License Confusion on GitHub

Yannik Schmidt

September 4, 2018

Overview

- Goals
- Perquisites
- License recognition
- Results
- Interpretation of the results
- Errors in the results

Goals

- How many license conflicts are there?
- What licenses are affected?
- How big is the impact on the individual project?

Software Licenses (1)

Licenses

- grant rights
- may have certain conditions
- huge differences in the restrictiveness of individual licenses

Software Licenses (2)

Do What the Fuck You Want To Public License (WTFPL)

- DO WHAT THE FUCK YOU WANT TO.

GPL

- must retain copyright header
- must retain license (copyleft)
- and more...

Software License Conflicts

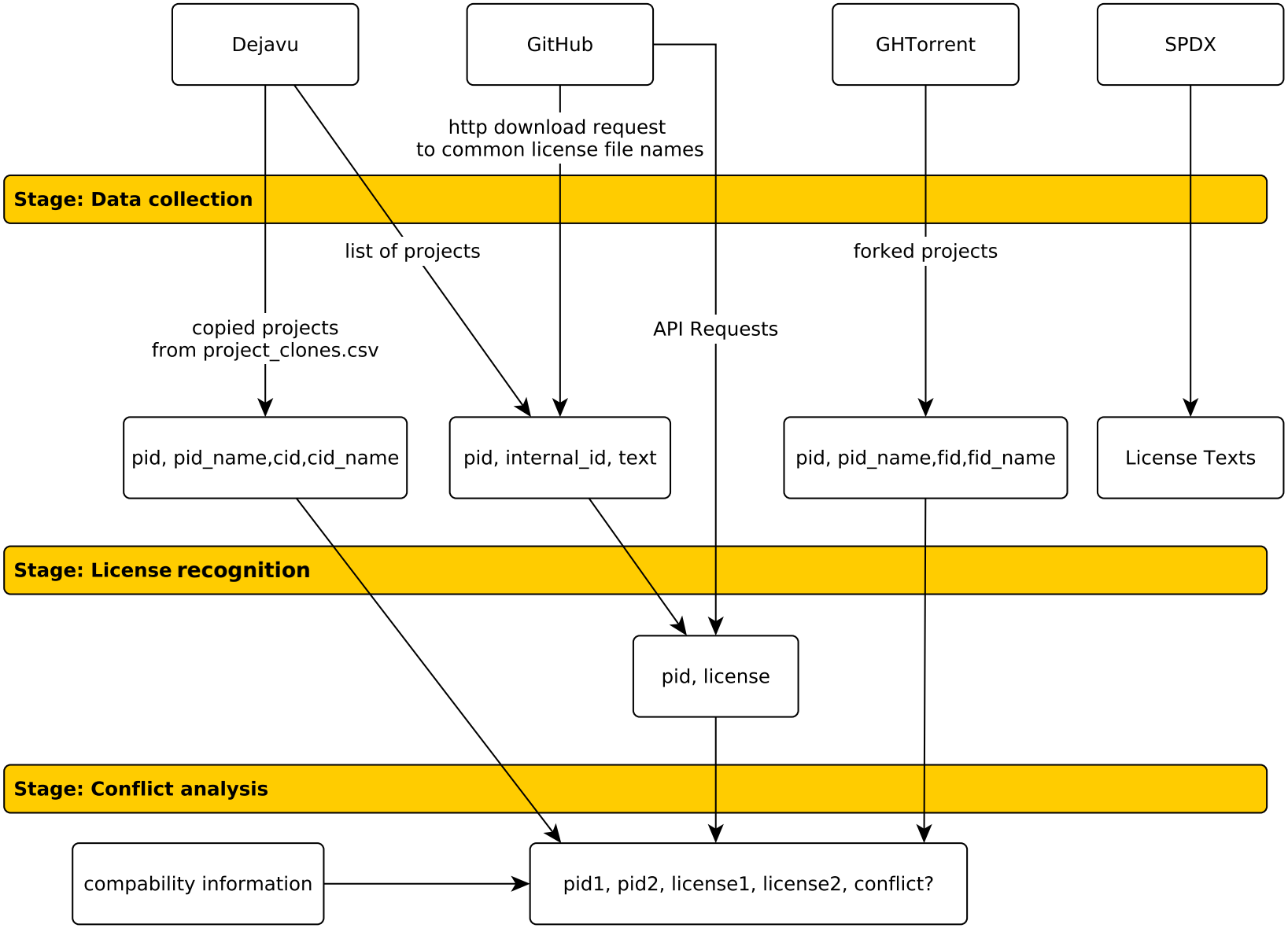
- Project A - WTFPL *(permissive license)*
- Project B - GPL *(strong copyleft license)*
- Project C - MIT License *(permissive license)*

