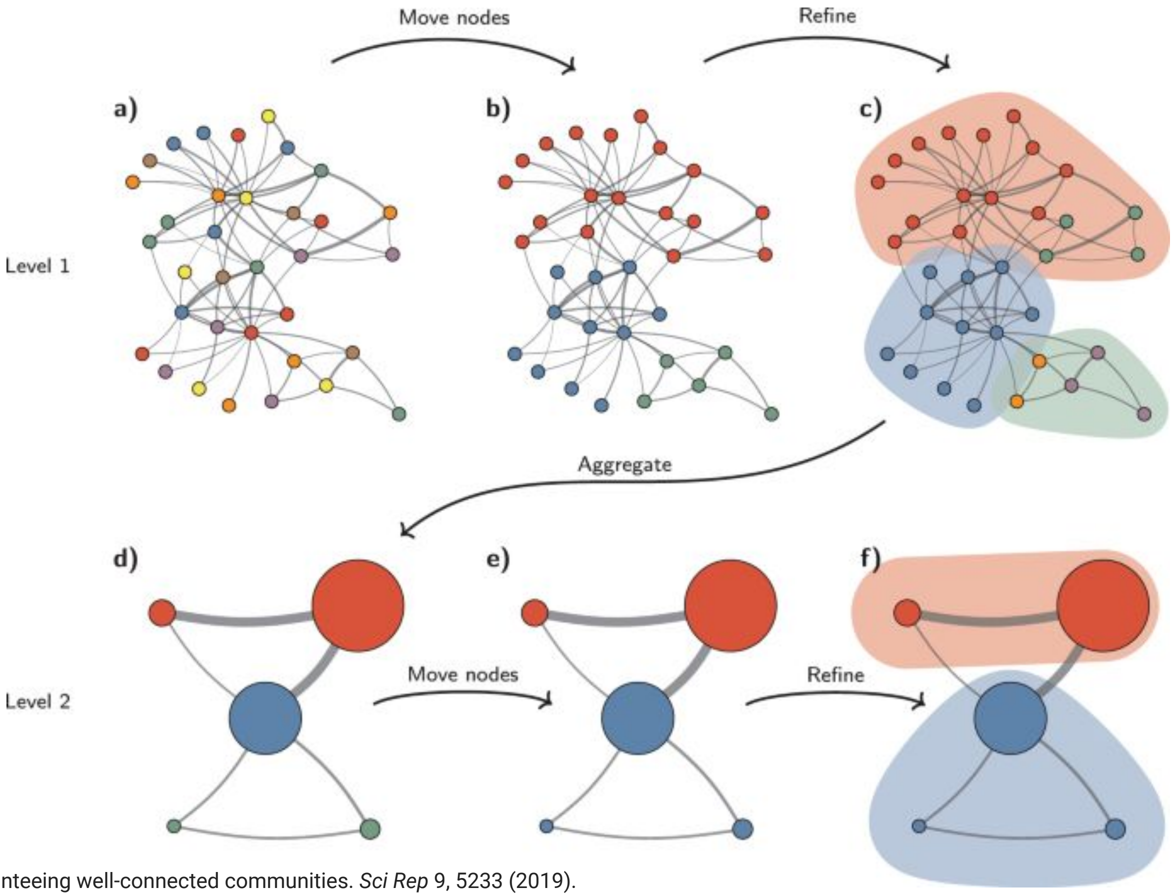


Rosvall, M., Axelsson, D. & Bergstrom, C. The map equation. *Eur. Phys. J. Spec. Top.* 178, 13–23 (2009).
Large Image

