A Large-scale Dataset of (Open Source) License Text Variants

Stefano Zacchiroli
stefano.zacchiroli@telecom-paris.fr
LTCI, Télécom Paris, Institut Polytechnique de Paris
Paris, France

ABSTRACT

We introduce a large-scale dataset of the complete texts of free/open source software (FOSS) license variants. To assemble it we have collected from the Software Heritage archive—the largest publicly available archive of FOSS source code with accompanying development history—all versions of files whose names are commonly used to convey licensing terms to software users and developers.

The dataset consists of 6.5 million unique license files that can be used to conduct empirical studies on open source licensing, training of automated license classifiers, natural language processing (NLP) analyses of legal texts, as well as historical and phylogenetic studies on FOSS licensing.

Additional metadata about shipped license files are also provided, making the dataset ready to use in various contexts; they include: file length measures, detected MIME type, detected SPDX license (using ScanCode), example origin (e.g., GitHub repository), oldest public commit in which the license appeared.

The dataset is released as open data as an archive file containing all deduplicated license files, plus several portable CSV files for metadata, referencing files via cryptographic checksums.

KEYWORDS

dataset, open source, software license, copyright, intellectual property, software engineering, natural language processing

ACM Reference Format:
Stefano Zacchiroli. 2022. A Large-scale Dataset of (Open Source) License Text Variants. In *19th International Conference on Mining Software Repositories (MSR '22), May 23–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3524842.3528491

1 INTRODUCTION

Free/Open Source Software (FOSS) is ubiquitous in modern IT solutions [30]. Its liberal licensing terms allow everyone, including industry players, to reuse, build open, and extend it, subject to conditions that vary from license to license [13, 25].

Many different software licenses exist and are used in public code. Some of those licenses are labeled as proper "open source" by the Open Source Initiative, others (with a significant overlap) as "free software" by the Free Software Foundation, others are neither "open" nor "free" but are applied to software components distributed in source code form (e.g., via GitHub or GitLab) and need to be dealt with when reusing those components. The ecosystem of licensing terms is so varied that industry standards like SPDX emerged to normalize license naming and identifiers [29].

Proper management of such an increasingly complex software supply chain [12] requires being able to deal with license combinations, their potential incompatibility [9], and auditing increasingly large code bases, ideally in an automated way [23].

These real-world needs have motivated over the years several empirical software engineering (ESE) studies on the evolution of open source licensing [4, 15, 32], on the emergence of open source *license variants* and exceptions [16, 33], as well as the development of industry-strength tools to automatically detect and classify (FOSS) licenses [10, 16, 21].

Contributions and use cases. We introduce a large-scale dataset of license files collected from more than 150 million public software origins including public Git repositories (from GitHub and GitLab), FOSS distributions (e.g., Debian), and package manager repositories (e.g., PyPI, NPM). The dataset is comprised of two parts:

(1) the content of 6 482 295 deduplicated license files (or *license blobs* in the following) retrieved from Software Heritage, the largest public archive of software source code,[1] carrying filenames that are commonly used by developers to distribute licensing terms to software recipients (e.g., `COPYING`, `LICENSE`, etc.; see section 2 for details);
(2) mined metadata about license files: length measures, detected MIME type, contained FOSS license detected using ScanCode [18], example origin, oldest and total number of public commits in which the license file appears.

The dataset serves use cases such as: (a) large-scale analyses of open source licensing, including license popularity, variants, and phylogenetics (how FOSS licenses evolve and mutate); (b) training supervised and unsupervised machine learning classifiers for FOSS licenses, which remains an open industry challenge with most state-of-the-art classifiers still relying on manually-tuned heuristics; (c) natural language processing (NLP) analyses and modeling of legal corpora in the semantic domain of software licensing.

Data availability. The dataset [34] is released as open data, together with a replication package to recreate it from scratch. It is available for download from Zenodo at https://doi.org/10.5281/zenodo.6379164 as a `tar` archive containing unique license blobs (deduplicated based on SHA1 checksums) in a sharded directory structure, together with a set of portable CSV files for derived metadata, cross-referenced to license blobs via SHA1 checksums.

