The General Index of Software Engineering Papers

Zeinab Abou Khalil, Stefano Zacchiroli

The General Index of Software Engineering Papers

Zeinab Abou Khalil
zeinab.abou-khalil@inria.fr
Inria
Paris, France

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

ABSTRACT

We introduce the *General Index of Software Engineering Papers*, a dataset of fulltext-indexed papers from the most prominent scientific venues in the field of Software Engineering. The dataset includes both complete bibliographic information and indexed n-grams (sequence of contiguous words after removal of stopwords and non-words, for a total of 577 276 382 unique n-grams in this release) with length 1 to 5 for 44 581 papers retrieved from 34 venues over the 1971–2020 period.

The dataset serves use cases in the field of meta-research, allowing to introspect the output of software engineering research even when access to papers or scholarly search engines is not possible (e.g., due to contractual reasons). The dataset also contributes to making such analyses reproducible and independently verifiable, as opposed to what happens when they are conducted using 3rd-party and non-open scholarly indexing services.

The dataset is available as a portable Postgres database dump and released as open data.

KEYWORDS

dataset, academic publishing, software engineering, ngrams, meta-analysis, natural language processing, fulltext index

ACM Reference Format:
Zeinab Abou Khalil and Stefano Zacchiroli. 2022. The General Index of Software Engineering Papers. 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/10.1145/3524842.3528494

1 INTRODUCTION

Meta-research [9, 10] is the use of the scientific method to study *science itself*. Meta-research is usually conducted by proxy, studying the artifacts that science produces as byproducts like research papers, datasets, reusable tools, etc. Meta-research studies are common in software engineering too, primarily in the form of *systematic literature reviews* (for example [3, 21]), which are the recommended evidence-based tool of introspection in this field [11, 12], but also via analyses of publication trends either in the field at large or in specific venues [4, 6, 16, 18]. While not yet significantly practiced for this reason in software engineering, scientific paper analyses are gaining traction in other fields, and most notably health sciences, to uncover biases related to research funding [1], gender diversity [2], grant evaluation [17], publishing policies [8], and more.

A significant burden for conducting meta-research on scientific papers is that most scientific literature is not (yet) available as Open Access [20]. Most papers are accessible only via paywalls that not all scholars have gratis access to, depending on the agreements that their institutions have with scientific publishers; and that already excludes all non-affiliated scholars who cannot afford to pay on their own for accessing the papers they desire to study.

Reproducibility is also a concern for meta-analyses. When conducted using primarily 3rd-party and non-open scholarly indexing services (e.g., Google Scholar), both meta-analyses and simple scholarly searches are neither reproducible by peers nor independently verifiable by other means. Search results can change without notice, be biased by service providers, or outright forbidden due to Terms-of-Service contractual provisions.

While these problem cannot be fixed without changing the balance of power in scientific publishing, the availability of *curated, open data paper indexes* can mitigate them. A recent initiative by Malamud, called the *General Index* [5], goes in this direction. Malamud has indexed the full text of 107 million papers from major journal across all scientific fields, extracting from them individual words and n-grams (corresponding to phrases that are short enough for not being considered copyrightable), and released the obtained index as open data.[1] The General Index allows scholars to search papers relevant for their meta-analyses without having to rely on third parties like Google Scholar or publisher websites. From there on, scholars who do have access to paywalls can retrieve them without fees; others can already analyze paper metadata (which are usually available as open data) or decide which paper to buy access for depending on search results. The coverage of software engineering in the General Index is limited though: we have verified (see Section 6 for details) that only major journals are included, leaving out conference proceedings which are very important in the field of software engineering.

Contributions. We introduce the *general index of software engineering papers*, an **open dataset of fulltext-indexed software engineering papers**, covering 34 major journals and conferences in the field (see Table 1), for a total of 44 581 papers published during the last 50 years (1971–2020). Papers metadata have been retrieved from DBLP [14], the reference bibliographic database for computer science, and are integrated into the dataset. Each paper in the dataset has been retrieved in full, converted to text, and text indexed by n-grams with length 1 to 5 (allowing to search papers by individual words up to short phrases), after removing English stop words. A total of 577 276 382 unique n-grams have been indexed and can be looked up in the dataset.