- Project B includes Project A $\rightarrow$ *no conflict*
- Project A includes Project B $\rightarrow$ *conflict*
- Project C includes Project A
  and Project A includes Project B $\rightarrow$ *(indirect) conflict*

Necessary Steps

- Find projects that include another or are copy of another project
- Identify the licenses of those projects
- Identify if the specific combination of licenses creates conflict

Overview of the pipeline



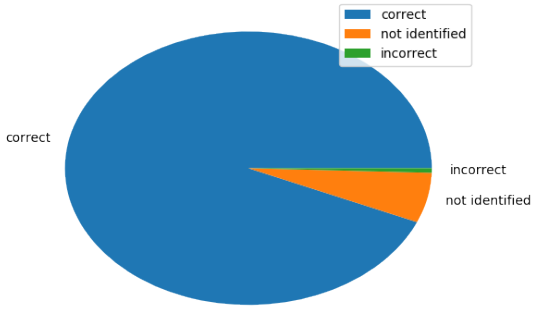
The Dejavu Dataset

- analysis of copied code/projects on GitHub
- organized as CSV
- publicly available

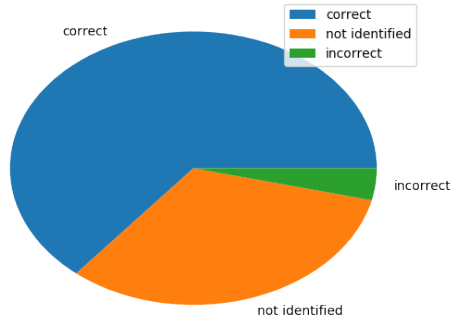
How to identify licenses

- Character or word distance
- Hashing
- **L**ocality **S**ensitive **H**ashing (LSH)
- Unique subsequences
- LSH with unique subsequences as fallback

