Harald Lüngen, Marc Kupietz, Piotr Bański,
Adrien Barbaresi, Simon Clematide, Ines Pisetta (eds.)

Proceedings of the Workshop on *Challenges in the Management of Large Corpora (CMLC-9) 2021*

Limerick, 12 July 2021
Online-Event

IDS | LEIBNIZ-INSTITUT FÜR DEUTSCHE SPRACHE

Leibniz-Institut für Deutsche Sprache · R 5, 6-13 · 68161 Mannheim
www.ids-mannheim.de

URN: urn:nbn:de:bsz:mh39-104676
DOI: https://doi.org/10.14618/ids-pub-10467

Challenges in the Management of Large Corpora 2021

Workshop Programme 12 July 2021

Session 1 (10.00 – 11.30)

Presentations

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary and Benoît Sagot:
Ungoliant: An Optimized Pipeline for the Generation of a Very Large-Scale Multilingual Web Corpus

Markus Gärtner, Felicitas Kleinkopf, Melanie Andresen and Sibylle Hermann:
Corpus Reusability and Copyright – Challenges and Opportunities

Nils Diewald, Eliza Margaretha and Marc Kupietz:
Lessons learned in Quality Management for Online Research Software Tools in Linguistics

Session 2 (11.45 – 12.30)

Panel Discussion on Research Software Management

Laurence Anthony (Waseda University, Japan)
Nils Diewald (IDS Mannheim, Germany)
Stefan Evert (Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany)
Andrew Hardie (Lancaster University, UK)
Miloš Jakubíček (Lexical Computing Ltd., UK)
Pavel Vondřička (Charles University, Prague, Czech Republic)

CMLC-9 Organising Committee

Piotr Bański, Marc Kupietz, Harald Lüngen	Leibniz-Institute for the German Language, Mannheim
Adrien Barbaresi	Berlin-Brandenburg Academy of Sciences
Simon Clematide	University of Zurich

CMLC-9 Programme Committee

Laurence Anthony	Waseda University, Japan
Vladimír Benko	Slovak Academy of Sciences, Slovakia
Felix Bildhauer	IDS Mannheim, Germany
Nils Diewald	IDS Mannheim, Germany
Tomaž Erjavec	Jožef Stefan Institute, Ljubljana, Slovenia
Stefan Evert	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Johannes Graën	University of Zurich, Switzerland
Andrew Hardie	Lancaster University, UK
Serge Heiden	ENS de Lyon/IHRIM, France
Miloš Jakubíček	Lexical Computing Ltd., UK
Paweł Kamocki	IDS Mannheim, Germany
Natalia Kotsyba	Samsung Poland
Dawn Knight	Cardiff University, UK
Michal Křen	Charles University, Prague, Czech Republic
Sandra Kübler	Indiana University, USA
Veronika Laippala	University of Turku, Finland
Jochen Leidner	Thomson Reuters, UK
Verena Lyding	EURAC Research, Italy
Paul Rayson	Lancaster University, UK
Laurent Romary	INRIA, France
Jan-Oliver Rüdiger	IDS Mannheim, Germany
Kevin Scannell	Saint-Louis University, USA
Roland Schäfer	FU Berlin, Germany
Roman Schneider	IDS Mannheim, Germany
Serge Sharoff	University of Leeds, UK
Irena Spasić	Cardiff University, UK
Ludovic Tanguy	University of Toulouse, France

CMLC-9 Homepage: http://corpora.ids-mannheim.de/cmlc-2021.html

Table of contents

Author Index

Preface

The ninth CMLC meeting continues the successful series of “Challenges in the management of large corpora” events, previously hosted at LREC (since 2012) and CL (since 2015) conferences. As in the previous meetings, we wish to explore common areas of interest across a range of issues in linguistic research data and tool management, corpus linguistics, natural language processing, and data science, with a special focus on tools, this time.

This year’s (online) event comprises a *presentation session* with three paper presentations and a *panel discussion* on research software management with six international panellists who are each responsible for the development of professional corpus research systems, from both the academic and the commercial sphere.

We invite the readers to peruse the submissions collected in the present volume and to consider joining the CMLC community at our future meetings.

The CMLC-9 Organising Committee

July 2021

Ungoliant: An Optimized Pipeline for the Generation of a Very Large-Scale Multilingual Web Corpus

Julien Abadji[1] **Pedro Javier Ortiz Suárez**[1,2] **Laurent Romary**[1] **Benoît Sagot**[1]
[1]Inria, Paris, France
[2]Sorbonne Université, Paris, France
{julien.abadji, pedro.ortiz
benoit.sagot, laurent.romary}@inria.fr

Abstract

Since the introduction of large language models in Natural Language Processing, large raw corpora have played a crucial role in Computational Linguistics. However, most of these large raw corpora are either available only for English or not available to the general public due to copyright issues. Nevertheless, there are some examples of freely available multilingual corpora for training Deep Learning NLP models, such as the OSCAR and Paracrawl corpora. However, they have quality issues, especially for low-resource languages. Moreover, recreating or updating these corpora is very complex. In this work, we try to reproduce and improve the goclassy pipeline used to create the OSCAR corpus. We propose a new pipeline that is faster, modular, parameterizable, and well documented. We use it to create a corpus similar to OSCAR but larger and based on recent data.Also, unlike OSCAR, the metadata information is at the document level. We release our pipeline under an open source license and publish the corpus under a research-only license.

1 Introduction

With the increasing interest in language modeling in recent years in Natural Language Processing (NLP) (Rogers et al., 2020), particularly concerning contextualized word representations[1] (Peters et al., 2018; Devlin et al., 2019), there has also been an explosion in interest for large raw corpora, as some of these latest models require almost 1TiB of raw text for pre-training (Raffel et al., 2020; Brown et al., 2020).

While most of these language models were initially trained in English (Devlin et al., 2019; Yang et al., 2019; Clark et al., 2020; Zaheer et al., 2020; Xiong et al., 2021) and consequently most of the large corpora used to pre-train them were in English, there has been a recent push to produce larger high quality corpora for other languages, namely those of Grave et al. (2018), CCNet (Wenzek et al., 2020), Multilingual C4 (mC4) (Xue et al., 2020) and OSCAR (Ortiz Suárez et al., 2019, 2020) for pre-training language models, as well as, Paracrawl (Esplà et al., 2019; Bañón et al., 2020), CCAligned (El-Kishky et al., 2020) and WikiMatrix (Schwenk et al., 2021) which are parallel corpora for training Machine Translation (MT) models. Of these, only OSCAR, Paracrawl, CCaligned and WikiMatrix are freely available and easily downloadable.

In this paper we propose a new multilingual corpus for language modeling, and for that we take inspiration in the OSCAR corpus and its pipeline goclassy[2] (Ortiz Suárez et al., 2019, 2020), but we propose a new pipeline *Ungoliant*[3] that is faster, modular, parametrizable and well-documented. We then use it to produce a new corpus similar to OSCAR, yet larger, based on recent data containing mentions of last years' events such as the COVID-19 pandemic, the 2020–2021 United States racial unrest, the Australian wildfires, the Beirut explosion and Brexit among others. Moreover, contrarily to OSCAR, our corpus retains metadata information at the document level. We release our pipeline under an Apache 2.0 open source license and we publish the corpus under a research-only use license following the licensing schemes proposed by OSCAR (Ortiz Suárez et al., 2019, 2020) and Paracrawl (Esplà et al., 2019; Bañón et al., 2020).

[1]In which one takes a unnanotated large textual corpus in a particular language and tries to predict a missing word in order to learn a vector space representation for it.

[2]https://github.com/oscar-corpus/goclassy

[3]https://github.com/oscar-corpus/ungoliant

2 Limitations of the OSCAR Corpus and its Generation Pipeline

2.1 OSCAR

OSCAR is a multilingual corpus derived from CommonCrawl[4], a project that provides web crawl data for everyone on a periodic manner, usually each month. CommonCrawl provides data in several formats, from raw HTML source code to pure text. OSCAR was generated from the pure text data version (WET files) of the November 2018 crawl, distributed in the form of 56,000 *shards*, that were then filtered and classified by language (Ortiz Suárez et al., 2019, 2020). OSCAR is available through several means, and has been used in numerous projects (Ortiz Suárez et al., 2019). OSCAR's generation pipeline also suffers from numerous issues, which we plan to address simultaneously with the release of a new, more powerful, stable, and higher quality pipeline

Simply put, OSCAR is composed of single language files that contain textual data (`ta.txt` for the Tamil language, for example). However, due to the often huge sizes of these files, and subsequently the impracticality of storage and distribution, OSCAR files are split and compressed in equally sized parts.

OSCAR comes in four different versions, each suited differently for different tasks, and allows less limited ways of sharing the corpus more widely. These versions are either *unshuffled* or *shuffled* (that is, for each language, lines have been shuffled, destroying records integrity), and *non-deduplicated* or *deduplicated* (since duplicate lines account for more than half of the total data[5] generated by the pipeline). For the unshuffled versions, each language file contains paragraphs that come from the same record, and each paragraph is separated by a newline.

OSCAR is inherently linked to its generation pipeline, and as such its quality partly depends on the pipeline's quality. While OSCAR is considered to be one of the cleanest multilingual corpora available (Caswell et al., 2020, 2021), several problems have been described, and the state of the publicly available code raises questions about maintenance and maintenability of the pipeline itself.

Apart from the fact that its content dates back to 2018, the current OSCAR corpus suffers from quality issues discussed in (Caswell et al., 2020, 2021), including:

- **Language label mismatches and inconsistencies**, which occurs earlier in the pipeline and would be fixable downstream,
- **Representation washing** as defined by Caswell et al. (2021), whereby low resource languages, while present in the corpus, are of a significantly lower quality than higher resource languages without any quality metric available publicly.

The most recent Common Crawl dump contains 64,000 shards. Each shard is composed of numerous records, and each record holds textual content along with metadata. While CommonCrawl shards hold document-level metadata that could be useful downstream, they were discarded and do not appear in OSCAR, whereas other corpora generated from the same source include them, e.g. CCNet (Wenzek et al., 2020). This limits OSCAR users to the textual content only, whereas metadata could have been distributed along with the corpus itself.

2.2 goclassy

OSCAR was built using *goclassy*, a high-performance asynchronous pipeline written in Go (Ortiz Suárez et al., 2019). However, it suffers from several caveats that makes the re-generation and update of the corpus relatively complex in practice.

While goclassy's source code is easily readable thanks to the choice of an uncluttered language and a pragmatic approach, the lack of structure in both the source and the project itself makes goclassy difficult to extend and maintain.

The pipeline is not functional out-of-the-box, as the user has to provide the compressed shards from CommonCrawl, manually install fasttext (Joulin et al., 2016, 2017) and create specific directories by themselves, since only partial instructions are given in the supplied README file.

goclassy also makes heavy use of I/O, as data is saved and loaded repeatedly between steps; as an example, the identification step stores language identification data and individual sentences in two files, before generating the final files (one per language). Despite these limitations, goclassy's performance is good due to Go's emphasis on easy and efficient parallelization and inherent speed. The pipeline uses clever handling of file descriptors, limiting I/O calls cost in some parts.

[4] https://commoncrawl.org

[5] OSCAR-orig: 6.3TB, OSCAR-dedup: 3.2TB

3 Building a new OSCAR-like corpus

We introduce *Ungoliant*, a new corpus generation pipeline that, like goclassy, creates a large-scale multilingual text corpus from a CommonCrawl dump. Contrarily to goclassy, Ungoliant is fully modular, better structured, and highly parametrizable; thereby allowing comparisons between several parallelization strategies. A specific effort was put in testing and documentation. Parts of Ungoliant are heavily inspired by goclassy, although it is implemented in Rust rather than in Go, which is sometimes faster.[6]

Additionally, we use Ungoliant to generate a new corpus from a recent Common Crawl dump. The new corpus includes metadata information while retaining backward compatibility with the OSCAR corpus.