[1] https://archive.org/details/GeneralIndex, accessed 2021-12-15

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

ACM ISBN 978-1-4503-9303-4/22/05…$15.00
https://doi.org/10.1145/3524842.3528494

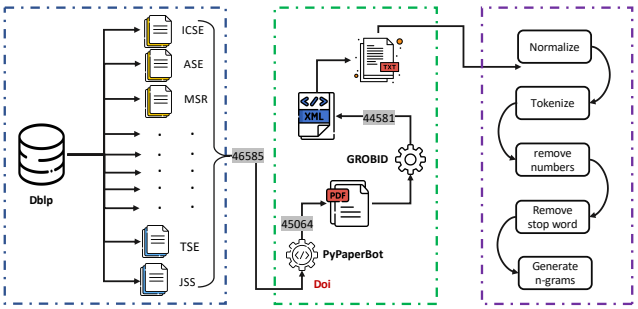

Figure 1: Dataset construction pipeline.

Data availability. The dataset is **released as open data and available from Zenodo** at https://zenodo.org/record/5902231, together with a full replication package to recreate it from scratch. It is provided as a Postgres database dump that expands to a 81 GiB database (including indexes), which is easily exploitable on commodity hardware.

2 METHODOLOGY AND REPRODUCIBILITY

Figure 1 depicts the methodology used to assemble the dataset.

Data sources. For selecting the venues of Table 1 we relied on previous work, adopting the list of Mathew et al. [16]. We then obtained the list of all papers published in those venues using DBLP [14], the reference bibliographic database for computer science publications, making its data available as XML dumps.[2]

Data gathering. For each selected paper, we retrieved complete bibliographic information from the DBLP dataset, which includes DOIs (Digital Object Identifiers).

Starting from DOIs, we then retrieved digital copies, in PDF format, of each paper using the PyPaperBot paper retrieval tool.[3] We excluded from download papers published in 2021, as the year was not yet complete at the time of data gathering. Of the remaining ones, 1521 papers could not be downloaded due to either missing DOIs from DBLP (325 entries) or PyPaperBot failures to resolve DOIs or retrieve the associated papers (1196 entries). This step took a few days for PyPaperBot to download all papers.

Text indexing. We converted PDF versions of the papers to plain text using GROBID [15] which, according to a recent evaluation [22], is the best performing tools for extracting content from scientific papers. Out of the entire corpus, 483 PDFs could not be parsed for various reasons, such as different file format, corrupted PDFs, or PDFs containing raster images instead of textual pages. In the end we obtained a corpus of 44 581 text documents. The PDF conversion took about a day to convert all the papers.

We cleaned up this textual corpus by applying standard NLP (Natural Language Processing) preprocessing steps, as implemented by the Natural Language Toolkit[4] (NLTK): tokenization (splitting text into words, symbols, and punctuation), case normalization to lower case, removal of non-alphabetic tokens, removal of (English) stop words. Note that we applied neither stemming nor lemmatization, as they can introduce clashes in search results for idiomatic expressions in the domain, e.g., "object orient*ed*", "machine learn*ing*", etc. If needed, stemming and lemmatization can be enforced at query time using Postgres text search functions.

Finally, we processed the (cleaned-up) fulltext of each paper to extract n-grams with $n \in \{1, \ldots, 5\}$, using NLTK. Extracted n-grams were ingested into a Postgres database with the schema of Figure 2, together with the corresponding bibliographic information from the DBLP dataset. This step took about 3 days on a standard issue work laptop.

Table 1: Breakdown of fulltext-indexed papers included in the dataset by venue, with coverage period for each.

