#1

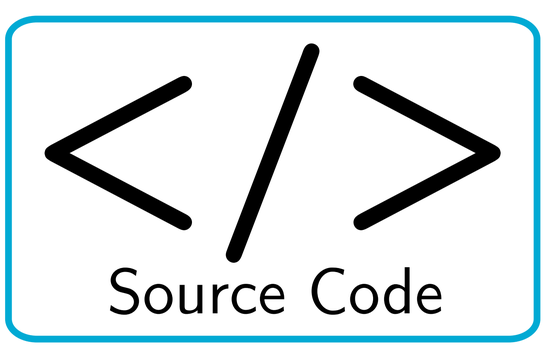
Software Clusterings with Vector Semantics and the Call Graph

Marios Papachristou
papachristoumarios@gmail.com
Business Analytics Lab (BALab), Athens University of Economics and Business
National Technical University of Athens
Advisor: Prof. Diomidis Spinellis

1.GOAL

Combine vector semantics and the call graph in order to produce module-level clusterings for software architecture recovery.
We will study the Linux Codebase and compare it against baselines and state-of-the-art.

2.METHOD



Source Code Preprocessing

1. Remove stopwords
2. Split identifiers
3. Lemmatize

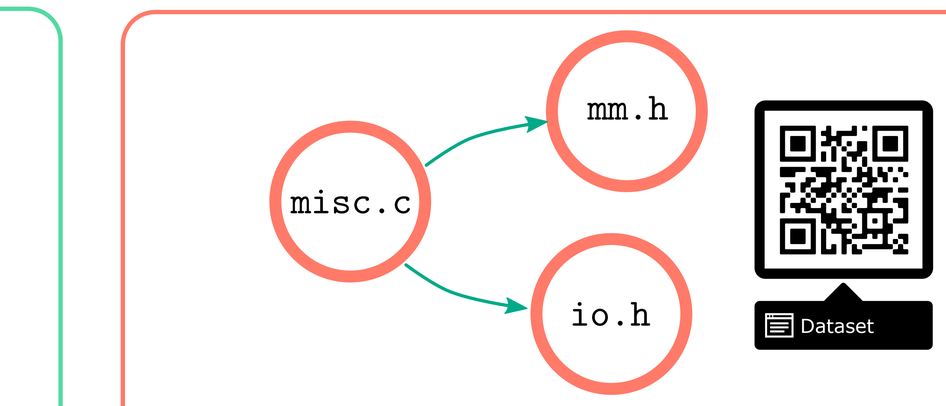
`zone_seqlock_init` → `in progress` → `literals`

Static Graph Analyzer (CScout)

Refererences

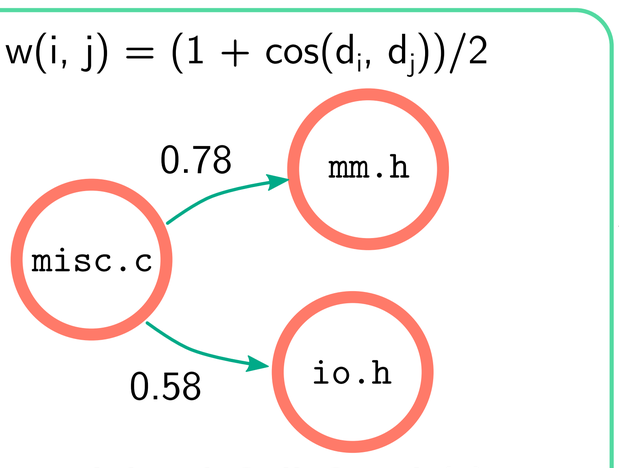
SourceCode — https://github.com/papachristoumarios/sade

Paper

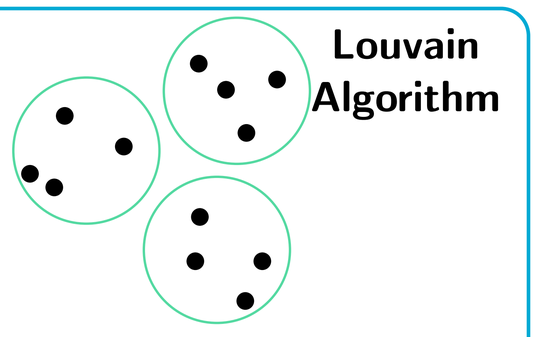


Document Embeddings

Call Graph(s)



Weighted Call Graph(s)

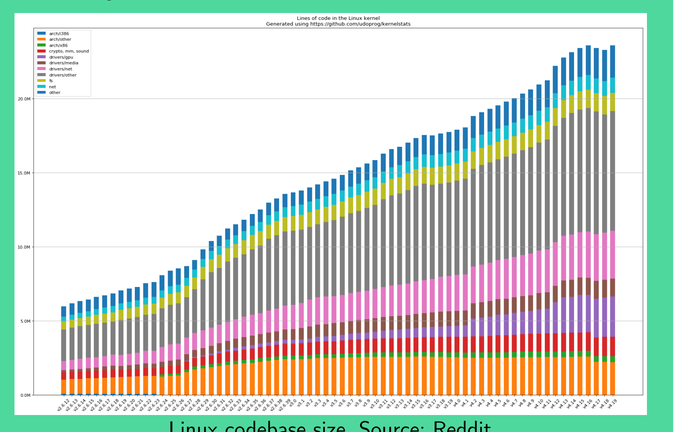


Clusters

3.EVALUATION

Choosing Linux for Evaluation

1. Large and complex system with more than 27 years of continuous development
2. HUGE(!) project consisting of 20.3 million lines of source code
3. Easy to establish ground truth, due to clear directory structure



Linux codebase size. Source: Reddit

The clusters we have used are the first-level directories. The embeddings are generated at the one-top directories. That means e.g. that the file drivers/net/ieee802154/mcr20a.c is groupped under drivers/net/ieee8021154 and belongs to the cluster drivers

4.RESULTS

Results

Algorithm	Dim.	n_c	Range	$\bar{x}$	σ	Median	MoJo
ACDC	–	9055	1 – 4245	5	46	2	33694
Average Linkage	*300*	*21*	1–3406	163	725	1	2092
Complete Linkage	*300*	*21*	1–1529	163	412	19	1710
LIMBO [1]	12317	*21*	50–1810	163	375	50	1482
Ward Linkage[2]	*300*	*21*	21–948	163	223	70	1138
SADE	*300*	10 (± 2)	2 (± 0) -132 (± 13)	64 (± 4)	40 (± 4)	65 (± 10)	243 (± 1)
SADE (Directed)	*300*	5 (± 2)	1 (± 1) - 614 (± 1)	141 (± 39)	253 (± 25)	2 (± 0.3)	237 (± 2)
Ground Truth	–	21	1–1348	163	341	11.0	–

MoJo Distance

Minimum number of operations to convert one clustering to another, where the only valid operations are:

1. **Move** a component between two clusters or create a new cluster.
2. **Join** two components together to a bigger cluster

Analysis of Results

Extremity

Generate reasonable cluster sizes (not too large and not too small)

Authoritativeness

Show great improvement in terms of MoJo clustering distance. Our approach also found a number of clusters near the ground truth without prior knowledge of the number of clusters

Conclusions

1. Suggest further usage of vector semantics in software clustering methods
2. Show significant improvement in terms of closeness to the ground truth, overperforming the other methods.
3. Production of balanced clusters

The research described has been carried out as part of the CROSSMINER Project which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 732223