The dataset has been around informally[2] since 2019 and recently refreshed for the 2021 release documented in this paper. It has

[1] https://archive.softwareheritage.org, accessed 2022-01-26
[2] https://annex.softwareheritage.org/public/dataset/license-blobs/, accessed 2020-03-23

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9303-4/22/05…$15.00
https://doi.org/10.1145/3524842.3528491

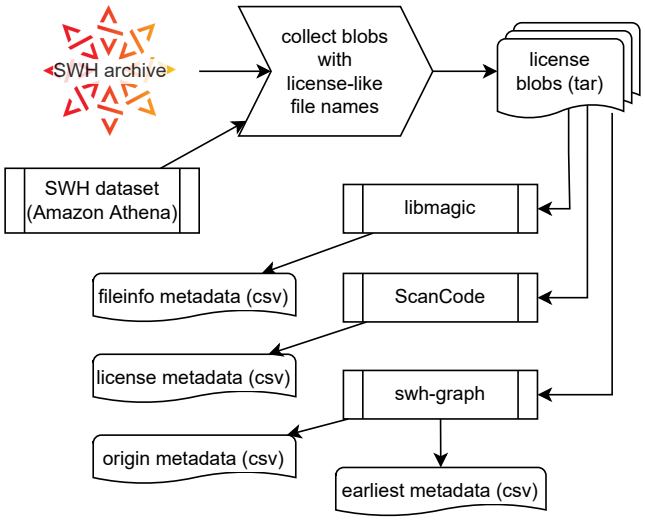

Figure 1: Dataset construction pipeline.

already been used to conduct research internships in computer forensics (applying TLSH hashing [20] to measure license distances) and is currently being used to conduct a large-scale study in open source license phylogenetics.

2 METHODOLOGY AND REPRODUCIBILITY

Figure 1 depicts the methodology used to assemble the dataset.

License files gathering. The first step consists in *selecting* all file blobs archived by Software Heritage and associated to filenames that *are likely* to contain license texts. To that end we used the Software Heritage graph dataset [24] hosted on Amazon Athena (version 20210323) to retrieve the SWHID and SHA1 identifiers of all file blobs associated to file names matching the SQL regular expression: `^([a-z0-9._-]+\.)?(copying|licen(c|s)(e|ing)|notice|copyright|disclaimer|authors)(\.[a-z0-9\._-]+)?$` (the complete SQL query is available as part of the dataset replication package, under the `replication/` directory).

This predicate is quite lax and will end up including files that contain data other than license texts. This was done on purpose, because while it is trivial to filter dataset blobs based on filenames using `fileinfo` metadata (see Section 3), it is cumbersome to *extend* the dataset downstream to add all blobs of interest.

We then retrieved all selected blobs from the Software Heritage archive [7] and archived them in a single `tar` file. (This step was conducted in collaboration with the Software Heritage team, but can be independently reproduced using any archive copy or mirror.)

Metadata mining. All collected blobs have then been mined to gather various types of metadata (see Figure 2, discussed in Section 3). The code we used for mining is available as part of the dataset replication package.

To detect file MIME types and character encodings, we invoked libmagic [31] on each blob via the `python-magic` Python bindings. For files with MIME type starting with `text/` and UTF-8 encoding (or *textual files* in the following for brevity) we computed line and word counts using custom Python code; for all files we computed file sizes in bytes.

The likely licenses contained in each blobs have been detected by running the ScanCode toolkit [18] using its Python API. We run ScanCode with no minimum score threshold—meaning that all detected licenses will be returned, no matter the tool confidence in the result—and with a timeout of 2 minutes (per blob).

Finally, we used the compressed in-memory graph representation [3] of the Software Heritage archive to `origins` and `earliest` metadata. For origins we used the `/randomwalk` API endpoint[3] to traverse the transposed Merkle DAG of the archive and navigate from each blob to a random origin referencing it. ≈12% blobs could not be mapped to an origin this way and lack origin metadata in the dataset.

For earliest commit information we used ad-hoc Java code to navigate the transposed graph from each blob to all commits referencing it, which were counted as the number of occurrences of the blob in the archive. Then we selected the commit with the oldest timestamp among them and extracted its identifier and Unix time. ≈11% blobs could not be mapped to an earliest commit this way and lack earliest metadata in the dataset.