Acronym	Name	Years	Papers
JSS	Elsevier - Journal of Systems and Software	1979–2020	4810
TSE	IEEE - Transactions on Software Engineering	1975–2020	3588
SPE	Software: Practice and Experience	1971–2020	3301
SW	IEEE Software	1984–2020	3260
ICSE	International Conference on Software Engineering	1988–2020	2947
IST	Information and Software Technology	1992–2020	2920
NOTES	ACM SIGSOFT Software Engineering Notes	1999–2020	2156
ASE	IEEE/ACM International Conference on Automated Software Engineering	1997–2020	2042
FSE	ACM SIGSOFT Symposium on the Foundations of Software Engineering	1993–2020	1795
ICSM	IEEE International Conference on Software Maintenance	1988–2013	1568
RE	IEEE International Requirements Engineering Conference	1993–2020	1321
IJSEKE	International Journal of Software Engineering and Knowledge Engineering	1999–2020	1151
ESE	Springer - Empirical Software Engineering	1996–2020	963
SOSYM	Software and System Modeling	2002–2020	890
MSR	Working Conference on Mining Software Repositories	2005–2020	808
ESEM	International Symposium on Empirical Software Engineering and Measurement	2007–2020	795
WCRE	Working Conference on Reverse Engineering	1993–2013	765
ISSTA	International Symposium on Software Testing and Analysis	1993–2020	746
SQJ	Software Quality Journal	1995–2020	721
MODELS	International Conference On Model Driven Engineering Languages And Systems	2005–2020	677
ICSME	International Conference on Software Maintenance and Evolution	2014–2020	629
ICPC	IEEE International Conference on Program Comprehension	2006–2020	626
FASE	Fundamental Approaches to Software Engineering	1998–2020	619
SMR	Journal of Software: Evolution and Process	2012–2020	576
TOSEM	ACM - Transactions on Software Engineering Methodology	1992–2020	513
ASEJ	Automated Software Engineering	1994–2020	507
REJ	Requirements Engineering Journal	1996–2020	504
SCAM	International Working Conference on Source Code Analysis & Manipulation	2001–2020	467
GPCE	Generative Programming and Component Engineering	2002–2020	392
ISSE	Innovations in Systems and Software Engineering	2005–2020	381
SSBSE	International Symposium on Search Based Software Engineering	2011–2020	243
Total		1971–2020	44 581

[2] https://dblp.uni-trier.de/xml/; specifically, we used the data dump `dblp-2021-04-01`.
[3] https://github.com/ferru97/PyPaperBot. We modified PyPaperBot to use DOIs as PDF filenames. The modified version is included in the replication package.
[4] https://www.nltk.org/, accessed 2022-01-07

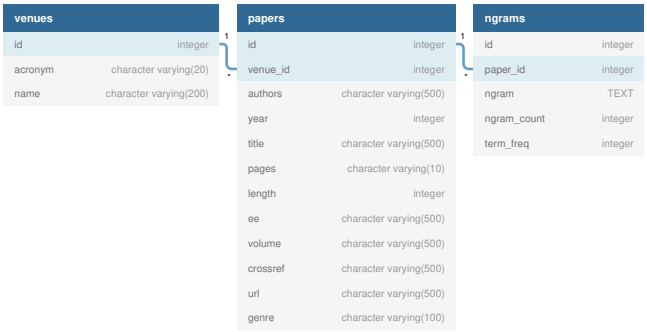

Figure 2: Relational database schema of the dataset.

3 DATA MODEL

The dataset is available as a portable dump of a Postgres [19] database, which can be imported into any instance of such a DBMS. The data model is straightforward for a textual index and is shown in Figure 2 as a relational database schema.

Querying will usually start from the `ngrams` table, which associates n-grams (column `ngram`) to papers (`paper_id`, referencing the `papers` table). Each row also includes the length of the n-gram (`ngram_count`) and its frequency in the paper (`term_freq`) as a measure of relevance of the underlying concept in the study.

The `papers` table contains bibliographic metadata for papers in the dataset. Most columns are self-descriptive, only a few caveats are worth noting: `genre` denotes the paper venue using BiBTeX style names ("inproceedings" v. "article") inherited from DBLP; `ee` is a URL to the electronic version of the paper; `length` is the paper length in pages. Most importantly, note that this table contains information about *all papers from selected venues*, even those with no declared electronic version or with unparsable PDFs; to filter the table on text-indexed papers join with the `ngrams` table (see Section 4 for an example).

The venue table is linked from `papers` via the `venue_id` column and contains acronyms and full names of all venues in the dataset.