Leiden

Optimizes **Modularity**

1. Move nodes between communities to create partitions
2. Refine partitions
3. Aggregate
4. Repeat



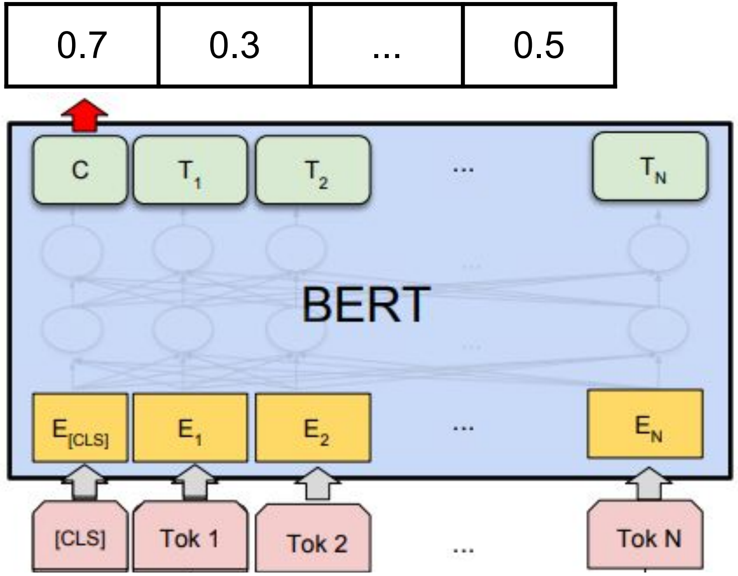
Traag, V.A., Waltman, L. & van Eck, N.J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* 9, 5233 (2019).

TFIDF

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

	Term 1	Term 2	Term ...	Term N
Doc 1	0	0.23	...	0
Doc 2	0.40	0.15	...	0
Doc 3	0.1	0	...	0.1

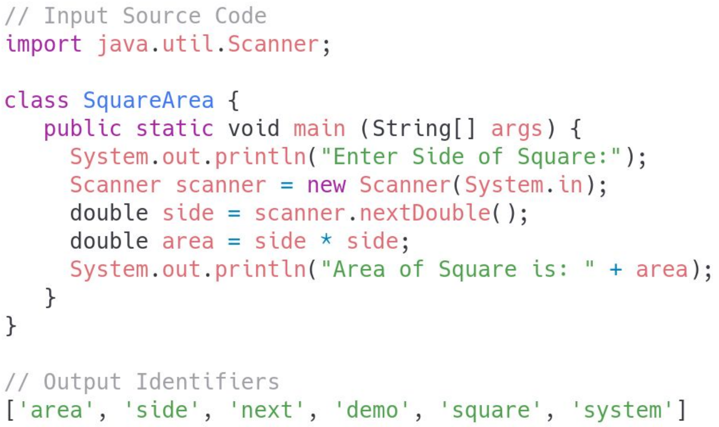
BERT



0.7	0.3	...	0.5

Document Representation

- TF-IDF
  - of identifiers
- BERT embeddings
  - of identifiers
  - of package + class name



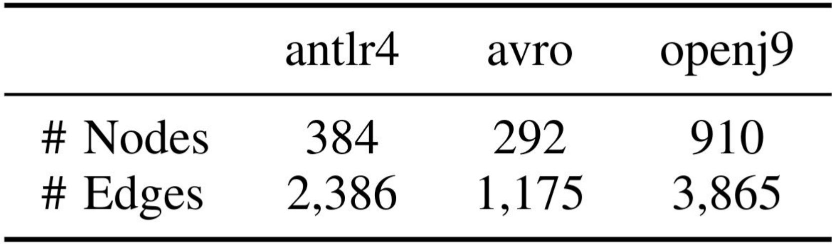
```
// Input Source Code
import java.util.Scanner;

class SquareArea {
   public static void main (String[] args) {
     System.out.println("Enter Side of Square:");
     Scanner scanner = new Scanner(System.in);
     double side = scanner.nextDouble();
     double area = side * side;
     System.out.println("Area of Square is: " + area);
   }
}

// Output Identifiers
['area', 'side', 'next', 'demo', 'square', 'system']
```