3.1 Ungoliant

3.1.1 Rationale and scope

While Ungoliant is heavily inspired by goclassy, it provides a better set of tools to download, process, filter and aggregate textual and contextual data from CommonCrawl. These operations can be sequential, parallel or both, depending on contexts and performance requirements.

We provide both batch and streaming processing, so that the whole pipeline could be run either online, with every step running on streams of data, or offline, with every step running on tangible files, or a mix of both, using already downloaded CommonCrawl dumps but streaming the rest of the process. Moreover, we embed numerous filtering and deduplication utilities directly inside Ungoliant, making these features available for pipeline composition and post-processing.

Ungoliant features a loosely defined pipeline interface, on which we re-implement goclassy's one, while improving performance by threading more aggressively and avoiding I/O where it is not necessary: While goclassy uses intermediate files for tags and sentences, we try to keep everything in memory in order to avoid losing time loading or writing files. The Rust language provides constructs that helps us build complex abstractions and pipelines while limiting proactive file I/O or computing, since nearly all the reimplemented pipeline is built around lazy evaluation. File I/O is only used when loading shards, and when writing sentences in language files.

Platform	#shards	goclassy	Ungoliant	Approx. speedup
Desktop	1	30s	13s	×2.3
	10	3m6s	2m12s	×1.3
	25	9m10s	5m47s	×1.5
HPC	1	40s	6s	×6.6
	25	2m40s	1m6s	×2.4
	100	7m59s	4m14s	×1.8

Table 1: Comparison of approximate generation times depending on platform and number of shards.

Through benchmarking we found that the best parallelization strategy is to use rayon[7], a work-stealing (Blumofe and Leiserson, 1999) parallel and concurrent library enabling massive parallelization. We parallelize on shard-, record- and sentence-level processing.

To evaluate Ungoliant performance, we run both goclassy and Ungoliant's implementation on 1, 10, 25 and 100 Common Crawl shards both on a middle-range laptop computer (i5-7200u, 8GB RAM, NVMe SSD) and a HPC node (Xeon 5218 (64 Threads), 180GB RAM). Results are shown in Table 1.

Ungoliant performs better than goclassy on all tasks, independently of the platform or number of shards processed. However, we can note that Ungoliant's speedup is higher on short tasks, which is explained by its aggressive multithreading strategy, while goclassy uses a record-scope multithreading at its finest granularity.

3.2 Iterating on the goclassy pipeline

CommonCrawl dumps contain metadata that hold useful information such as related records, recognized language(s), or origin URLs. Since OSCAR pipeline discards metadata and sentences can be shuffled, we lose the ability to investigate those metadata themselves, as well as working on potentially multilingual documents, since we separate text from metadata.

The new pipeline (and the resulting new corpus schema) aims to establish a first link between textual data and metadata from CommonCrawl, while staying backward compatible with the existing OSCAR schema.

In other words, switching from the original OSCAR corpus and the newly generated one should be a drop-in operation.

[6] https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/rust-go.html

[7] https://github.com/rayon-rs/rayon

3.2.1 Metadata extraction and linking

Our choice of keeping the corpus backward compatible with the original OSCAR introduces changes in the way the corpus is generated, namely regarding metadata: a record's body is composed of sentences that aren't guaranteed to be of the same language. Since OSCAR merges sentences from multiple records into a single file, special attention has to be paid to the metadata dispatch too.

Approaches to tackle this problem range from (1) storing all metadata in a single location to (2) having language-specific metadata files that contain the metadata for each line in the language file.

Both (1) and (2) have their strengths and weaknesses, namely:

1. Having all metadata at the same place may facilitate wide queries about whole metadata, but at a cost of a very large size (which harms both accessibility and performance).

2. Getting the metadata for a given line is fast since line numbers are synchronized, but there is repeated information and a potentially important increase in size.

We choose a hybrid approach which keeps metadata local to each language, while trying to limit the information repetition by keeping an entry by group of chunks rather than by line, where a chunk is a series of contiguous sentences that share the same language from the same document.

An overview of the pipeline can be seen in Figure 1, with a more precise view on record processing and metadata extraction in Figure 2.

Metadata are distributed via JSON-encoded files holding an ordered list of metadata entries, along with offsets (o) and paragraph lengths (l), enabling any user to get the content of a said metadata by querying for lines $(o, o + l]$ in the content file.

This approach still has drawbacks, in particular when looking for the corresponding metadata of a given sentence/paragraph, where one has to perform a search on the metadata file, or when working with multilingual documents. Another drawback is the resulting cost of potentially merging back numerous language parts: Since metadata query is offset-based, merging back metadata files implies updating those offsets.

Having paragraphs and metadata linked by offsets in a highly parallelized pipeline implies to take special care at the offset level. The solution is to use shard-scoped offsets (starting from 0 for each language), and to keep global offsets protected by a mutex guard. This way, when a given shard is done processing and is ready to be written on disk, we convert shard-scoped offsets to global-scoped ones, update the global-scoped ones and then write text and metadata on disk.

Platform	#shards	OSCAR	With Metadata	Speedup
Desktop	1	13s	12s	×1.1
	10	2m12s	1m55s	×1.1
	25	5m47s	4m50s	×1.2
HPC	1	6s	7s	×0.9
	25	1m6s	1m12s	×0.9
	100	4m14s	4m36s	×0.9

Table 2: Comparison of approximate generation times with and without metadata generation.

Version	Source	Textual (dedup)	Metadata	Total (increase)
2018	7.42TB	6.3TB (3.2TB)	N/A	6.3TB
2021	8.06TB	7.2TB (3.3TB)	1.2TB	8.4TB (+33%)

Table 3: Comparison of CommonCrawl and OSCAR sizes between 2018 and 2021 versions. Compressed (CommonCrawl) sources are from November 2018 and February 2021. Total is Textual + Metadata without deduplication.

We compare running times for the reimplementation of the goclassy pipeline, and our new pipeline adding metadata extraction, using both desktop and HPC contexts. The results are reported in Table 2.

Metadata generation does not seem to influence generation time dramatically. However, we can notice a slight performance difference between HPC and Desktop contexts. These differences may lie in the storage medium differences, I/O layout, or algorithmic peculiarities benefiting desktop contexts because of other bottlenecks.

3.3 Characteristics of our new backward compatible OSCAR-like corpus

We evaluate the newly generated corpus, assessing its ability to reflect events that occurred after the publication of OSCAR 2018 and detail the metadata format and potential use.

3.3.1 Comparison with OSCAR

While it is expected that our new corpus has a larger file size than OSCAR since CommonCrawl itself grew from 7.42TB to 8.06TB, metadata quickly adds up and take for nearly 15% of the whole uncompressed data.

The size augmentation is not the same for each language, and while the whole corpus is bigger

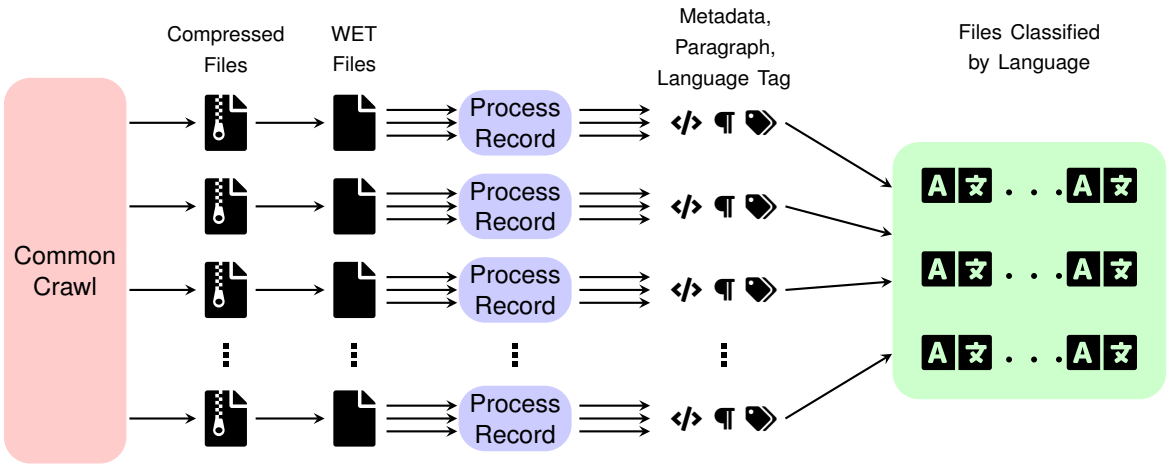

Figure 1: Scheme of the Ungoliant pipeline. The red square represents CommonCrawl content hosting, where the compressed shards are fetched. The *Process Shard* steps hold shard processing, paragraph creation and merging (see Figure 2), and are internally parallelized.

- : CommonCrawl compressed shard.
- : Uncompressed shard, containing records.
- </>: Record Metadata
- : Language identification
- ¶: Paragraph, composed of sentences identified as

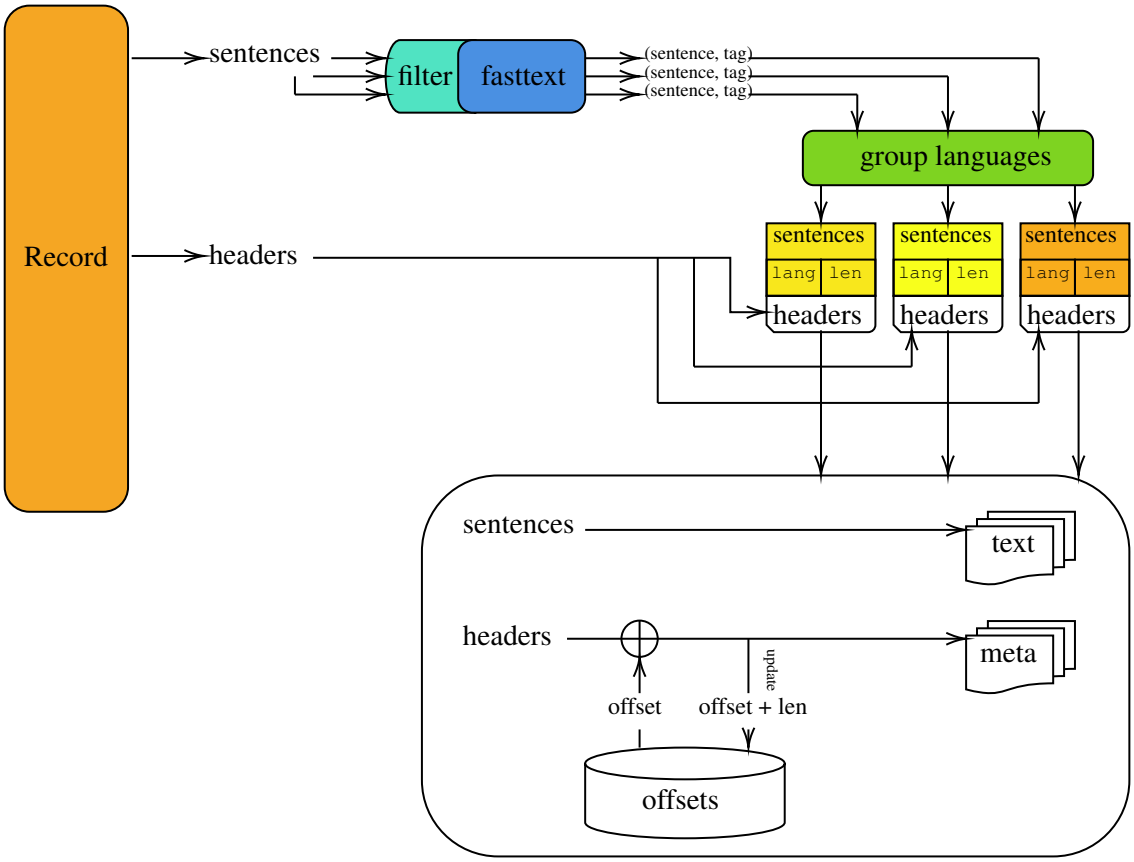

Figure 2: Record processing with metadata extraction. Headers are kept aside while sentences are identified and grouped into same-language bins. Headers are then cloned for each bin, and are sequentially stamped with an offset that is recorded for the whole operation, and written to disk into text and metadata files by language.

now, some languages are smaller than they were before.