4 TUTORIAL

4.1 Data import

The dataset is available as a Postgres database dump which comes as a single compressed file: `swepapers.pgsql.gz`. Provided Postgres is installed and the user has the needed permissions, the dataset can be imported as follows on a UNIX shell:

```
$ createdb swepapers
$ zcat
```

The dataset can then be accessed using `psql swepapers`, or equivalent, and perused via SQL queries.

By default, only indexes on primary key columns will be created. Most dataset use cases will also need an index to efficiently lookup specific n-grams, which can be created with the SQL command **`create index on`** `ngrams (ngram)`. On disk the imported dataset occupies about 58 GiB with default indexes, 81 GiB after adding the n-gram index.

Table 2: Totals and ratios of software engineering papers mentioning "machine learning" (ML) over time (excerpt).

Year	ML papers	All papers	Ratio
2009	93	1636	5%
2010	87	1487	5%
2011	121	1881	6%
2012	157	1996	7%
2013	223	2209	10%
2014	203	2023	10%
2015	246	1963	12%
2016	309	2049	15%
2017	350	2017	17%
2018	459	2236	20%
2019	547	2225	24%
2020	705	2246	31%

4.2 Sample queries

The venue overview of Table 1 was obtained using query:

```
select acronym, name, min(year), max(year),
  count(*) as papers
from venues
join papers on papers.venue_id = venues.id
where exists
  (select 1 from ngrams where paper_id = papers.id)
group by venues.id order by papers desc ;
```

note the filtering based on the `ngrams` table to skip non-text-indexed papers; commenting it will instead give an overview of all papers whose bibliographic information are available in the dataset.

We can extract the total number and percentage of papers mentioning "machine learning" (ML) over time as follows:

```
with all_papers as (
  select year, count(*) as tot from papers group by year ),
sel_papers as (
  select year, count(*) as sel
  from papers join ngrams on ngrams.paper_id = papers.id
  where ngrams.ngram = 'machine learning'
  group by year )
select year, sel, tot, (sel * 100 / tot) as pct
from all_papers natural join sel_papers
order by year ;
```

Table 2 shows the query results for the past decade.

Repeating similar queries with n-grams corresponding to various types of software artifacts, one can study the evolution of the community interest in them over time, as shown in Figure 3 for selected artifacts. A proper study of this topic will need to canvas relevant artifacts from the dataset, determine associated n-grams, deal with their variants, weight term frequency, etc.; this preliminary example is only meant to illustrate the potential of the dataset.

Software development is technology-driven and rapidly evolving; empirical software engineering papers reflect that by looking into the usage of different technologies over time. As an example, one can use the dataset to observe the evolution of research interest in version control system (VCS) technology over time as follows:

```
with all_papers as (
  select year, ngrams.ngram
  from papers join ngrams on ngrams.paper_id = papers.id
```

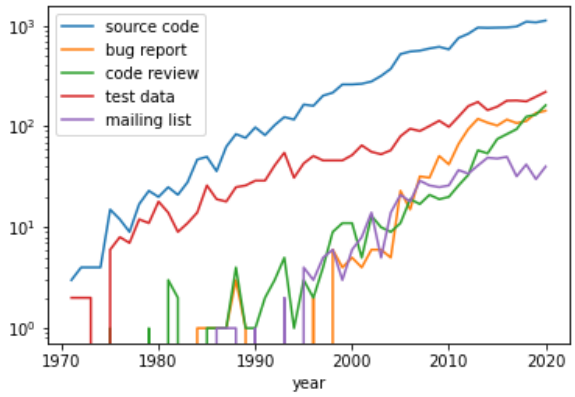

Figure 3: Number of software engineering papers mentioning selected types of software artifacts over time (log scale).