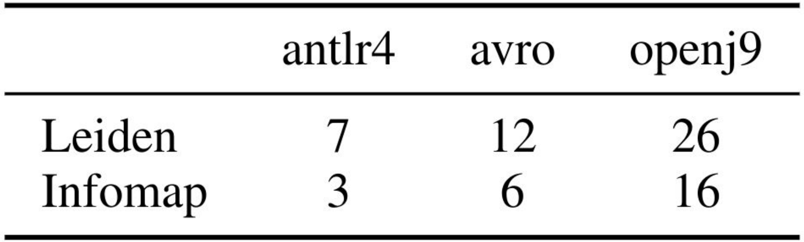
Data

Project Size



	antlr4	avro	openj9
# Nodes	384	292	910
# Edges	2,386	1,175	3,865

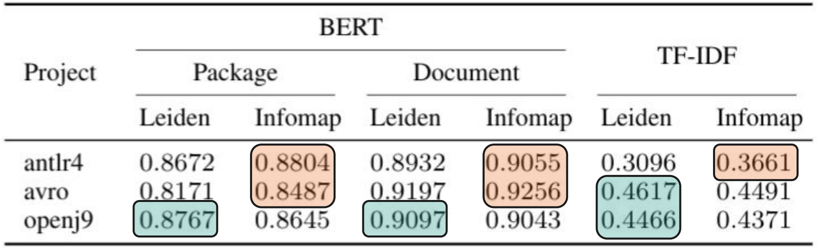
Extracted Communities



	antlr4	avro	openj9
Leiden	7	12	26
Infomap	3	6	16

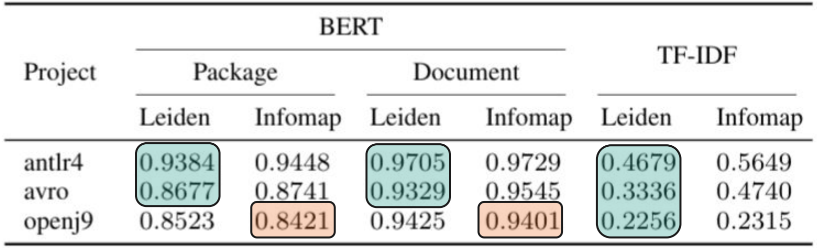
Evaluation 1/2

TABLE III: Average cohesion of components



Project	BERT				TF-IDF	
	Package		Document			
	Leiden	Infomap	Leiden	Infomap	Leiden	Infomap
antlr4	0.8672	0.8804	0.8932	0.9055	0.3096	0.3661
avro	0.8171	0.8487	0.9197	0.9256	0.4617	0.4491
openj9	0.8767	0.8645	0.9097	0.9043	0.4466	0.4371

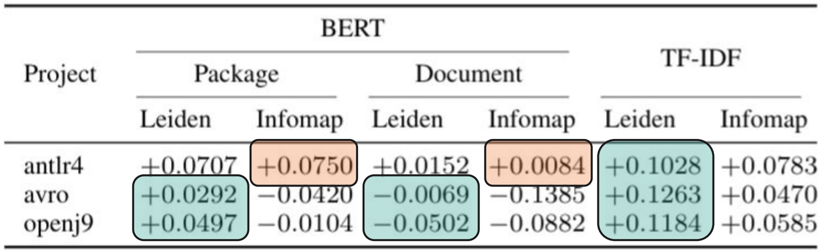
TABLE IV: Average similarity between components



Project	BERT				TF-IDF	
	Package		Document			
	Leiden	Infomap	Leiden	Infomap	Leiden	Infomap
antlr4	0.9384	0.9448	0.9705	0.9729	0.4679	0.5649
avro	0.8677	0.8741	0.9329	0.9545	0.3336	0.4740
openj9	0.8523	0.8421	0.9425	0.9401	0.2256	0.2315

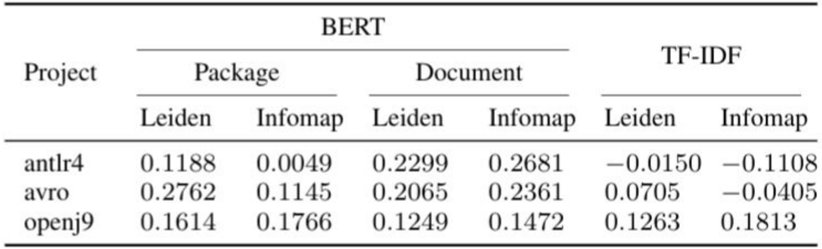
Evaluation 2/2

TABLE V: Silhouette scores for the extracted communities



Project	BERT				TF-IDF	
	Package		Document			
	Leiden	Infomap	Leiden	Infomap	Leiden	Infomap
antlr4	+0.0707	+0.0750	+0.0152	+0.0084	+0.1028	+0.0783
avro	+0.0292	−0.0420	−0.0069	−0.1385	+0.1263	+0.0470
openj9	+0.0497	−0.0104	−0.0502	−0.0882	+0.1184	+0.0585