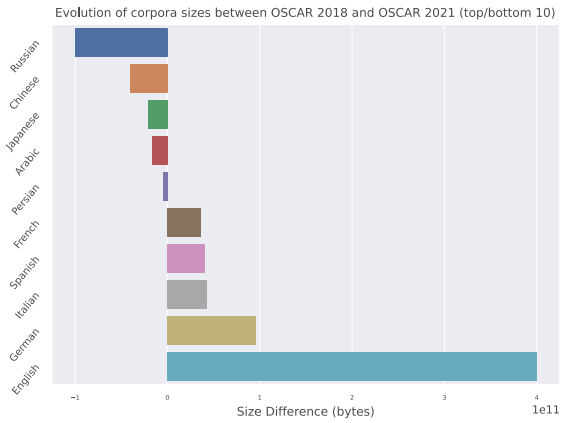

Figure 3: Comparison of language size (in bytes) between OSCAR 2018 and OSCAR 2021 (top/bottom 5 only).

Results show that already largely represented languages gain more and more data (like the English language, which constitutes more than a third of the original OSCAR), except for the Russian language which loses approximately 100Gb of textual content. These results are summarized in Figure 3.

However, in a context where the number of languages is very high (higher than 150) and of varying sizes, evolution can't be analyzed via a mere size evaluation. By computing, for each language, the relative size difference between the 2018 and 2021 releases of OSCAR, less resourced languages do appear, hinting at a better representation of some of them. These results can be found in Figure 4.

Numerous languages have been omitted from Figure 4, either:

- because they were present in the original OSCAR and are now absent (*Central Bikol* and *Cantonese*)
- because they were absent in the original OSCAR and are now present (*Manx*, *Rusyn*, *Scots* and *West Flemish*)

Precautions have to be taken when using these corpora and further work has to be done to correctly assess the quality of low-to-mid resource languages in order to better reflect the quality of each corpus to the OSCAR users. Some languages exhibited either a particularly low number of sentences or a very low quality, and as such couldn't be usable, while still accounting for a language in the total language count of the original OSCAR.

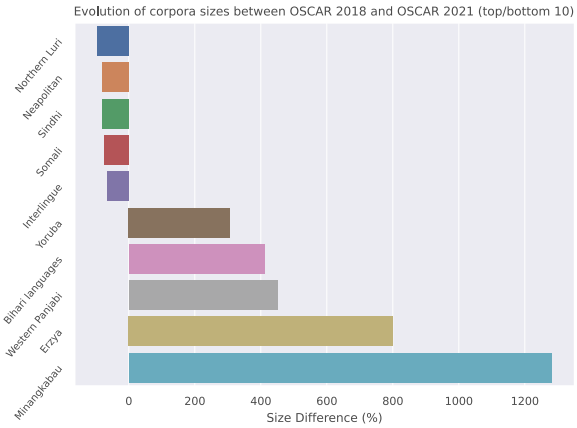

Figure 4: Comparison of language percentage between OSCAR 2018 and OSCAR 2021 (top/bottom 5 only).

3.3.2 Metadata

Metadata provides new contextual data that is useful to evaluate the corpus and draw metrics.

The total size of metadata is 1.2TB, ranging from 4Kb to 500Gb, depending on the number of lines. Relative size varies from 100% to 20%, diminishing with the textual data size, which is expected.

Metadata are provided in single files for now, but split versions of both textual and contextual data will be released soon after the release of the corpus, enabling easy access.

Our choice of keeping metadata aside from the main content adds some complexity when working with both textual and contextual data:

- When trying to get the metadata of given sentence, one has to get the line number k, then sequentially (or use a search algorithm since offsets are sorted) look for the record (with offset o and length l), where $k \in [o, o + l]$.
- Looking for lines corresponding to a particular metadata entry is easier: one has to read the textual file, skipping until the o-th line, then read l lines.

3.3.3 Presence of events

Using a sample of an English part of our corpus, we perform a simple search of terms in order to assess and compare the presence of pre- and post-2018 events and persons in both corpora. Terms and frequency are grouped in Table 4.

Our corpus keeps around the same number of occurrences for pre-2018 events or public figures such as Barack Obama, while increasing the occurrence of people linked to more recent events (Joe Biden).

Language	Term	2018	2021
Arabic	Beirut port explosion	0	31
Burmese*	Min Aung Hlaing	387	3439
English	Obama	30039	27639
English	Biden	990	19299
French	Yellow Vests	2	96

Table 4: Comparison of occurrences of news-related terms between OSCAR and our corpus in a sample of 100 CommonCrawl shards. For the Burmese language, we use the whole 2018 and 2021 corpus since it is a low resource language. Terms are translated in the corpus language.

We include search terms linked to post-2018 events in French and Arabic which are smaller corpora (resp. 200 and 80 GB), and in Burmese, a mid-resource language (approximately 2GB). We observe a term occurrences evolution that reflects the linked events' timing and importance.

3.4 License

This new corpus will be released under a research-only license that is compliant with the EU's exceptions for research in text and data mining. Contrarily to the original OSCAR, no shuffled version of the corpus will be distributed, instead we will put in place an authentication system that will allow us to verify that requests for the corpus come from research institutions. A contact form will be also provided for independent researchers so that we can study their particular cases and determine if the utilization of the corpus corresponds to a legitimate research use.

Moreover, the introduction of metadata makes our corpus far more queryable, thus simplifying and speeding up the handling of take-down GDPR requests. For this reason, we will be releasing the complete set of metadata under a CC0 public domain license, so that any individual can check if their personal or even copyrighted data is in our new corpus and make a request accordingly.

4 Conclusion

We show that our solution is able to generate an OSCAR-like corpus that is augmented with metadata without breaking compatibility, while being faster, better tested and thoroughly documented. We believe our new pipeline and corpus will be useful for applications in computational linguistics as well as in corpus linguistics in general.

The generated corpus is of a larger size when including metadata and without deduplication. However, deduplicated textual content is of the same magnitude between OSCAR 2018 and OSCAR 2021, while reflecting topic changes from all over the world. This fact suggests that old data may be lost with the time passing, and could be resolved by using CommonCrawl releases to build an incremental corpus, with every version augmenting the corpus size.

Metadata enables queries and statistics on the generated data, and we believe that it can be used to filter OSCAR to generate corpora that respond to certain criteria.

We plan to make this new version of OSCAR available under research constraints, with split versions of both textual content and metadata along with tools to operate on the corpus, enabling fast and easy operation on the corpus for researchers.

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Robert D. Blumofe and Charles E. Leiserson. 1999. Scheduling multithreaded computations by work stealing. *J. ACM*, 46(5):720–748.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona,

Spain (Online). International Committee on Computational Linguistics.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *arXiv:2103.12028 [cs]*. Presented at the AfricaNLP 2021 workshop.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In *Proceedings of the 7th Workshop on Challenges in the Management of Large Corpora*, Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Corpus Reusability and Copyright – Challenges and Opportunities

Markus Gärtner[1] **Felicitas Kleinkopf**[2] **Melanie Andresen**[1] **Sibylle Hermann**[3]
[1]Institute for Natural Language Processing, University of Stuttgart
[2]Center for Applied Legal Studies (ZAR), Karlsruhe Institute of Technology
[3]University Library, University of Stuttgart
markus.gaertner@ims.uni-stuttgart.de, felicitas.kleinkopf@kit.edu,
melanie.andresen@ims.uni-stuttgart.de, sibylle.hermann@ub.uni-stuttgart.de

Abstract

Making research data publicly available for evaluation or reuse is a fundamental part of good scientific practice. However, regulations such as copyright law can prevent this practice and thereby hamper scientific progress. In Germany, text-based research disciplines have for a long time been mostly unable to publish corpora made from material outside of the public domain, effectively excluding contemporary works. While there are approaches to obfuscate text material in a way that it is no longer covered by the original copyright, many use cases still require the raw textual context for evaluation or follow-up research. Recent changes in copyright now permit text and data mining on copyrighted works. However, questions regarding reusability and sharing of such corpora at a later time are still not answered to a satisfying degree. We propose a workflow that allows interested third parties to access customized excerpts of protected corpora in accordance with current German copyright law and the soon to be implemented guidelines of the Digital Single Market directive. Our prototype is a very lightweight web interface that builds on commonly used repository software and web standards.

1 Introduction

In several fields of text-based research with corpora such as corpus linguistics, digital humanities, and computational literary studies, researchers have for a long time been faced with the precarious situation that text corpora cannot be published for reuse due to legal issues. While the Fair Use doctrine of United States law and similar legal systems expresses a rather usage-friendly idea for copyrighted material, other legislatures take approaches that focus primarily on the rightholders instead. One of these is the German copyright law[1] (as substantially determined by EU law) as the legal context in which this work is situated.

Using copyrighted material as the basis of public text corpora was generally not possible under German copyright until recently. Naturally the option of making special arrangements with individual rightholders always existed and has been used by larger projects and institutions, often focusing on data from the news domain. But given the time investment needed to reach such agreements and the relatively short lifespan of most (smaller) research projects, those cases remain exceptions. As a direct result, large portions of the corpora created from German texts outside the news domain are based on material that has already been in the public domain.[2]

With recent changes in German and European copyright, protected material is now available for non-commercial research (see Section 2). However, the question about archiving and public availability of research data and corpora created from copyrighted material after the official end of associated projects is still not solved to a satisfying degree.

To address this issue, we present an architecture concept and its prototypical implementation that allows researchers to make excerpts of otherwise non-publishable copyrighted text corpora available for (scientific) reuse. For this approach, the intelligent choice of excerpts tailored to the user's needs is key, because having only a randomly or statically selected part of a corpus available is of limited benefit for some research questions. Therefore, the system additionally integrates a dedicated query component. In order to maximize the utility, users can express their interest based on available annotations in the corpus and as such receive excerpts of higher relevance for them.

[1]For the remainder of this text "copyright" or "local copyright" refers to German copyright law, unless explicitly stated otherwise.

[2]Usually due to the original author being dead for a sufficiently long period of time.

The approach is tailored to the current legal situation in Germany, but can easily be transferred to other legal frameworks that contain regulations of similar setup. With the upcoming implementation of the DSM-Directive (see Section 2) into national laws, the copyright situation for text and data mining (TDM) within the EU becomes more homogeneous. As such the concept in this paper can serve as a blueprint for corpus reusability in this shared legal sphere.

We discuss the relevant legal framework in Section 2 and contextualize our work in Section 3. Sections 4 and 5 describe the XSample approach and the current prototype implementation and finally Section 6 concludes.