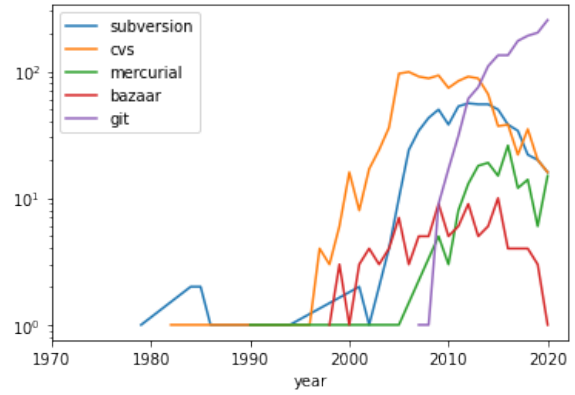

Figure 4: Number of software engineering papers mentioning selected version control systems (VCS) over time (log scale).

```
  where ngrams.ngram in
  ('subversion', 'mercurial', 'cvs', 'git','bazaar'))
select year, ngram, count(ngram) as cnt
from all_papers
group by year, ngram order by year ;
```

Figure 4 shows the query results, highlighting Git as the top mentioned VCS technology since the early 2010s, with every other VCS on the decline.

To conclude, quiz time: what are the most common n-grams across software engineering scientific literature? Queries like the following will satisfy your curiosity (results excerpts are at the end of the paper, so that you can take a guess!):

```
select ngram, sum(term_freq) as freq
from ngrams join papers on ngrams.paper_id = papers.id
group by ngram order by freq desc ;
```

5 LIMITATIONS

Maintenance. To remain valuable in the future, the dataset will need periodic releases and curation. On the one hand, included venues will need to evolve as venues gain prominence, split/merge, etc. Also, papers are published regularly and will need to be indexed and added to the dataset. We plan to maintain the dataset current, but we are also making it easier for *others* to release new versions of it in the future by (1) releasing the dataset creation and maintenance tooling as part of the dataset, and (2) making the dataset update incrementally: there is no need to re-index all previous papers to create a new release, indexing new papers is enough and does not require significant computing resources.

External validity. The dataset is not meant to be representative of *all* scientific venues in software engineering, but only of the curated list of "major" ones detailed in Table 1. This curation criterion is admittedly arbitrary, but is consistent with previous meta-analyses in the field [16, 23]. If other venues gain prominence in the future, they can easily be added to future releases of the dataset. Extending the dataset, both to include new venues and new articles, is trivial; the process is detailed in the dataset documentation at https://zenodo.org/record/5902231.

We rely on DBLP as ground truth for bibliographic information about papers published in the selected venues, so we do not cover (nor could detect the lack of) papers missing from DBLP. Given the preeminence of DBLP in computer science and that of its dataset for meta-studies in the field, we consider this risk to be very low.

Internal validity. As detailed in Section 2, we could not obtain the fulltext of all papers from selected venues, due to either lack of electronic edition/DOI information in DBLP or unparsable PDFs. In the end, 4.3% of the papers in the 1971–2020 period (2004 out of 46 585) could not be text indexed. As the amount is relatively low and most impactful in the 1976–1989 period, we consider it to be an acceptable limitation. Scholars of early software engineering history should, however, consider manually retrieving and text indexing (e.g., using OCR technology) the missing papers.

The choice of stopping at n-grams of length 5 is partly arbitrary and partly dictated by copyright law. N-grams are an excerpt of works that are, for the most part, copyrighted works owned by scientific publishers and released under restrictive licensing terms. While there are no strict thresholds on the length of sentences to be copyrightable, "short" excerpts are generally permitted under fair use doctrine, whereas "long" excerpts will at some point violate article licensing terms. Length 5 for n-grams is a very conservative choice, shared by the General Index [5], which is a much larger dataset in terms of indexed articles, currently considered acceptable in terms of copyright law.

6 RELATED WORK

The General Index (GI) [5] is a fulltext-indexed dataset of papers from scientific journals in all fields of science, and the inspiration for this dataset. GI falls short of good coverage of software engineering (SWE) though. Comparing the list of DOIs in the two datasets shows that 14 668 from ours (36.2%) are missing from GI. Further inspection of the venues containing the word "software" in GI shows that it does not index conference proceedings, explaining the gap.

Vasilescu et al. [23] published a historical dataset of papers accepted at SWE conferences and associated program committee information up to 2012. Kotti et al. [13] investigated the use of

dataset papers published at MSR both using historical trends and interviewing paper authors. Our dataset complements these works by adding the fulltext angle of papers, and extending both the observation period and set of SWE venues.