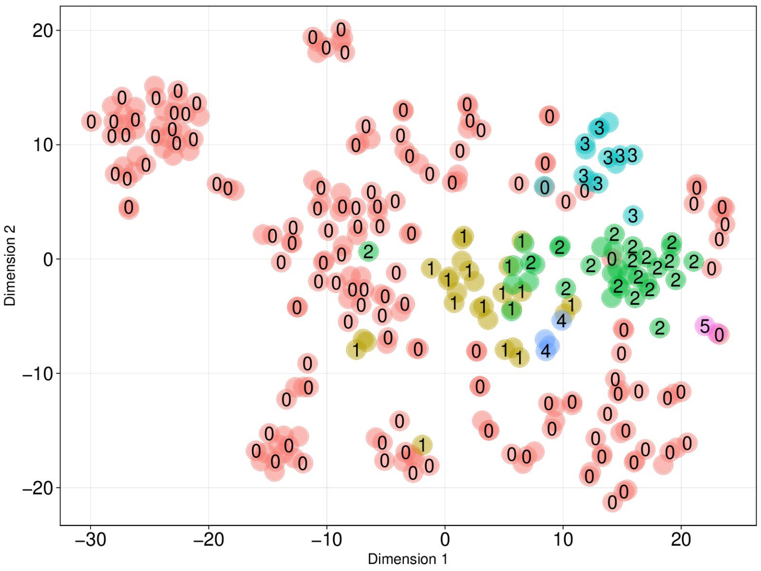
TABLE VI: Pearson's r for the number of dependencies between components and the semantic similarity.



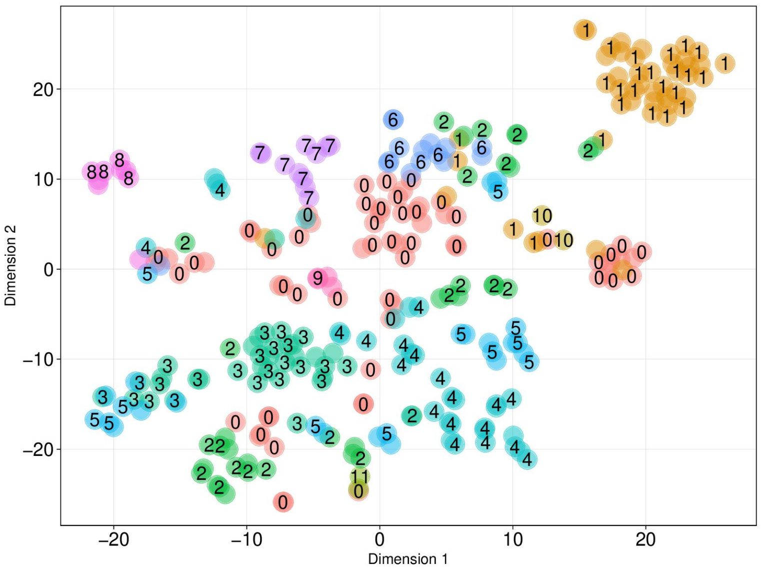
Project	BERT				TF-IDF	
	Package		Document			
	Leiden	Infomap	Leiden	Infomap	Leiden	Infomap
antlr4	0.1188	0.0049	0.2299	0.2681	−0.0150	−0.1108
avro	0.2762	0.1145	0.2065	0.2361	0.0705	−0.0405
openj9	0.1614	0.1766	0.1249	0.1472	0.1263	0.1813

Visualization - TSNE of Avro's TFIDF

Infomap



Leiden



Conclusions

- **Leiden**:
  - Less cohesive
  - Better separated
  - Better clustered components
  - Lower dependency on similar components
- **Infomap**:
  - More cohesive
  - Slightly overlapping clusters
  - Higher dependency on similar components

Future Work

- Increase the Sample
- Qualitative Evaluation
- Components Classification