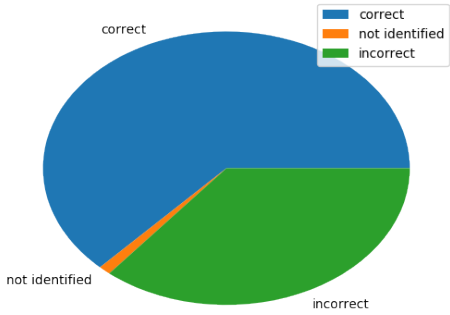
Accuracy of identification



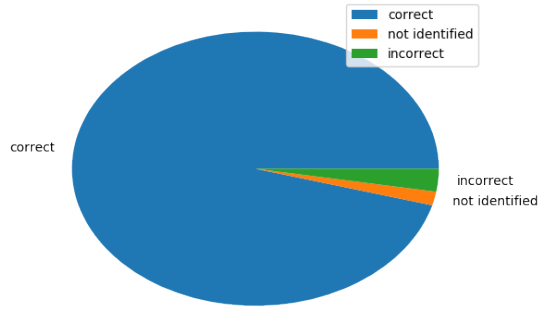
(a) unique subsequence search



(b) “licensecheck“ package

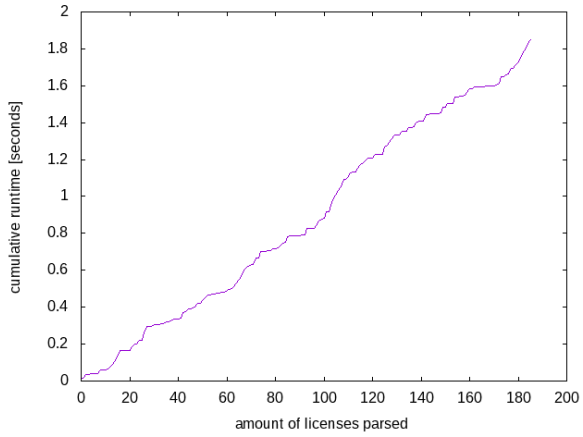


(c) keyword search with LSH and normal hashing

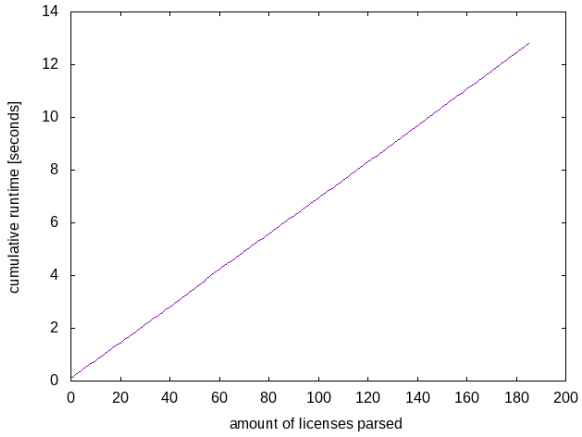


(d) unique subsequence search with LSH and normal hashing

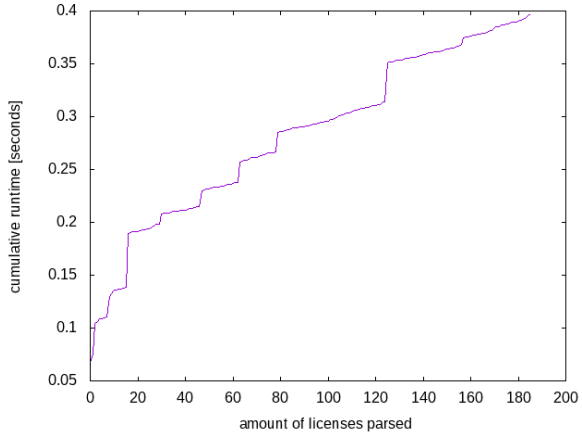
Performance of identification



(a) unique subsequence search



(b) keywords with normal hashing



(c) unique sub sequence approach with LSH and normal hashing

Determining Conflicts (1)

license	**type**	**copyleft**	**sub license as**	**GPL compatible**
MIT	permissive	no	all	yes
WTFPL	permissive	no	none	yes
GPL	protective	yes	none	yes
LGPL	protective	yes	none	yes
no license	restrictive	n/a	none	no
...	...	...	...	...

Determining Conflicts (2)

- projects have the same license $\leftarrow$ *no conflict*
- any type includes permissive $\leftarrow$ *no conflict*
- protective includes GPL compatible $\leftarrow$ *no conflict*
- everything else $\leftarrow$ *conflict*

Results (1)

	Python	C++	Java	JavaScript
none found	664994 (73%)	278315 (75%)	1361427 (91%)	675703 (79%)
identified	235015 (26%)	87011 (23%)	115125 (8%)	166118 (19%)
not identified	9281 (1%)	4114 (1%)	4916 ($<$ 1%)	4633 ($<$ 1%)
total	909290	369440	1481468	846454

Table: Success rates of license search (stage 1) and recognition (stage 2).

	Python	C++	Java	JavaScript
conflicts	12272	9745	1892	23280
total projects	909290	369440	1481468	846454
total copies	27362564	35925821	10169471	130000000
conflicts/projects	1.34%	2.64%	0.13%	2.75%
conflicts/copies	0.045%	0.027%	0.019%	0.017%

Table: Amount of conflicts in relation to amount of projects and amount of copy-relations.

Results (2)

	Python	C++	Java	JavaScript
direct conflicts	12272	9745	1892	23280
indirect conflicts	7054	4325	1148	15889
cumulative conflicts	19326	14070	3040	39169
cumulative conflicts/projects	2.1%	3.81%	0.21%	4.63%

Table: Indirect and cumulative conflicts per language.

Evaluation

- Licenses conflicts increase with restrictiveness of license
- JavaScript has the highest conflict to project ratio despite having mostly permissive licenses
- most copies in JavaScript are libraries copied into the project
- more restrictive licenses lead to more conflicts
- amount of conflicts and amount of copies correlate linearly or worse
- average included projects probably vastly underestimated
- JavaScript is probably the most accurate number

Threats to validity

- False legal interpretation of licenses
- Unchecked license requirements
- Unparsable semantic
- Reduced license set
- Project of origin
- Missed license information
- Bad data in the Dejavu data set

Future Work

- run the entire Dejavu data set if the Hardware becomes available
- include data from library/package-managers like NPM
- look more closely at reasons for conflicts
- check additional restrictions, especially transference of copyright