3 DATA MODEL

License files. All 6 482 295 license blobs are shipped in a single `tar` archive file (`blobs.tar.zst`) compressed with Zstandard [5] and weighting 14 GiB. Contained files are organized in a 2-level-deep sharded directory structure based on the SHA1 checksum of each file, e.g., `blobs/02/52/0252d93ad297ec183a567ee813ab8c8d61ece655` for a random file in the archive. Note hence that license files are *fully deduplicated* in the dataset based on SHA1 checksums: each different license blob will appear exactly once in the archive.

The dataset also includes `blobs-sample20k.tar.zst`, a smaller archive containing "only" 20 000 randomly selected license files. It can be used to conduct trial experiments on a small dataset before attacking the entire corpus.

Metadata. License file metadata are provided as a set of textual CSV [27] files, compressed with Zstandard. Each of them corresponds to a table in the relational model shown in Figure 2. They can be used as is or trivially imported into an actual database management system. Metadata can be cross-referenced to the actual license files (in `blobs.tar.zst`) using SHA1 checksums as keys. Each table captures the metadata described below.

`blobs` (CSV file: `license-blobs.csv.zst`) is the master index of all license files (or "blobs") in the dataset. The first column is the Software Heritage persistent identifier (SWHID) [6] of a blob, e.g., `swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2` for a popular variant of the GPL version 3 text; the second the SHA1 checksum of the file. `filename` is the local name given to this license variant *in a given context* (e.g., one or more commits in a public Git repository). This variant of the GPL text is found with 604 different names, including `"COPYING"`, `"LICENSE.GPL3"`, and `"a2ps.license"`. Note that both `swhid` and `sha1` are used by other tables as foreign key targets and that there is no unique primary key in `blobs`, due to multiple filenames associated to each license file.

[3] https://docs.softwareheritage.org/devel/swh-graph/api.html#get--graph-randomwalk--src--dst, accessed 2021-01-25

Figure 2: Relational data model for license blob metadata.

`fileinfo` (`blobs-fileinfo.csv.zst`) provides basic information about license files, cross-referencing them to `blobs` via the `sha1` column. `mime_type` and `encoding` are respectively the file MIME type and character encoding, as detected by libmagic [31]. `size` is the file size in bytes; for textual files, `line_count` and `word_count` report file sizes in lines and (blank-separated) words, respectively.

`scancode` (`blobs-scancode.csv.zst`) reports about the license(s) contained in a given file, as detected by the ScanCode toolkit [18, 21]. Multiple licenses can be detected within a single file, due to either multiple license texts being included or to different confidence levels in the answer reported by ScanCode. For each license file (`sha1` column), `license` reports the license via the associated industry-standard SPDX [8, 29] identifier (e.g., `"GPL-3.0-only"`) and `score` its confidence level as a float in the $[0, 100]$ range (100 being maximum confidence).

`origins` (`blobs-origins.csv.zst`) contains information about where license blobs were found, i.e., which "projects" have distributed them in the past. As each unique license blob can be distributed by tens of million repositories, only a *single example* of an origin is given for each license blob via the `url` field of this table. Obtaining from Software Heritage a list of *all* the projects known to ship a given license blob is possible [26], but out of scope for this dataset. For example, the aforementioned variant of the GPL-3 text was found (among others) in the Git repository at https://github.com/pombreda/Artemis.

`earliest` (`blobs-earliest.csv.zst`) provides historical and popularity information. `earliest_swhid` gives the SWHID of the oldest known public commit that contained the license file, e.g., `swh:1:rev:088313246501c78ae9d7f08e46aaea45855c5c7e` for a variant of the MIT license that includes a Russian copyright notice; `timestamp` is the commit timestamp as Unix time. Referenced commit can be then looked up using the Software Heritage Web UI,[4] API, or filesystem [1]. `occurrences` reports the total number of commits known by Software Heritage as containing the license file; it can be used as a (rough) measure of file popularity.

Table 1: Top-10 words in the license corpus by frequency.

Word	Frequency	Word	Frequency
software	60 539 515	without	22 323 420
license	47 336 592	gplv2	21 486 437
copyright	41 946 018	including	20 824 553
use	28 240 621	nasl	20 746 863
work	23 706 422	notice	19 279 466

4 USING LICENSE (META)DATA

We give below some examples of dataset usage, by conducting preliminary analyses of the license corpus and associated metadata.