2 Legal Framework

In order to discuss why copyright problems arise in relation to the reuse of TDM corpora, the legal framework of TDM is first presented below in Section 2.1. Of enormous importance in this respect is recent European law, the Directive on Copyright in the Digital Single Market (Section 2.2), which, although not directly applicable to national law, had to be implemented by the member states by June 7th, 2021. Finally, we discuss why the reusability of corresponding corpora remains legally unclear and what approach should be considered to address this problem under European (Section 2.3) and German law (Section 2.4) .

2.1 Text and Data Mining and Copyright Law

Copyright law must only be observed when practicing text and data mining if text and data are protected under copyright or related rights. The preconditions on a protection depend partially on national law and partially on European law. According to the case law of the European Court of Justice (ECJ), a work protected by copyright exists if it is the author's own intellectual creation.[3] However, the necessary level of creation is low: It can already be reached by a part of a work that consists of eleven words.[4] In research areas that deal with text-based resources, a protection by copyright must be assumed in most cases. Moreover, databases are protected under a so-called sui generis right in case of being a qualitatively and/or quantitatively substantial investment in either the obtaining, verification or presentation of the contents, Article 7 (1) of the directive 96/9/EC on the legal protection of databases (Database-Directive).

In fact, the analyses performed in text and data mining processes as such do not violate intellectual property. However, in terms of preparing research data it is necessary to copy and to make works and related rights available to the public within research groups, see also Raue (2018, p. 381) and Geiger et al. (2018, p. 817 f.).[5]

By Articles 2 and 3 of the Directive 2001/29/EC on the harmonization of certain aspects of copyright and related rights in the information society (InfoSoc-Directive), these acts of exploitation are exclusively entitled to the holders of the copyrights and related rights as the reproduction right and the right of communication to the public. Therefore, these acts require the rightholder's permission or an exception or limitation provided for by law.[6]

2.2 A Developing Legal Framework within the European Union

Although research on copyrighted works is still a rarity in the digital humanities, it has already been allowed in several member states of the European Union for a few years: The first member state to implement a national regulation that allows text and data mining research has been the United Kingdom in 2014[7], followed by France in 2016[8], Estonia in 2017[9] and Germany in 2018[10] (Geiger et al., 2018, p. 830 f.). In introducing those limitations or exceptions to copyright and related rights, these states referred to Article 5 (3a) InfoSoc-Directive, an authorization to grant additional rights for non-commercial scientific research.

By Articles 3 and 4 of the new Directive 2019/790 on copyright and related rights in the Dig-

[3] ECJ – Infopaq, ECLI:EU:C:2009:465, no. 30 ff.

[4] ECJ – Infopaq, ECLI:EU:C:2009:465, no. 38.

[5] DSM-Directive, recital 9. However, after the case law of the ECJ, the requirement of being publicly available "refers to an indeterminate number of potential listeners, and, in addition, implies a fairly large number of persons", e.g. ECJ, ECLI:EU:C:2012:140, Società Consortile Fonografici (SCF)/Marco Del Corso, no. 84. Therefore, only large research groups can be considered public.

[6] Contrary to the literal meaning, an exception or a limitation does not limit usage but enables it by permitting particular usages, for instance reproducing a copyrighted work or making it publicly available.

[7] Regulation 3 of the British Copyright and Rights in Performances Regulations 2014, No 1372, Article 29a

[8] Article 38 of the French law no. 2016-1321, 7th October 2016

[9] Estonian Copyright Act 19 (3)

[10] BT-Drs. 18/12329 (printed matters of the German parliament), Regulation 17, § 60d

ital Single Market (DSM-Directive), all EU member states are now obliged to provide for mandatory exceptions or limitations for text and data mining for the benefit of non-commercial scientific research and also for other purposes. Both permissions are subject to the condition that practitioners of TDM already have lawful access to the protected works. Rightholders can prevent TDM in non-commercial contexts only to the extent absolutely necessary by invoking the operability and safety of their systems, Article 3 (3) DSM-Directive, whereas it is possible to express a reservation in a machine-readable manner within commercial contexts, Article 4 (3) DSM-Directive. According to the European legislator, the interests of rightholders are affected by TDM research only to a minor extent: In any case, the member states are explicitly not to provide for compensation of rightholders for the acts of exploitation carried out to prepare corpora.[11]

Due to this obligation it must be legally possible in all EU member states to carry out research in terms of text and data mining on copyrighted works or databases in any case from June 2021 onward. By missing this deadline, member states risk infringement proceedings, Articles 258 ff. Treaty on the Functioning of the European Union (TFEU).

2.3 Remaining Legal Uncertainties: Reusing Copyrighted Corpora

It was and still is not only the legal uncertainty in terms of research on copyrighted works that was and is slowing down scientific progress, but also the uncertainty regarding the fate of research data after completion of the respective research, e. g. the scientific review at a later stage, the storage of research data and the reusability of corpora (Kleinkopf et al., 2021): According to Article 4 (2) DSM-Directive, corpora may be retained in commercial contexts only for as long as is necessary for the purposes of the TDM. In contrast, Article 3 (2) DSM-Directive does not provide for a time limit for retention in non-commercial contexts, but limits retention to the purposes of non-commercial scientific research (and also requires appropriate safeguarding). In this respect, it is up to the member states to implement this into national law in the most research-friendly way possible.

Regarding the question of a legal option to reuse the corpora, recital 15 of the DSM-Directive should find attention. This recital refers to Article 5 (3a) InfoSoc-Directive that authorizes member states to allow acts of reproduction of protected works and communicating them to the public for the purposes of non-commercial scientific research. Therefore, the national exceptions or limitations for those purposes could be applied (Kleinkopf et al., 2021). According to Article 25 of the DSM-Directive, member states are also explicitly allowed to go beyond the requirements of the DSM-Directive and grant extended authorizations on the basis of the InfoSoc-Directive. This includes that other national exceptions or limitations in favor of non-commercial scientific research are still applicable.

The combination of different copyright limitations is not new in European law: In "Eugen Ulmer/TU Darmstadt", the ECJ decided that it is possible to combine different exceptions and limitations under the InfoSoc-Directive, provided that the requirements of each are met.[12] The specific case concerned the combination of copyright exceptions and limitations under Article 5 (2b) and Article 5 (3n) of the InfoSoc-Directive. Because the exceptions and limitations of the InfoSoc-Directive continue to apply under the DSM-Directive, the idea behind this approach is to transfer this case law to Article 3 of the DSM-Directive and Article 5 (3a) of the InfoSoc-Directive (Kleinkopf et al., 2021). In addition, recital 15 of the DSM-Directive assumes the cumulation of exceptions and limitations under copyright law.

One limit is the three-step test under Article 5 (5) of the InfoSoc-Directive, which must be observed both in the context of legislation and in the context of judicial interpretation of the law (Stieper, 2009, p. 73 with further evidence).[13] The three-step test states that copyright limitations must be limited to certain special cases (step one), they must not interfere with the normal exploitation of the work (step two) and they must not unreasonably prejudice the legitimate interests of the author (step three). The scientific reuse of TDM corpora must be regarded as such a special case, furthermore, the primary market is not affected if only parts of the corpora are reused (Kleinkopf et al., 2021). The requirements for unreasonableness for rightholders as stated in the third stage tend to be regarded as high (Senftleben, 2004, p. 210 f.) and, in view of

[11] DSM-Directive, recital 17

[12] ECJ - Eugen Ulmer, ECLI:EU:C:2014:2196, No. 50 ff.

[13] German federal High Court of Justice (BGH), judgment of 11th July 2002 - I ZR 255/00, GRUR 2002, 963 – Elektronischer Pressespiegel

the worthiness of protection of scientific interests under Article 13 Charter of Fundamental Rights of the European Union, are not met at least in the case of an obligation to pay remuneration (Kleinkopf et al., 2021). This remuneration is often granted in general permissions of non-commercial scientific research, see e. g. § 60h of the German Copyright Law (Urheberrechtsgesetz).

2.4 Reusability of Copyrighted Corpora under German Law

The German legislator recently updated the German permission to use copyrighted works in favor of non-commercial, scientific text and data mining research, § 60d Urheberrechtsgesetz.[14] While the permission of acts of reproduction of protected works has been extended to commercial purposes, § 44b Urheberrechtsgesetz, the permission for scientific research has not been significantly extended: Although it is permitted to make the corpora accessible for peer review procedures and to retain them for research purposes, it is legally unclear whether the corpora may also be archived by third parties (Kleinkopf and Pflüger, to appear). Moreover, the German legislator missed the chance to add the possibility to make corpora explicitly reusable. The more general permission of exploiting copyrighted works under German Copyright Law is § 60c Urheberrechtsgesetz that implements Article 5 (3a) InfoSoc-Directive in national law. By applying § 60c on § 60d Urheberrechtsgesetz, it is possibly to reuse corpora at least partly. In detail, § 60c allows the usage of extracts that do not exceed 15% of the total work. It also allows to use individual short publications such as journal articles that are no more than 25 pages long completely.[15]

3 Related Work

The official opening of copyrighted works for research has only been implemented quite recently and copyright regulations have also changed several times. As such, much of previous work has been designed under different legal conditions. However, development of infrastructure or support software was generally focused on circumventing copyright in legal ways.

Research projects working with German corpora have established different ways of dealing with the legal situation. The Institute for German Language (IDS) hosts the German Reference Corpus DeReKo that comprises more than 50 billion words.[16] Access to the data is regulated by more than 200 individual license agreements with rightholders (Kupietz et al., 2018). However, access to DeReKo is still restricted: It is available "for non-commercial, scientific research by registered users and strictly within the query-and-analysis-only framework" (Kupietz et al., 2018, p. 4353). Consequently, researchers can never access full texts but only get results for specific queries with limited context. The situation is similar for the second large reference corpus for German, the DWDS corpora[17] (Geyken et al.). While the effort of individual license agreements is possible for large and long-term funded institutions as in these examples, this is not feasible for most individual projects with few employees on short-term contracts. Many projects therefore fall back to historic data that are already in the public domain or do not publish their corpora at all.

One recent suggestion that is implementable for small projects and individual researchers is the concept of derived text formats ("abgeleitete Textformate") by Schöch et al. (2020b), see also Schöch et al. (2020a). The authors propose methods to obfuscate the original text in a way that the result is no longer covered by the original copyright and may be published and shared freely. This includes, for instance, the publication of word (or n-gram) frequency lists or a text version with scrambled word order. These derived text formats allow for some types of analysis that are popular in the digital humanities, like stylometry or topic modeling.

Whether derived text formats or results for a specific query only are useful or access to more context is required depends, of course, on the research question. In the XSample project, we explore the different needs via two use cases from the humanities. The first use case is the project CAUTION[18] in literary studies that explores the phenomenon of unreliable narrators that are, for instance, lying or do only have limited knowledge about the narrated world. It can easily be seen that this phenomenon cannot be captured in word frequency lists, but requires a lot of textual context. The second use

[14] Bundesgesetzblatt (German federal law gazette), Bgbl. 2021 Teil I Nr. 27 p. 1204 ff.