We are not aware of related work other than the above providing relevant datasets. On the other hand, a significant body of previous work has been conducted using information analogous to those we provide with this dataset; in particular: studies on trends in SWE in general [16, 18], MSR [3, 4], topic modeling [21], or distance learning [6]. In the field of SWE, this kind of studies can be conducted or replicated in the future using this dataset.

Several works [7, 24] in bibliometry and medical sciences have pointed out the shortcomings and risks of relying on Google Scholar as a bibliographic index and search engine. With this dataset we are providing an independent and reproducible alternative for performing fulltext searches across the SWE scientific literature.

7 CONCLUSION

We introduced the General Index of Software Engineering Papers, a dataset of 44 581 papers from top venues in the field, fulltext-indexed using n-grams. The dataset enables reproducible meta-analyses on software engineering literature, without having to depend on third-party and non-open scholarly indexing services.

As future work we plan to explore including in the dataset semantic-oriented representations of the paper corpus such as latent semantic indexes and related vectorial representations.

Quiz answer: the two most common n-grams in software engineering literature are 1-grams "software" and "data"; top 2-grams are "test cases", "source code", and "software engineering".

REFERENCES

[1] Juan P. Alperin, Carol Muñoz Nieves, Lesley A. Schimanski, Gustavo E. Fischman, Meredith T. Niles, and Erin C. McKiernan. 2019. Meta-Research: How significant are the public dimensions of faculty work in review, promotion and tenure documents? *eLife* (Feb 2019). https://doi.org/10.7554/eLife.42254

[2] Jens Peter Andersen, Mathias Wullum Nielsen, Nicole L. Simone, Resa E. Lewiss, and Reshma Jagsi. 2020. Meta-Research: COVID-19 medical papers have fewer women first authors than expected. *eLife* (Jun 2020). https://doi.org/10.7554/eLife.58807

[3] Mário André de Freitas Farias, Renato Lima Novais, Methanias Colaço Júnior, Luis Paulo da Silva Carvalho, Manoel G. Mendonça, and Rodrigo Oliveira Spínola. 2016. A systematic mapping study on mining software repositories. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016*, Sascha Ossowski (Ed.). ACM, 1472–1479. https://doi.org/10.1145/2851613.2851786

[4] Serge Demeyer, Alessandro Murgia, Kevin Wyckmans, and Ahmed Lamkanfi. 2013. Happy birthday! a trend analysis on past MSR papers. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, San Francisco, CA, USA, May 18-19, 2013*, Thomas Zimmermann, Massimiliano Di Penta, and Sunghun Kim (Eds.). IEEE Computer Society, 353–362. https://doi.org/10.1109/MSR.2013.6624049

[5] Holly Else. 2021. Giant, free index to world's research papers released online. Available online at https://www.nature.com/articles/d41586-021-02895-8, accessed 2021-12-15. *Nature* (Oct 2021). https://doi.org/10.1038/d41586-021-02895-8

[6] Fatih Gurcan and Nergiz Ercil Cagiltay. 2020. Research trends on distance learning: a text mining-based literature review from 2008 to 2018. *Interactive Learning Environments* (Sep 2020), 1–22. https://doi.org/10.1080/10494820.2020.1815795

[7] Gali Halevi, Henk F. Moed, and Judit Bar-Ilan. 2017. Suitability of Google Scholar as a source of scientific information and as a source of data for scientific evaluation - Review of the Literature. *J. Informetrics* 11, 3 (2017), 823–834. https://doi.org/10.1016/j.joi.2017.06.005

[8] Chun-Kai (Karl) Huang, Cameron Neylon, Richard Hosking, Lucy Montgomery, Katie S. Wilson, Alkim Ozaygen, and Chloe Brookes-Kenworthy. 2020. Meta-Research: Evaluating the impact of open access policies on research institutions. *eLife* (Sep 2020). https://doi.org/10.7554/eLife.57067

[9] John P. A. Ioannidis. 2010. Meta-research: The art of getting it wrong. *Res. Synth. Methods* 1, 3-4 (Jul 2010), 169–184. https://doi.org/10.1002/jrsm.19