Any preliminary analysis of a large textual corpus starts by looking at *word frequencies*. So let's do that. Iterating on all license blobs to tokenize, case-normalize, and count words is left as an exercise for the reader. Assuming a CSV file with ⟨word, frequency⟩ columns is produced at the end, the following Python snippet using Pandas [17] and NLTK [2] will extract the top 100 words in the corpus by frequencies, after removal of English stopwords and single-character tokens.

```
words = pd.read_csv("blobs-wordfreqs.csv") \
  .sort_values(by="frequency", ascending=False)
stop_words = stopwords.words('english') + \
  list(string.digits) + list(string.ascii_lowercase)
interesting_words = words[~words["word")
interesting_words.nlargest(100, columns="frequency")
```

Table 1 provides an excerpt of the results, which correspond to meaningful terms in the semantic domain of open source licensing.

How about *non textual license files*? We can analyze the top detected MIME types using included `fileinfo` metadata:

```
fileinfo = pd.read_csv("blobs-fileinfo.csv")
fileinfo["mime_type"].value_counts()
```

We omit the results for brevity, but they show that 84% of the corpus blobs are `text/plain` and 98% `text/` of some kind (including HTML, XML, and LaTeX). Other interesting (small) classes are rich text formats like RTF as well image files, including PDFs. We have manually verified that at least some of these are actually used to distribute licensing terms; the rest is a small amount of noisy data.

Let's now look at the top *open source licenses detected* in the corpus files. They are trivial to analyze using the ScanCode metadata included in the dataset:

```
scancode = pd.read_csv("blobs-scancode.csv")
scancode["license"].value_counts().nlargest(10)
```

[4] e.g., said Russian MIT variant can be browsed at https://archive.softwareheritage.org/swh:1:rev:088313246501c78ae9d7f08e46aaea45855c5c7e. Accessed 2021-01-25

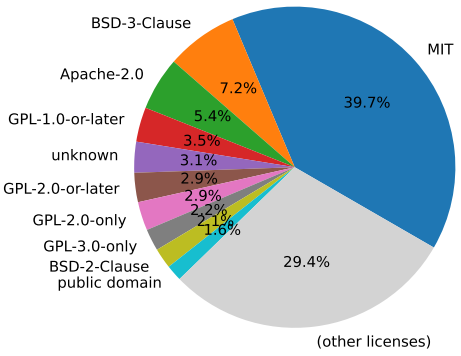

Figure 3: Top licenses in the corpus, as detected by ScanCode.

Note that this is very crude, we are counting all licenses no matter the associated score; at the same time it is easy to verify (e.g., looking at `scancode["score"].describe()`) that the average accuracy is very high, with an average of 93 and a 99% percentile of 100. More accurate analyses, e.g., of only the licenses detected with score 100 would be trivial to conduct.

Results in Figure 3 show that MIT is the most popular open source license variant in the corpus, followed by 3-clause BSD, and Apache 2. Considering that we are counting *license variants* here, MIT at the top makes intuitive sense, because its text also includes a copyright notice which needs to be instantiated by individual authors. Note that this is not a measure of license *popularity*, but interested researchers can obtain insights about that by joining these results with `origin` and `earliest` metadata from the dataset.

5 LIMITATIONS

Internal validity. Datasets built from large amounts of real-world data tend to be noisy and contain bogus data (non-license files, in this case). Rather than thoroughly trying to clean up license blobs incurring the risk of false negatives, we have decided to *augment* them with extra metadata that enable researcher to filter data downstream. We have already observed in Section 2 how to restrict the filename pattern if so desired. Similarly, researchers can filter on MIME types (e.g., if only interested in textual files) or on length metrics (e.g., only keep oneliner files to focus on copyright notices or machine-readable SPDX tags). Study-specific filtering is also best left to dataset users and we provide several types of metadata to support it.

The main inconsistency in the dataset come from the incompleteness of `origin` and `earliest` metadata, which are missing for 11–12% of the dataset blobs. This is due to a version misalignment between the Software Heritage archive and the compressed graph we used for mining these metadata; it could be fixed in the future when a fresher version will become available. Also, due to the ease of forging Git timestamps, some earliest commit metadata are bogus having timestamps set to the UNIX epoch. Both metadata coverage (which remains very high) and timestamp quality can be improved by cross-referencing license blobs to external data sources thanks to the persistent identifiers used in the dataset as keys.