[15] BT-Drs. 18/12329, p. 35

[16] https://www.ids-mannheim.de/digspra/kl/projekte/korpora/

[17] https://www.dwds.de

[18] https://dfg-spp-cls.github.io/projects_en/2020/01/24/TP-Caution/

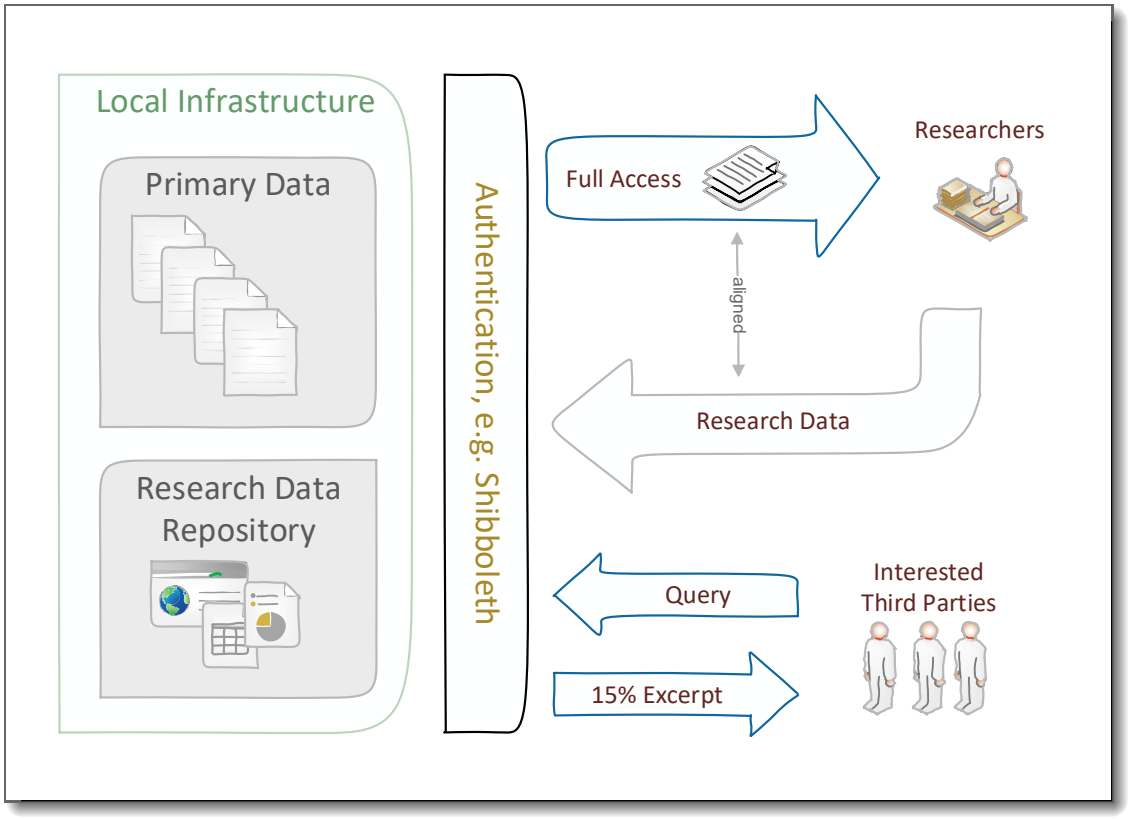

Figure 1: Architecture overview for XSample with the main parties involved and the general data flow.

case follows a linguistic research question about the academic language of linguistics and literary studies, replicating Andresen (to appear). The core of the analysis is based on a comparison of frequencies that can be performed on derived text formats. However, the interpretation of the quantitative results requires that findings can be recontextualized in the original texts, making full text access highly desirable, if not mandatory.

In sum, most qualitative approaches require as much context as possible right away, some quantitative approaches can be performed with little context, but their interpretation and evaluation has to rely on context as well. We therefore think that the given possibilities for making corpus data available can be complemented by a more flexible, individual approach. Our workflow is based on the right to distribute excerpts of texts and will be described in the following section.

4 The XSample Workflow

Our approach is based on combining § 60c and § 60d Urheberrechtsgesetz. Those regulations allow researchers the use of copyrighted material and libraries the passing on of excerpts of protected material up to a certain limit, respectively. With

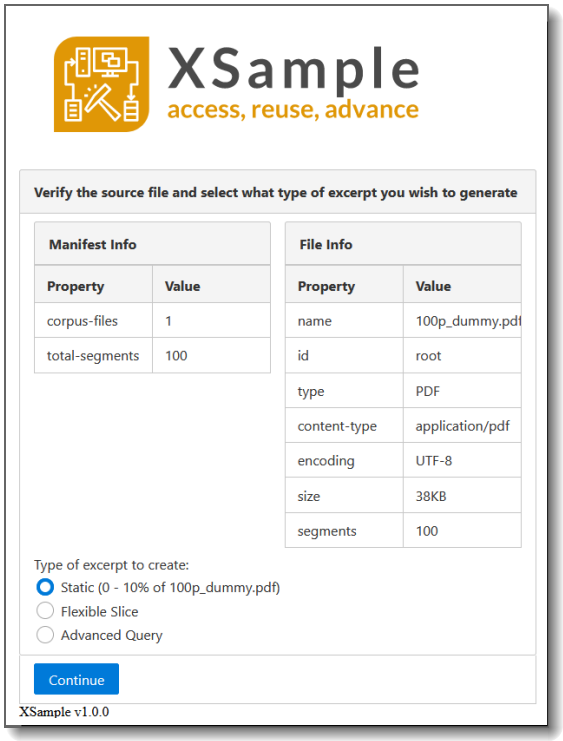

Figure 2: Screenshot of the XSample landing page.

libraries or other archiving institutions as central actors, the architecture depicted in Fig. 1 supports the entire workflow from copyrighted primary data to individualized excerpts for end users.

In an initial ingest step (upper loop in Fig. 1) researchers in an active project process copyrighted texts and deposit research data (intermediate annotations or finalized corpus resources) into the repository. The "aligned" label in Fig. 1 signifies that there has to be a way to map annotations or linguistic units back to the segments (pages) of the primary data they appear on or refer to. This reverse mapping is a crucial prerequisite for the query-driven excerpt generation described in Section 5.2.

Unsurprisingly, both use cases in our project highlighted the fact that typical corpus generation processes need to be adjusted in order to keep or restore this kind of mapping information. In both cases, text was originally extracted from PDF and EPUB documents either directly or via OCR, cleaned and then transformed into formats specific to the use case, losing the page mapping in the process. Subsequent modifications of the processes led to manual restoration of mappings as annotations in one case and automatic preservation as external (tabular) mapping files in the other.

Both the primary data and generated annotations[19] are stored in the shared or private domain, making them not directly available to the public. Additionally, special metadata following the XSample schema in JSON-LD[20] format is added to the repository in the public domain. This metadata makes the protected data findable and serves as entry point for end users that wish to receive excerpts of the corpus for inspection or evaluation (lower loop in Fig. 1). Actual end users of the XSample concept can be any kind of interested third parties, but are primarily expected to be other researchers that wish to evaluate the data for either reproducibility or suitability in the context of their own projects.

During the excerpt generation process users are subsequently redirected to the XSample web interface, visible in Fig. 2 in a horizontally compacted layout. There they are able to further specify exactly how the excerpt should be generated. Available options at this time include the following:

1. A **static** excerpt generation configurable by the original corpus creators within the XSample metadata file.
2. A user-defined continuous section or **slice**. Selection of this slice is achieved via a simple GUI with the same double-knob slider also used in the query approach in Fig. 4.
3. Filtering of (linguistic) annotations in the corpus based on user interests expressed in a formal **query**. This approach is explained in greater detail in Section 5.2 and also showcased in Fig. 4.

After successful completion of the XSample workflow, users are presented with a zip archive for download. Contained within this archive are the actual pages of the primary data that represent the excerpt itself alongside with annotations for those parts. At present the prototype implementation supports annotations in the tabular format of the CoNLL 2009 Shared Task (Hajič et al., 2009) and an extension for a TEI[21] subset is being worked on.

To conform with current law, the system must not give access to more than 15% of any particular resource[22] to individual users and therefore needs to track quotas. In order to minimize integration footprint and authentication overhead, the prototype implementation does not manage users itself, but relies on information provided by the Dataverse repository for identification and tracking of individual users.

5 Architecture

The XSample prototype is implemented completely web-based[23] and only consists of a few components in order to keep it lightweight and minimize the need for adjustments when integrating it into existing infrastructure. It is still under active development and while the current version can already serve a large portion of the basic XSample workflow, it is not yet feature complete. The source code is publicly available on GitHub[24] under an open source license. Sections 5.1 to 5.3 describe the integration into an existing Dataverse repository, excerpt generation and the handling of composite corpora in more detail.

[19] Strictly speaking this only concerns annotations which are still covered by copyright, i. e. those dissimilar to the derivational approach described by Schöch et al. (2020b).

[20] https://json-ld.org/

[21] Text Encoding Initiative https://tei-c.org/

[22] See Section 5.3 for details on how special cases are handled with respect to excerpt size limits.

[23] Using the Jakarta Server Faces (JSF) framework.

[24] https://github.com/ICARUS-tooling/xsample-server

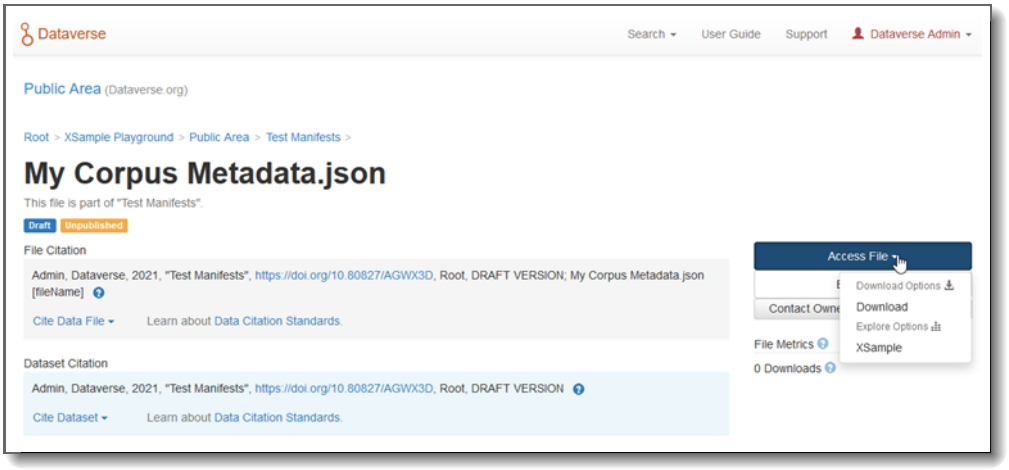

Figure 3: Screenshot showcasing the integration of XSample into the Dataverse user interface nd a metadata file as entry point for the XSample workflow. XSample is added as an *External Tool* for files of a specific content type and therefore appears as additional access option, visible in the dropdown menu to the right in the screenshot.

5.1 Dataverse Integration

The XSample prototype implementation is geared towards interfacing with a Dataverse[25] repository instance. Dataverse is an open source repository software built on JSF that is widely used for research data management and offers the granularity in access control required for the XSample workflow outlined in Section 4. Since Dataverse is also able to interface with existing authentication providers of the university or institute the system is deployed on, we can already rely on identification of unique users for excerpt quota tracking.

For integration, Dataverse's *External Tools API*[26] is used. It allows to register external web services for datasets[27] or files of specific content types in a way that does not require code modifications for the repository. External tools registered that way are then added as menu items when interacting with the Dataverse web interface. When used, they can send the user to a predefined server or service and also transmit various additional parameters, depending on their configuration. Possible parameters (all of which are used for XSample) are, among others, the resource ID, the public URL of the Dataverse repository or the user's API token.

Figure 3 shows an example snippet of the Dataverse interface for a metadata file[28] that serves as entry point to the XSample workflow. The "Access File" menu to the right contains the link to the external XSample server, usable to initiate the excerpt generation process.

5.2 Query-Driven Excerpt Generation

Depending on the use case, composing the excerpt of static (e. g. the first 15% of a corpus) or random elements might be of little benefit as there is no guarantee that passages or phenomena relevant to a user's particular interests are covered. In order to optimize excerpt generation, XSample includes a corpus query interface in the excerpt step (lower loop in Fig. 1) of the workflow.