[10] John P. A. Ioannidis, Daniele Fanelli, Debbie Drake Dunne, and Steven N. Goodman. 2015. Meta-research: Evaluation and Improvement of Research Methods and Practices. *PLoS Biol.* 13, 10 (Oct 2015). https://doi.org/10.1371/journal.pbio.1002264

[11] Barbara A. Kitchenham, Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen G. Linkman. 2009. Systematic literature reviews in software engineering - A systematic literature review. *Inf. Softw. Technol.* 51, 1 (2009), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

[12] Barbara A. Kitchenham, Tore Dybå, and Magne Jørgensen. 2004. Evidence-Based Software Engineering. In *26th International Conference on Software Engineering (ICSE 2004), 23-28 May 2004, Edinburgh, United Kingdom*, Anthony Finkelstein, Jacky Estublier, and David S. Rosenblum (Eds.). IEEE Computer Society, 273–281. https://doi.org/10.1109/ICSE.2004.1317449

[13] Zoe Kotti, Konstantinos Kravvaritis, Konstantina Dritsa, and Diomidis Spinellis. 2020. Standing on shoulders or feet? An extended study on the usage of the MSR data papers. *Empir. Softw. Eng.* 25, 5 (2020), 3288–3322. https://doi.org/10.1007/s10664-020-09834-7

[14] Michael Ley. 2002. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In *String Processing and Information Retrieval, 9th International Symposium, SPIRE 2002, Lisbon, Portugal, September 11-13, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2476)*, Alberto H. F. Laender and Arlindo L. Oliveira (Eds.). Springer, 1–10. https://doi.org/10.1007/3-540-45735-6_1

[15] Patrice Lopez. 2008–2021. GROBID - GeneRation Of BIbliographic Data. https://grobid.readthedocs.io/ Accessed 2022-01-25.

[16] George Mathew, Amritanshu Agrawal, and Tim Menzies. 2018. Finding Trends in Software Research. *IEEE Transactions on Software Engineering* (2018). https://doi.org/10.1109/TSE.2018.2870388 To appear.

[17] David G. Pina, Ivan Buljan, Darko Hren, and Ana Marušić. 2021. Meta-Research: A retrospective analysis of the peer review of more than 75,000 Marie Curie proposals between 2007 and 2018. *eLife* (Jan 2021). https://doi.org/10.7554/eLife.59338

[18] Sanam Fayaz Sahito, Abdul Rehman Gilal, Rizwan Ali Abro, Ahmad Waqas, and Khisaluddin Shaikh. 2019. Research Publication Trends in Software Engineering. In *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. 1–4. https://doi.org/10.1109/MACS48846.2019.9024767

[19] Michael Stonebraker and Greg Kemnitz. 1991. The Postgres Next Generation Database Management System. *Commun. ACM* 34, 10 (1991), 78–92. https://doi.org/10.1145/125223.125262

[20] Peter Suber. 2003. Removing the barriers to research: an introduction to open access for librarians. *College & research libraries news* 64 (2003), 92–94. available at https://dash.harvard.edu/bitstream/handle/1/3715477/suber_crln.html.

[21] Xiaobing Sun, Xiangyue Liu, Bin Li, Yucong Duan, Hui Yang, and Jiajun Hu. 2016. Exploring topic models in software engineering data analysis: A survey. In *17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2016, Shanghai, China, May 30 - June 1, 2016*, Yihai Chen (Ed.). IEEE Computer Society, 357–362. https://doi.org/10.1109/SNPD.2016.7515925

[22] Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, and Joeran Beel. 2018. Machine learning vs. rules and out-of-the-box vs. retrained: An evaluation of open-source bibliographic reference and citation parsers. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*. 99–108.

[23] Bogdan Vasilescu, Alexander Serebrenik, and Tom Mens. 2013. A historical dataset of software engineering conferences. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, San Francisco, CA, USA, May 18-19, 2013*, Thomas Zimmermann, Massimiliano Di Penta, and Sunghun Kim (Eds.). IEEE Computer Society, 373–376. https://doi.org/10.1109/MSR.2013.6624051

[24] Rita Vine. 2006. Google Scholar. *J. Med. Libr. Assoc.* 94, 1 (Jan 2006), 97. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1324783