Construct validity. There is no guarantee that all license blobs in the dataset contain license texts considered open/free by OSI/FSF (hence the parentheses around "(open source)" in this paper title), only that they come from public code. If relying on ScanCode as ground truth is acceptable, `scancode` metadata in the dataset can be used for filtering. Otherwise the free/open determination will need to be done independently by dataset users.

Due to selecting license files by filename, *license notices* that *only appear within* source files are underrepresented in the dataset. This applies to, e.g., both the recommended GPL notice "This program is free software [...] under the terms of the GNU General Public License [...]" and SPDX tags [29] like "SPDX-License-Identifier: GPL-3.0-or-later" when they are included only as comments at the beginning of source files. As the dataset is meant to enable studying license *texts*, rather than notices, this is an acceptable limitation. Also, notices are included in the dataset when *also* shipped under license-related filenames. Thoroughly extracting license notices from Software Heritage and including them in the dataset is left as future work.

External validity. By its own nature the dataset provides an incomplete snapshot of reality; as such we do not claim full generality/representativeness of all existing license variants. The reality is a moving target, with new license variants constantly released as public code. The archive we started from is not full-encompassing either. Still, to the best of our knowledge, this is to date the largest, publicly available dataset of (open source) license variants. We plan to mitigate this risk by periodically making available new dataset releases, as we have done up to now.

6 RELATED WORK

The Software Heritage (SWH) graph dataset [24], which we used to select license blobs, is a large dataset underpinning the SWH archive. It stores information analogous to those captured by version control systems (VCS), minus actual file contents. It can be used in conjunction with the dataset presented here, joining information via SWH identifiers.

World of Code [14] is a large dataset and analysis infrastructure, available to researchers to mine public code. It is larger than our initial data source and can be used in conjunction with this dataset to find additional origins/occurrences of licenses blobs of interest. Our dataset is smaller, can be self-hosted, and comes with several relevant metadata precomputed (e.g., ScanCode results).

GHTorrent [11] is a dataset of archived GitHub REST API events. It contains information about public GitHub projects, but as of today does not include the license that GitHub detected as the main license of a given project. (Nor license texts, as source code is out of scope for GHTorrent.)

ScanCode LicenseDB [19] is a public database by the ScanCode authors listing all the licenses they have encountered in the wild during the constant tuning of their detection heuristics. It includes 1879 different *canonical* license texts which are used as comparison reference, but does not provide all variants of them as we do with this dataset; nor it provides associated metadata. Both the Open Source Initiative and the SPDX project maintain analogous public databases [22, 28] covering the canonical texts of, respectively, OSI-approved and SPDX-named licenses, for about ≈500 texts in total.

In summary, this dataset appears to be unique in nature and size, filling an unattended niche. It can also be used in synergy with preexisting datasets about FOSS and public code.

7 CONCLUSION

We have introduced a large-scale dataset of open source license texts. It consists of 6.5 million unique files archived from public code and carrying a name related to software licensing terms. Derived metadata—about file lengths, types, detected open source license in them, and their provenance—are also included in the dataset and trivial to cross-reference with the text corpus.

Future extensions. As future work we intend, on the one hand, to keep the dataset current with the constant evolution of archived public code, gathering license texts from additional data sources. On the other hand we will explore adding to the metadata precomputed text representations of the entire corpus that are commonly needed for natural language processing (NLP) and machine learning analyses, such as word embeddings, latent semantic indexes, and other vectorial text representations.

REFERENCES

[1] Thibault Allançon, Antoine Pietri, and Stefano Zacchiroli. 2021. The Software Heritage Filesystem (SwhFS): Integrating Source Code Archival with Development. In *43rd IEEE/ACM International Conference on Software Engineering: Companion Proceedings, ICSE Companion 2021, Madrid, Spain, May 25-28, 2021*. IEEE, 45–48. https://doi.org/10.1109/ICSE-Companion52605.2021.00032

[2] Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*, Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle (Eds.). The Association for Computer Linguistics. https://doi.org/10.3115/1225403.1225421

[3] Paolo Boldi, Antoine Pietri, Sebastiano Vigna, and Stefano Zacchiroli. 2020. Ultra-Large-Scale Repository Analysis via Graph Compression. In *SANER 2020: The 27th IEEE International Conference on Software Analysis, Evolution and Reengineering*. IEEE.