In this interface, users can express their interest in a formal query language which the query backend evaluates on the annotation contents of the corpus to produce excerpt candidates. Candidates are determined by mapping the raw hits of a query result, for instance sentences when searching for a specific syntactic phenomenon, to actual segments in the primary data used for excerpt generation. In the case of primary data being in PDF format, the segments and candidates will be individual pages.

The distribution of candidates and their underlying raw hits over the entire corpus is subsequently visualized (cf. Fig. 4) to give users a preview of the expected size of their excerpt and to allow them to further refine the query. This visualization does, however, not contain the raw text or annotations

[25] The Dataverse Project, https://dataverse.org/

[26] https://guides.dataverse.org/en/latest/api/external-tools.html

[27] Within a Dataverse repository "datasets" and are used to organize file resources into logical groups.

[28] Due to an inconsistency in Dataverse 5.3, the version currently used for the XSample prototype, API tokens of users are not transmitted to external tools for public files. This issue has been raised in the Dataverse developer community and is being worked on. As a temporary workaround XSample metadata files in the test setup are therefore required to be private/drafts (cf. Fig. 3) until the inconsistency is fixed.

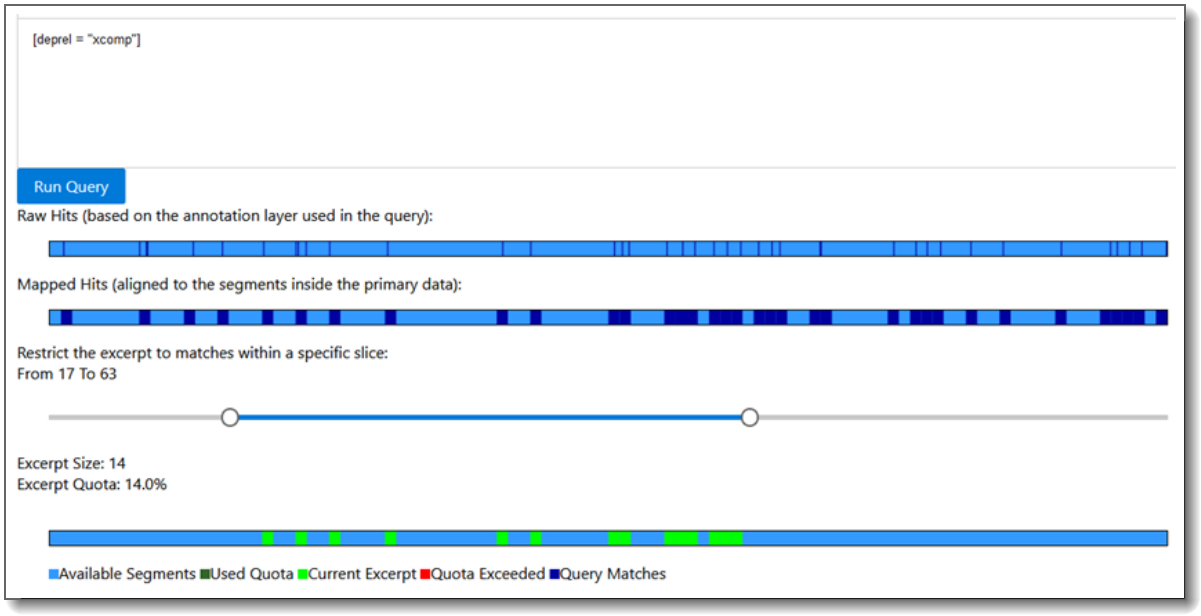

Figure 4: Screenshot snippet of the graphical interface for query-based excerpt generation. From top to bottom the screenshot shows components for query definition, raw hit visualization, mapped hit visualization and a control component with final excerpt visualization similar to the slice excerpt generation mentioned in Section 4.

of the corpus, as access to those is only granted when downloading a finished excerpt. Once satisfied with the result of their query, users can restrict the excerpt to be composed of candidates in a selectable region (similar to the "slice" option in Section 4) or request a random subset of candidates to be used. This way, the potential relevance of excerpts for individual users can be maximized.

The dependence on available annotations and specific formats in the corpus poses a major challenge for the implementation of the query-driven excerpt generation: Both the query backend and the component responsible for splitting annotation files for the excerpt must be able to handle a given set of corpus files to make this approach viable: The former to evaluate the query in the first place and the latter to split the annotation files when they are requested to be part of the excerpt.

For rapid prototyping we initially chose the query component of ICARUS (Gärtner et al., 2013) as evaluation backend, as it readily supports the CoNLL 2009 format and provides a simple bracket-style query language. In parallel, an alternative based on a more general middleware solution (Gärtner and Kuhn, 2018) is being worked on. Since the interface between the query backend and both the user interface and excerpt generation component is rather slim, plugging in a new implementation to support additional formats or query languages can be done fairly easy.

5.3 Composite Corpora

For simple corpora that consist of only a single copyrighted work, applying current regulations and size limits to the excerpt generation process is pretty straightforward. They apply directly to the entire corpus and in special cases such as certain journal articles or *small-scale* works the corpus is completely exempt from the 15% limit (cf. Section 2.4).

The situation becomes much more complicated when dealing with composite corpora, that is, corpora composed of a collection of individually copyrighted works: In such cases all rules and exceptions refer to contained works rather than the corpus as a whole.[29] As a direct result, the XSample server cannot deliver a blanket 15% excerpt for a composite corpus, but takes measures to ensure that the 15% limit is adhered to for each individual work. The server is informed of the actual corpus composition by the metadata (see Section 4) that serves as entry point for the XSample workflow. While the metadata schema allows for arbitrarily complex corpus compositions, the current server implementation is more limited: On the backend a nesting depth of one[30] is supported and the user

[29] While corpora could be viewed as databases themselves, researchers interested in making them available for reuse are typically consenting to copyright uses.

[30] A corpus may consist of multiple copyrighted works without further subdivision.

interface is only able to handle single-work corpora at this time but is currently being adjusted to match the backend capabilities. Especially the inclusion of works that are exempt from the 15% limit in composite corpora poses a serious challenge when developing the user interface while also aiming for a high degree of usability and intuitiveness.

6 Conclusion

In this paper we analyzed the evolving legal situation in Germany regarding copyright in the context of the European Digital Single Market, highlighting the shortcomings for research in text-based disciplines. We then proposed the XSample workflow as a concept for providing excerpts of copyrighted (text) material in order to support reproducibility and reusability. Our prototype implementation is web-based and initially designed to interface with Dataverse repositories only. It also features a query component to guide the excerpt generation process to more relevant samples based on a user's interests. However, having a very small integration footprint for both the repository and query components, it can also be adjusted to work with other systems. In the future we intend to widen the support for different corpus or annotation formats and also explore the possibility to apply the concept to material beyond text, such as audio or video resources.

Acknowledgments

This work was funded by the Ministry for Science, Research and the Arts in Baden-Württemberg (MWK) via project XSample through the funding program "BW-BigDIWA – Wissenschaftliche Bibliotheken gestalten den digitalen Wandel".

References

Melanie Andresen. to appear. *Datengeleitete Sprachbeschreibung mit syntaktischen Annotationen. Eine Korpusanalyse am Beispiel der germanistischen Wissenschaftssprachen*. Korpuslinguistik und interdisziplinäre Perspektiven auf Sprache (CLIP). Narr Francke Attempto.

Markus Gärtner and Jonas Kuhn. 2018. A Lightweight Modeling Middleware for Corpus Processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1087–1095, Miyazaki, Japan. European Language Resources Association (ELRA).

Markus Gärtner, Gregor Thiele, Wolfgang Seeker, Anders Björkelund, and Jonas Kuhn. 2013. ICARUS – An Extensible Graphical Search Tool for Dependency Treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Sofia, Bulgaria.

Christophe Geiger, Giancarlo Frosio, and Olekasndr Bulayenki. 2018. Text and Data Mining in the Proposed Copyright Reform: Making the EU Ready for an Age of Big Data? Legal Analysis and Policy Recommendations. *Int. Review of Intellectual Property and Competition Law (IIC)*, pages 814–844.

Alexander Geyken, Adrien Barbaresi, Jörg Didakowski, Bryan Jurish, Frank Wiegand, and Lothar Lemnitzer. Die Korpusplattform des "Digitalen Wörterbuchs der deutschen Sprache" (DWDS). *Zeitschrift für germanistische Linguistik*, 45(2):327–344.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA.

Felicitas Kleinkopf, Janina Jacke, and Markus Gärtner. 2021. Text- und Data-Mining: urheberrechtliche Grenzen der Nachnutzung wissenschaftlicher Korpora und ihre Bedeutung für die Digital Humanities. *MMR: Zeitschrift für IT-Recht und Recht der Digitalisierung*, pages 196–200. Open Access version available at http://dx.doi.org/10.18419/opus-11445.

Felicitas Kleinkopf and Thomas Pflüger. to appear. Digitale Bildung, Wissenschaft und Kultur – Welcher urheberrechtliche Reformbedarf verbleibt nach Umsetzung der DSM-RL durch das Gesetz zum Urheberrecht im digitalen Binnenmarkt? *Zeitschrift für Urheber- und Medienrecht (ZUM)*.

Marc Kupietz, Harald Lüngen, Pawel Kamocki, and Andreas Witt. 2018. The German Reference Corpus DeReKo: New Developments – New Opportunities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Benjamin Raue. 2018. Free Flow of Data? The Friction between the Comission's European Data Economy Initiative and the Proposed Directive on Copyright in the Digital Single Market. *Int. Review of Intellectual Property and Competition Law (IIC)*, pages 379–383.

Christof Schöch, Frédéric Döhl, Achim Rettinger, Evelyn Gius, Peer Trilcke, Peter Leinen, Fotis Jannidis,

Maria Hinzmann, and Jörg Röpke. 2020a. Abgeleitete Textformate: Prinzip und Beispiele. *RuZ - Recht und Zugang*, 1(2):160–175.

Christof Schöch, Frédéric Döhl, Achim Rettinger, Evelyn Gius, Peer Trilcke, Peter Leinen, Fotis Jannidis, Maria Hinzmann, and Jörg Röpke. 2020b. Abgeleitete Textformate: Text und Data Mining mit urheberrechtlich geschützten Textbeständen. *Zeitschrift für digitale Geisteswissenschaften.*

Martin Senftleben. 2004. Grundprobleme des urheberrechtlichen Dreistufentests. *GRUR International Journal of European and International IP Law*, pages 200–211.

Malte Stieper. 2009. *Rechtfertigung, Rechtsnatur und Disponibilität der Schranken des Urheberrechts*. Ph.D. thesis, Tübingen.

Lessons learned in Quality Management for Online Research Software Tools in Linguistics

Nils Diewald and **Eliza Margaretha** and **Marc Kupietz**
Leibniz Institute For The German Language, Mannheim, Germany
{diewald|margaretha|kupietz}@ids-mannheim.de

Abstract

In this paper, we present our experiences and decisions in dealing with challenges in developing, maintaining and operating online research software tools in the field of linguistics. In particular, we highlight reproducibility, dependability, and security as important aspects of quality management – taking into account the special circumstances in which research software is usually created.