[4] Matthieu Caneill, Daniel M. GermÃ¡n, and Stefano Zacchiroli. 2017. The Debsources Dataset: Two Decades of Free and Open Source Software. *Empirical Software Engineering* 22 (June 2017), 1405–1437. https://doi.org/10.1007/s10664-016-9461-5

[5] Yann Collet. 2021. RFC 8878 - Zstandard Compression and the "application/zstd" Media Type. https://datatracker.ietf.org/doc/html/rfc8878 Accessed 2022-01-24.

[6] Roberto Di Cosmo, Morane Gruenpeter, and Stefano Zacchiroli. 2018. Identifiers for Digital Objects: the Case of Software Source Code Preservation. In *Proceedings of the 15th International Conference on Digital Preservation, iPRES 2018, Boston, USA*. https://doi.org/10.17605/OSF.IO/KDE56

[7] Roberto Di Cosmo and Stefano Zacchiroli. 2017. Software Heritage: Why and How to Preserve Software Source Code. In *Proceedings of the 14th International Conference on Digital Preservation, iPRES 2017*. https://hal.archives-ouvertes.fr/hal-01590958/

[8] Robin A. Gandhi, Matt Germonprez, and Georg J. P. Link. 2018. Open Data Standards for Open Source Software Risk Management Routines: An Examination of SPDX. In *Proceedings of the 2018 ACM Conference on Supporting Groupwork, GROUP 2018, Sanibel Island, FL, USA, January 07 - 10, 2018*, Andrea Forte, Michael Prilla, Adriana S. Vivacqua, Claudia Müller, and Lionel P. Robert Jr. (Eds.). ACM, 219–229. https://doi.org/10.1145/3148330.3148333

[9] Daniel M. Germán and Massimiliano Di Penta. 2012. A Method for Open Source License Compliance of Java Applications. *IEEE Softw.* 29, 3 (2012), 58–63. https://doi.org/10.1109/MS.2012.50

[10] Robert Gobeille. 2008. The FOSSology project. In *Proceedings of the 2008 International Working Conference on Mining Software Repositories, MSR 2008 (Co-located with ICSE), Leipzig, Germany, May 10-11, 2008, Proceedings*, Ahmed E. Hassan, Michele Lanza, and Michael W. Godfrey (Eds.). ACM, 47–50. https://doi.org/10.1145/1370750.1370763

[11] Georgios Gousios and Diomidis Spinellis. 2012. GHTorrent: Github's data from a firehose. In *9th IEEE Working Conference of Mining Software Repositories, MSR*, Michele Lanza, Massimiliano Di Penta, and Tao Xie (Eds.). IEEE Computer Society, 12–21. https://doi.org/10.1109/MSR.2012.6224294

[12] Nikolay Harutyunyan. 2020. Managing Your Open Source Supply Chain-Why and How? *Computer* 53, 6 (2020), 77–81. https://doi.org/10.1109/MC.2020.2983530

[13] Van Lindberg. 2008. *Intellectual property and open source: a practical guide to protecting code*. O'Reilly Media, Inc.

[14] Yuxing Ma, Tapajit Dey, Chris Bogart, Sadika Amreen, Marat Valiev, Adam Tutko, David Kennard, Russell Zaretzki, and Audris Mockus. 2021. World of code: enabling a research workflow for mining and analyzing the universe of open source VCS data. *Empir. Softw. Eng.* 26, 2 (2021), 22. https://doi.org/10.1007/s10664-020-09905-9

[15] Yuki Manabe, Yasuhiro Hayase, and Katsuro Inoue. 2010. Evolutional analysis of licenses in FOSS. In *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), Antwerp, Belgium, September 20-21, 2010*, Andrea Capiluppi, Anthony Cleve, and Naouel Moha (Eds.). ACM, 83–87. https://doi.org/10.1145/1862372.1862391

[16] Trevor Maryka, Daniel M. Germán, and Germán Poo-Caamaño. 2015. On the Variability of the BSD and MIT Licenses. In *Open Source Systems: Adoption and Impact - 11th IFIP WG 2.13 International Conference, OSS 2015, Florence, Italy, May 16-17, 2015, Proceedings (IFIP Advances in Information and Communication Technology, Vol. 451)*, Ernesto Damiani, Fulvio Frati, Dirk Riehle, and Anthony I. Wasserman (Eds.). Springer, 146–156. https://doi.org/10.1007/978-3-319-17837-0_14

[17] Wes McKinney et al. 2011. pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing* 14, 9 (2011), 1–9.

[18] nexB. 2022. ScanCode. https://www.aboutcode.org/projects/scancode.html Accessed 2022-01-25.

[19] nexB. 2022. ScanCode LicenseDB. https://scancode-licensedb.aboutcode.org/ Accessed 2022-01-26.

[20] Jonathan Oliver, Chun Cheng, and Yanggui Chen. 2013. TLSH: a locality sensitive hash. In *2013 Fourth Cybercrime and Trustworthy Computing Workshop*. IEEE, 7–13.

[21] Philippe Ombredanne. 2020. Free and Open Source Software License Compliance: Tools for Software Composition Analysis. *Computer* 53, 10 (2020), 105–109. https://doi.org/10.1109/MC.2020.3011082

[22] Open Source Initiative. 2022. Machine readable OSI license information. https://github.com/OpenSourceOrg/licenses/ Accessed 2022-01-26.

[23] Simon Phipps and Stefano Zacchiroli. 2020. Continuous Open Source License Compliance. *Computer* 53, 12 (2020), 115–119. https://doi.org/10.1109/MC.2020.3024403

[24] Antoine Pietri, Diomidis Spinellis, and Stefano Zacchiroli. 2019. The Software Heritage graph dataset: public software development under one roof. In *Proceedings of the 16th International Conference on Mining Software Repositories, MSR 2019, 26-27 May 2019, Montreal, Canada.*, Margaret-Anne D. Storey, Bram Adams, and Sonia Haiduc (Eds.). IEEE / ACM, 138–142. https://dl.acm.org/citation.cfm?id=3341907

[25] Lawrence Rosen. 2005. *Open source licensing*. Vol. 692. Prentice Hall.

[26] Guillaume Rousseau, Roberto Di Cosmo, and Stefano Zacchiroli. 2020. Software Provenance Tracking at the Scale of Public Source Code. *Empirical Software Engineering* 25, 4 (2020), 2930–2959. https://doi.org/10.1007/s10664-020-09828-5

[27] Yakov Shafranovich. 2005. RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files. https://datatracker.ietf.org/doc/html/rfc4180 Accessed 2022-01-24.

[28] SPDX Workgroup. 2019. Software Package Data Exchange Licence List. https://spdx.org/license-list https://spdx.org/license-list, retrieved 30 March 2020.

[29] Kate Stewart, Phil Odence, and Esteban Rockett. 2010. Software package data exchange (SPDX) specification. *IFOSS L. Rev.* 2 (2010), 191.

[30] Synopsis. 2020. *2020 Open Source Security and Risk Analysis Report (OSSRA)*. Technical Report. Synopsis. https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/2020-ossra-report.pdf Accessed 2020-04-15.

[31] The Open Group. 2018. file: determine file type. https://pubs.opengroup.org/onlinepubs/9699919799/utilities/file.html Accessed 2022-01-25.

[32] Christopher Vendome, Gabriele Bavota, Massimiliano Di Penta, Mario Linares Vásquez, Daniel M. Germán, and Denys Poshyvanyk. 2017. License usage and changes: a large-scale study on gitHub. *Empir. Softw. Eng.* 22, 3 (2017), 1537–1577. https://doi.org/10.1007/s10664-016-9438-4

[33] Christopher Vendome, Mario Linares Vásquez, Gabriele Bavota, Massimiliano Di Penta, Daniel M. Germán, and Denys Poshyvanyk. 2017. Machine learning-based detection of open source license exceptions. In *Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017*, Sebastián Uchitel, Alessandro Orso, and Martin P. Robillard (Eds.). IEEE / ACM, 118–129. https://doi.org/10.1109/ICSE.2017.19

[34] Stefano Zacchiroli. 2022. *A Large-scale Dataset of (Open Source) License Text Variants*. https://doi.org/10.5281/zenodo.6379164