1 Introduction

Research data and its management has been very much in the focus of linguistics and related disciplines in the digital humanities over the last 15 years. Although research tools were often mentioned in this context, they have played a subordinate role in terms of development, maintenance, operations, and requirements such as management plans. This discrepancy is probably unfavourable in general, however, it is especially problematic in the case of research based on language data, since there the data often cannot be used, analyzed or interpreted at all without specialized research software. In addition, in the software software, methodology and implementation are often intertwined, so that errors affect findings in ways that are not immediately apparent and often hard to reproduce. Furthermore, there are general problems with software developed in academic contexts, such as lack of training, lack of community support, lack of reputational incentives (Cohen et al., 2021), and lack of structures for sustainable maintenance and operation. Hence we focus on ensuring reproducibility, dependability, and security regarding quality management.

In the following, we report on our experiences with these challenges in operating and further developing the corpus query and analysis platforms KorAP and Cosmas II, which are used, among others, by the more than 40,000 users of the German Reference Corpus DeReKo (Kupietz et al., 2010, 2018) for a broad spectrum of corpus linguistic research.

2 Online Research Software Tools

In this article we distinguish between *research software* in general and *research software tools* in particular. While research software can be the result of a research project (especially in the field of computer science) or specific to the achievement of a single research result (see Hasselbring et al., 2020), we understand research software tools to be products that can be used to conduct or support research, i.e. which can be reused in a research area to a wide field of research objects and therefore requires constant *maintenance*[1]. By focusing on online tools, we refer to platforms that provide remote access to research data, and in the field of corpus linguistics, we are referring in particular to corpora that may be exclusively accessible via these platforms (e.g. for legal reasons, as is often the case for contemporary corpora; see Kupietz et al., 2010). These tools therefore require not only to be maintained but also to be *operated*.[2]

An important aspect of the use of computer-assisted methods in research is the reproducibility of results. Especially in the case of quantitative studies, not only the data are relevant for the reproducibility by different teams[3], but also the software used for the data analysis. Reproducibility including research software is a challenge (see Stodden and Miguez, 2014, regarding *best practices* on reproducibility in computational sciences), particularly in the case of online tools, since neither the underlying data nor the software used are usually in a state that is easy for the user to preserve and

[1] For a taxonomy on and criteria for research software, see https://rseng.github.io/rseng/.

[2] All other aspects listed here also predominantly apply to locally operated research software tools.

bundle with the research findings.

An additional aspect when providing online tools for (scientific) work is that operation must be guaranteed to be secure, uninterrupted and failure-free (in the best case), since on the one hand users depend on them for their research and on the other hand the validity of the results depend on their correctness.

3 Reproducibility

In order to make scientific studies with computer-assisted methods reproducible, not only prerequisites have to be fulfilled by the study design and the underlying data, but also the software should be designed and developed in such a way that it can be run on other systems in the form used for the initial study. This poses numerous challenges (Ivie and Thain, 2018), especially for online tools, but first and foremost, it requires

1. **licensing** that is as open as feasible,
2. transparent **versioning** of the software, and
3. a high degree of **portability**, so the software can be run independent of its environment and time.

It should be noted, of course, that reproducibility can basically only be achieved according to the *best effort* principle. Full control over and complete documentation of the environment can rarely be guaranteed (i.e. full control over the hardware, the operating system, the compiler or interpreter used, etc.).

3.1 Licensing

To enable autonomous reproduction of a study using computer-assisted methods, the software used must be accessible for everyone without restriction in the best case. In order for the implementation to be completely transparent, publication as open source is essential (Hasselbring et al., 2020). This not only helps with reproducibility, but can also reveal problems in the analysis, originating from errors in the software used (Goldacre et al., 2019).

The decisive criterion in the selection of software licences for the publication of KorAP modules was to restrict their use as little as possible and not without reason. Therefore, we have published most of the KorAP modules under the very liberal *BSD 2-clause license*[4] as open-source software on our Gerrit server[5] and on GitHub[6]. The biggest concern we had with our license choice was that the BSD licence does not exclude the subsequent removal of externally developed code. Our compromise solution to this consisted of pointing out in the licence notes (certainly not completely legally secure) that externally contributed code would also automatically fall under the BSD licence. For the case that substantial code parts were contributed by external developers, we also planned to introduce Contribution License Agreements (CLA) independently via corresponding hooks in GitHub and Gerrit. So far, we have not had any bad experiences with our licensing policy.

The decision on Cosmas II licensing was made against opening up the source code in the mid-1990s. Therefore all aspects of reproducibility were in the responsibility of the project owner.

3.2 Versioning

As software continues to be developed, there are differences that may disrupt reproducibility. At times backwards incompatibility is also inevitable. This is where versioning plays an important role. Versioning ensures consistent behaviour of a software by identifying and recording immutable states of a software (i.e. that are called versions) over time. By using version control systems, older versions of a software can be rebuilt. We use Git for versioning KorAP and SVN for Cosmas II. Moreover, we use GitHub for hosting the KorAP Git repository.

A version number or hierarchy (e.g. “1.5.2”) is often used to communicate changes between states. While the different levels of a version number can indicate different forms of change (compare with *semantic versioning*[7]), it is crucial that the behaviour of a released version of software is immutable, and that it can be restored at any time. In the case of KorAP, different components are in play (Diewald et al., 2016), which are operated and released independently of each other. In order to provide users with information about the whole software stack in use, a central API endpoint was designed that returns the individual version numbers of the com-

[3] We follow the terminology by the Association for Computing Machinery (2020). Please note, that the definitions for “reproducibility” and “replicability” were revised in version 1.1; see Plesser (2018) for an overview on the terminology.

[4] Also called *Simplified BSD License* or *FreeBSD License*

[5] https://korap.ids-mannheim.de/gerrit/admin/repos

[6] https://github.com/KorAP

[7] https://semver.org/

ponents involved.

Hashing and tagging can be used for identifying and naming particular changes respectively (Ivie and Thain, 2018). Git uses a hash function to create a unique identifier for each change or commit (Chacon and Straub, 2014). An accurate versioning system would involve the transparent communication of git commit hashes. Tags on these commits are often used to mark releases, but they are not necessarily unique. In KorAP, we take advantage of Git commit hashes as commit references, and Gerrit change-id (see sec. 4.1) to group commits belonging to the same review. Moreover, we use tags to mark releases both in KorAP and Cosmas II.

Releases can be made citable by archiving them in Zenodo, an open access repository for depositing research resources, as supported by GitHub[8]. Zenodo will issue a Digital Object Identifiers (DOI) for each release in a GitHub repository connected to it. We have published recent KorAP releases in Zenodo.

Beside software versioning, it is important to maintain API versioning to support clients using older APIs, especially when there are breaking changes in the newer APIs such as changes in the request or response formats and types. API versioning is commonly achieved by including the API version number in the service URL (i.e. URI versioning), adding a custom header or using the Accept header indicating the API version number. For KorAP API, we support API versioning by including the API version number within its service URL path.

3.3 Portability

Exact repetition of a computer-assisted scientific study would require "building the same program with the same compiler running on the same hardware and the same operating system" (Ivie and Thain, 2018, p. 63:4), which is rarely possible[9]. In the case of online tools, this is further complicated because the server architecture (both hardware and software) is seldom communicated to make attacks more difficult (see sec. 4). This requirement is even more ambitious to meet if a study is to be reproduced far in the future, when common hardware and software have changed to a great extent.

Therefore we instead aim at a high degree of portability of the system while ensuring equivalence of the final result. As a single criterion for equivalent behaviour, we consider the error-free run of all test suites of the system – including their dependencies (see sec. 4.3). To test the error-free operation in different environments, we use *Continuous Integration* for some components via GitHub (see sec. 4.3). To facilitate full building of the overall system locally, we provide both Vagrant[10] and Docker[11] files as our way to enable a virtualization approach (Howe, 2012).

At the beginning of the development of COSMAS II (Bodmer, 1996), the only requirement in terms of portability was the use of GNU-C, so it was usually necessary to access the existing environment to reproduce behaviour.[12]

3.4 Replicability

To additionally enable replicability of a computer-assisted study (i.e. re-implementation of the design by a different team using methods developed independently; see Footnote 3), further detailed documentation of the methods used is necessary. In the case of research software tools, this is sometimes part of the official documentation and thus does not require repeated explanation in publications. In many cases, such as the use of collocation measures, independently implemented methods already allow for the replicability of results (at least from the software point of view).

In KorAP, to facilitate replicability of studies, different query languages were implemented to allow comparison of results across multiple corpus analysis platforms. The use of virtual corpora enables the replicability of studies with different data bases (for example, considering comparable corpora; Kupietz et al., 2020); APIs, URL-encoded queries and various client libraries should help facilitate this. The functionality of KorAP and Cosmas II is documented in scientific publications, in manuals, in GitHub Readmes and Wikis, and commented in the code.

4 Dependability and Security

Avizienis et al. (2004) provide a taxonomy of dependable and secure computing, whose individual

[8] https://guides.github.com/activities/citable-code/

[9] This may still lead to different results in case of non-deterministic behaviour.

[10] https://github.com/KorAP/KorAP-Vagrant

[11] https://hub.docker.com/u/korap

[12] Only due to multiple migrations of the software, for instance from Solaris to a modern 64-bit Linux architecture, did the aspect of portability come to the front, albeit not in the context of facilitating reproducibility.

parts can be seen as cornerstones of quality management in the provision of software. One definition of dependability is "the ability of a system to avoid service failures that are more frequent or more severe than is acceptable", attributed with the *Availability*, *Reliability*, *Safety*, *Integrity* and *Maintainability* of the system. When taking security concerns into account, the *confidentiality* of the system is another important attribute (Avizienis et al., 2004, ch. 2.3).[13]

4.1 Availability and Reliability

Availability is defined as the "readiness for correct service" and reliability as its "continuity". With respect to security, this means a limitation to authorized actions only.

In order to keep the availability of KorAP at a high level, we use the service monitoring tool Icinga[14], which monitors the availability of the web services themselves and the status of the servers involved in order to be able to recognise emerging problems early on. To indicate planned downtimes, we currently use the start page of KorAP's web interface. In order to also be able to notify API users in the future, a corresponding message is planned for the functions for establishing connections in KorAP's client libraries. A fail-safe server structure with load-balancing and automatic switching between servers is not yet implemented for KorAP. This is also because with limited resources and in the context of research tools, we do not prioritise availability over reliability.

Concerning reliability in corpus linguistic research, in particular, it is a commonplace that interesting corpus findings are often initially artefacts of corpus composition and that corpus sampling and analysis cycles should therefore be regarded as an iterative process (Kupietz, 2016). One could add that the findings that remain after the elimination of confounders may also not represent true properties of the language domain under study, but also results of software bugs.

A proven means of reducing software errors is the use of code review (McIntosh et al., 2016), which in the context of research tools can also often take on the role of a classic peer review, at a fine granularity level. Among assisting systems, Gerrit Code Review[15] has become particularly well established in recent years.[16] Gerrit is an open-source team collaboration tool that is typically operated via a web interface. Developers can use it to review others' changes to their source code and comment, improve, augment, approve or reject those changes. It is tightly integrated with Git and can be considered an interface layer on top of Git.

The multiple-eyes principle not only helps to avoid errors and to be able to make design decisions collaboratively, but also ensures that code knowledge is distributed among several people, even if only individuals are responsible for a code base. In this way, personnel failures or absences do not necessarily lead to serious disruptions in operations, maintenance and development.

Admittedly, the use of Gerrit means an increase in the entry threshold, especially if the users are not yet very experienced in dealing with Git either. In addition, the review effort is certainly not to be neglected and the maintenance of Gerrit also involves a certain additional effort. Nevertheless, in view of the direct comparison with projects running in parallel without code review and the advantages already mentioned above, we are convinced that in the case of research tools, the use of a code reviewing system is worthwhile at least from a project size of 2-3 people. The initial and recurring costs incurred are more than made up for by the avoidance of errors and the distributed code knowledge. In our experience, another positive side effect of code reviews in terms of reliability is that commits are typically smaller and that pieces of parallel development strains can be more often combined without conflicting merges. This also increases the readability and traceability of the commit history.

To prevent unauthorized activities, we use integrate the OAuth 2.0 framework (Hardt, 2012) allowing users to grant their applications access to the KorAP APIs[17]. These applications may thus perform operations within the scope of their grants, e.g. searching and retrieving annotations, on behalf of the users.

4.2 Safety, Confidentiality, and Integrity

Safety is defined as the "absence of catastrophic consequences on the user(s) and the environment", confidentiality – as an attribute to security only – as "the absence of unauthorized disclosure of inform-

[13] The following definition quotes are taken from Avizienis et al. (2004).

[14] https://icinga.com/

[15] https://www.gerritcodereview.com/

[16] Gerrit is used by several prominent companies and projects, such as Android, SAP, LibreOffice, Volvo, Skia, TYPO3, ARM, and Wikimedia.

[17] https://github.com/KorAP/Kustvakt/wiki

ation" and integrity as the "absence of improper system alterations", which in regard to security includes unauthorized alterations.

Such security risks are in particular a threat to unmaintained online tools and can bring a very quick end to their operation. When potentially serious security vulnerabilities of an online tool become known and there are no longer any responsible parties, an academic institution usually has no choice but to take the tool offline immediately. Especially since, unlike research data management plans, research software management plans are probably a rarity. But even with tools that are still in development or maintenance, it is not obvious how to identify security problems with reasonable effort. In the case of KorAP, however, the publication of the source code on GitHub already helped us a lot without further ado. GitHub has an integrated security scanner that is enabled by default for public repositories. It detects so called *Common Vulnerabilities and Exposures (CVE)* in used dependencies and notifies the repository owners. Similar code scanner plugins for IDEs can serve as a supplement or alternative. There also seem to be open source approaches for such scanners, but we have no experience with them.

After having received a notification about CVE of a library and there is already an update to this that resolves the vulnerability, a common problem is that the update often also requires the update of other libraries, which again depend on other library updates and so on. This can mean that fixing a security vulnerability in one used library ultimately requires significant work to adapt the code to all the interface and behaviour changes of all the necessarily updated libraries (see sec. 4.3). Doing this quickly without taking the tool offline in the meantime can be a major challenge, even for software that is still under active maintenance.

Unless the use of external libraries is dropped, which however causes other costs and issues (see following section), the only secure option is to update library dependencies regularly and to factor this in from the outset as permanent maintenance costs for the operation of the online tool.

Tools that can help with the continuous updating of library dependencies have proven useful for certain projects (Mirhosseini and Parnin, 2017; Wessel et al., 2018). Dependabot[18], a so called dependency scanner, not only informs about updated dependencies, but also automatically makes merge requests to update, in the case of Java[19], Maven or Gradle project files. Following GitHub's acquisition of Dependabot in May 2019, the feature was added natively to GitHub. In the meantime, there is also an open-source project based on the original Dependabot core, that makes Dependabot available for GitLab.

We have been using Dependabot with GitHub since July 2020 for the KorAP component Kustvakt and have since received an average of 15 update pull requests per month. These update requests are particularly useful and easy to handle with corresponding continuous integration workflows, which can be used to automatically check whether the software is still buildable and operational with the updated library (see 4.3, below).

4.3 Maintainability

Maintainability is the "ability to undergo modifications, and repairs" of a system. Continuous maintenance is necessary not only to fix bugs, to accommodate changes in demand and to address security issues (sec. 4.2), but also to accommodate changes in the behaviour of client or server environments.

Modularity has proved to be useful to simplify a complex system by breaking it down into smaller independent modules or components. Smaller modules are easier to maintain and reuse than a complex system, since they are easier to understand, test and restructure independently of others. KorAP is composed of small independent components, both the service (Diewald et al., 2016) as well as the preprocessing pipeline.

Due to the increased complexity of the system, the maintenance of software dependency trees requires a great deal of effort, so that a constant trade-off must be made between reuse and reimplementation of functions (i.e. *re-inventing the wheel* vs *dependency hell*; see Abdalkareem et al., 2020). In KorAP, we decided to use both approaches, namely reusing functions and libraries as far as possible (as long as indirect dependencies are manageable) and re-implementing only when necessary (e.g. when existing functions are not adequate to cope with new requirements).

All KorAP components are equipped with comprehensive test suites. The test suites help us on the one hand to check new functions for proper beha-

[18] https://dependabot.com/

[19] Besides Java, various other programming languages are also supported, such as JavaScript and Python.

viour, and on the other hand to automatically ensure that program changes do not alter any previous behaviour (cf. Rafi et al., 2012). It is also significant for checking if updated dependencies break or modify the system flow. We also use the automatic detection of test coverage to identify deficiencies and gaps in the test suites. It should be noted that the maintenance of the test suites involves significant extra costs. In some areas, we recently employ additional *fuzzing* techniques (Miller et al., 1990) to test unexpected input to address shortcomings in manually crafted tests.

An important automatable instrument to control the functionality of software in the last instance are continuous integration (CI) tests. We now apply such CI test workflows to the production branches of almost all KorAP components, using *GitHub Actions*[20]. The workflows check if the tool can be built and apply all its available tests partly in different operating system environments (see sec. 3.3). Our CI workflows are usually configured in such a way that they are also automatically apply merge requests submitted via GitHub, so that it is immediately apparent if these affect the functionality of the software [21].

5 Discussion

Developing, maintaining, and operating online research software tools presents numerous challenges including ensuring reproducibility, dependability and security, as it requires knowledge and skills in many different areas. In fact, however, like research software in general, these tools are predominantly developed by individuals from academia – and less with a background in software development (Cohen et al., 2021). In our case, this background is in linguistics.

Cohen et al. (2021) introduce a model of four pillars considered essential to develop sustainable research software in such an environment, with *software development* being only one of the pillars. As additional pillars, they introduce *community* involvement for collaborative problem solving, *training* of researchers in software development techniques, and *policy* development for institutional support. We consider these points to be essential for the development of online research software tools as well, whereby we would add another pillar for maintaining and operating the system.

While the perception of the importance of software for scientific work is growing, development, maintenance, and operation is rarely associated with gaining scientific merit: "many activities are software maintenance – new functionalities or endless bug fixing – and hardly publishable" (Goble, 2014, p. 6). Anzt et al. (2021) therefore propose to accept contributions to open source projects (so-called "pull requests") as a new form of academic contributions to conferences[22], in order to increase the motivation to participate in the development of research software tools, which is beneficial for the wider scientific community.

Our remarks regarding quality management to ensure reproducibility, dependability and security of online research software tools should in no way be misunderstood as *best practices*. They are only meant to reflect our choices and experiences in these areas running the corpus analysis platforms KorAP and Cosmas II, whereby all these decisions were based on a cost-benefit calculation. Especially when creating research software, the effort to run a practicable quality management is often not compatible with the circumstances. We are however convinced that the aforementioned aspects are worth to be considered in the development, operation, and maintenance – maybe even in the planning – of online research software tools in general.

Acknowledgements

We thank our four anonymous reviewers for helpful comments on earlier drafts of the manuscript.

References

Rabe Abdalkareem, Vinicius Oda, Suhaib Mujahid, and Emad Shihab. 2020. On the impact of using trivial packages: an empirical case study on npm and PyPI. *Empirical Software Engineering (EMSE)*, 25:1168–1204.

Hartwig Anzt, Eileen Kuehn, and Goran Flegar. 2021. Crediting pull requests to open source research software as an academic contribution. *Journal of Computational Science*, 49:101278.

Association for Computing Machinery. 2020. Artifact review and badging version 1.1. Technical report, Association for Computing Machinery.

[20] https://docs.github.com/en/actions

[21] Consisting of multiple components developed in various programming languages and frameworks having distinct formats and structures, KorAP is too complex and not suitable for uniform code and comment styles and conventions.

[22] With a focus on High Performance Computing.

Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33.

Franck Bodmer. 1996. Aspekte der Abfragekomponente von COSMAS II. *LDV-INFO*, 8:142–155.

Scott Chacon and Ben Straub. 2014. *Pro Git*, 2nd edition. Apress, USA.

Jeremy Cohen, Daniel S. Katz, Michelle Barker, Neil Chue Hong, Robert Haines, and Caroline Jay. 2021. The four pillars of research software engineering. *IEEE Software*, 38(1):97–105.

Nils Diewald, Michael Hanl, Eliza Margaretha, Joachim Bingel, Marc Kupietz, Piotr Banski, and Andreas Witt. 2016. KorAP architecture – Diving in the deep sea of corpus data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3586–3591, Portorož/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2016/pdf/243_Paper.pdf.

Carole Goble. 2014. Better Software, Better Research. *IEEE Internet Computing*, 18(5):4–8. Conference Name: IEEE Internet Computing.

Ben Goldacre, Caroline E. Morton, and Nicholas J. DeVito. 2019. Why researchers should share their analytic code. *BMJ (Clinical research ed.)*, 367:l6365.

Dick Hardt. 2012. The OAuth 2.0 Authorization Framework. RFC 6749.

Wilhelm Hasselbring, Leslie Carr, Simon Hettrick, Heather Packer, and Thanassis Tiropanis. 2020. Open source research software. *Computer*, pages 84–88.

Bill Howe. 2012. Virtual appliances, cloud computing, and reproducible research. *Computing in Science Engineering*, 14(4):36–41.

Peter Ivie and Douglas Thain. 2018. Reproducibility in scientific computing. *ACM Computing Surveys*, 51:1–36.

Marc Kupietz. 2016. Constructing a corpus. In Philip Durkin, editor, *The Oxford Handbook of Lexicography*, pages 62 – 75. Oxford University Press, Oxford.

Marc Kupietz, Cyril Belica, Holger Keibel, and Andreas Witt. 2010. The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1848–1854, Valletta/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2010/pdf/414_Paper.pdf.

Marc Kupietz, Nils Diewald, Beata Trawiński, Ruxandra Cosma, Dan Cristea, Dan Tufiş, Tamás Váradi, and Angelika Wöllstein. 2020. Recent developments in the European Reference Corpus (EuReCo). In *Translating and Comparing Languages: Corpus-based Insights*, Corpora and Language in Use, pages 257–273, Louvain-la-Neuve. Presses universitaires de Louvain.

Marc Kupietz, Harald Lüngen, Paweł Kamocki, and Andreas Witt. 2018. The German Reference Corpus DeReKo: New Developments – New Opportunities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'18)*, pages 4353–4360, Miyazaki/Paris. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2018/pdf/737.pdf.

Shane McIntosh, Yasutaka Kamei, Bram Adams, and Ahmed E Hassan. 2016. An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering*, 21(5):2146–2189.

Barton P. Miller, Louis Fredriksen, and Bryan So. 1990. An empirical study of the reliability of UNIX utilities. *Communications of the ACM*, 33(12):32–44.

Samim Mirhosseini and Chris Parnin. 2017. Can automated pull requests encourage software developers to upgrade out-of-date dependencies? In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 84–94.

Hans Plesser. 2018. Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11.

Dudekula Mohammad Rafi, Katam Reddy Kiran Moses, Kai Petersen, and Mika V. Mäntylä. 2012. Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In *2012 7th International Workshop on Automation of Software Test (AST)*, pages 36–42.

Victoria Stodden and Sheila Miguez. 2014. Best practices for computational science: Software infrastructure and environments for reproducible and extensible research. *Journal of Open Research Software*, 2:1–6.

Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The power of bots: Characterizing and understanding bots in oss projects. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW).