A Workflow for Efficient and Interactive Analysis of the Google Books Ngram Corpus

Fabian Richter
fabian.richter@kit.edu
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

Klemens Böhm
klemens.boehm@kit.edu
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

Abstract

Across many humanities disciplines, researchers analyze how word frequencies change over time. The *Google Books Ngram Corpus* provides this data for roughly 6% of all printed books. However, current tools and methods for analyzing this massive corpus are limited, hindering researchers' ability to address all their questions effectively. We propose a flexible workflow adaptable to existing research problems from literature. This approach enables more comprehensive and efficient corpus analysis. It builds on the observation that many current methods involve two distinct, computationally unbalanced steps: subsampling (expensive) and analysis (typically less so). By separating these steps into a preprocessing stage (long-running) and an interactive analysis stage, our workflow gives way to efficiency when working with that corpus. We demonstrate this by replicating existing studies using our proposed workflow.

CCS Concepts

• **Information systems** → **Digital libraries and archives**.

Keywords

Google Books Ngram Corpus, Data Analysis, Digital Humanities.

ACM Reference Format:
Fabian Richter and Klemens Böhm. 2024. A Workflow for Efficient and Interactive Analysis of the Google Books Ngram Corpus. In *The 2024 ACM/IEEE Joint Conference on Digital Libraries (JCDL '24), December 16–20, 2024, Hong Kong, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3677389.3702604

1 Introduction

Analyzing the development of word frequencies over time is a valuable method for many branches of the humanities [16, 17, 20, 29, 32]. For such analyses, *Temporal n-Gram Corpora (TNC)* are used. Based on a collection of texts together with their respective publication dates, such corpora contain the frequencies of words and expressions over time. A prominent example is the *Google Books Ngram Corpus (GBNC)* [10, 12, 14]. The GBNC is based on millions of books that have been digitized in the *Google Books* project. It spans several centuries and contains at least around 6% of all books ever published [14]. The sheer size of the GBNC, which covers several orders of magnitude more tokens than any other corpus, makes it stand out as a highly interesting resource. This is evidenced by over 2,000 publications.[1] Most studies rely on very small subsets of the corpus – often only a few hand-picked n-grams, rarely more than 100. The information requirements in these studies are diverse, but we have observed common abstractions: While the following information requirements target different subsets of the corpus, they are structurally similar: (a) 'Given a list of terms related to *psychonanalysis*, sum their frequencies per year.' [7] (b) 'Given a list of n-grams with positive sentiment, sum their frequencies per year.' [19] Identifying similarities between frequencies is another common information need: 'Are frequencies of terms related to *depression* correlated with terms related to *digitalization*?' [21]

Many existing studies follow the same procedure: Researchers define a list of n-grams for analysis, then manually extract their frequencies from the full GBNC, and work on this subset. This workflow lacks dedicated tool support. Existing studies rely on custom-made toolchains that typically combine the *Google Books Ngram Viewer* (GBNV), and separate software packages for statistical analysis. Building and using these tools can be difficult for researchers without much technical background. To create a tool that supports that procedure, a combination of several challenges needs to be overcome:

- **C1** (Expressiveness): Users have diverse information needs for the GBNC, requiring flexible query systems. Identifying common patterns in these needs is crucial to keep this flexibility manageable.
- **C2** (Scalability): At 13.5 terabytes, the English subcorpus alone already is very large.
- **C3** (Ease of Use): Humanities users tend to lack a strong technical background, necessitating a user-friendly interface for corpus analysis.

Individual challenges tend to have workarounds, but a unified solution is not available.

Our contributions are as follows:

- a categorization of information needs from literature;
- a description of a workflow to identify, retrieve and analyze interesting subsets of the GBNC;
- a case study to highlight our workflow's ability to reproduce existing research results.

Paper outline: Section 2 presents information requirements from literature and Section 3 reviews existing query systems. Section 4 describes our workflow. Section 5 covers our case study. Section 6 summarizes our findings.

JCDL '24, December 16–20, 2024, Hong Kong, China

ACM ISBN 979-8-4007-1093-3/24/12
https://doi.org/10.1145/3677389.3702604

[1] https://scholar.google.com/scholar?q="google+ngram"+"humanities", last accessed October 1, 2024.

2 Related Work: Information Requirements and Insights

This section describes information requirements common in existing literature. It also introduces abstractions from these requirements, thus going beyond a conventional review of related work. Our literature study considers 11 articles, which we attributed to different research areas: *medicine* ([8, 13, 21]), *philosophy* ([7, 9, 19, 26]), and *social sciences* ([3, 11, 27, 30]).

The findings of our literature study are summarized in Table 1. While the concrete information requirements differ between the studies, the general structure of the works is similar. In all studies we reviewed, researchers analyzed only a small part of the GBNC. This highlights that the main computational effort of the studies comes from the construction of subcorpora, not the analysis itself.

Three distinct steps are required to satisfy the information requirements studied in almost all the publications: (a) generation of n-gram lists, to restrict the corpus to a relevant subcorpus; (b) extraction of frequencies for the n-grams from that list; and (c) downstream processing of these frequencies. Step (a) underscores that the GBNC's true strength is the breadth of texts and topics it covers, rather than its size. In many studies, researchers only require small subsets of the GBNC. The broadness of the corpus ensures that such subsets are available for all information needs we have found in the literature. Step (b) is necessary, as the temporal dimension of the GBNC is what makes it interesting to many researchers. Nonetheless, the GBNV presents frequencies only as plots and does not natively support extracting numerical values. Different approaches exist to mitigate this: (1) downloading parts of the GBNC data and implementing a custom filtering pipeline [19]; (2) extracting frequencies from the GBNV, either by using it as an API and extracting data from the HTML code as described in [4], or manually [8, 13, 21]; (3) taking screenshots of the GBNV and extracting the data from these using Computer Vision techniques [9]. Step (c) builds upon Step (b). The numerical values of the frequencies are required, as plots created by the GBNV are not sufficient to answer many research questions from literature.

Filter Conditions. Steps (a) and (b) in combination construct a subcorpus from the full GBNC. This filtering process can be of arbitrary complexity. Two important types of filters are the following:

- *Dictionary-based*: In all but one of the studies we have examined, researchers specify a list of n-grams they are interested in, usually related to the topic of their research question. Only the frequencies of these specific n-grams are considered in the further analysis.
- *Top-k*: In [19], the 5,000 most frequent 1-grams of the English and the Spanish GBNC are used for analysis.

The two types illustrate an important difference between filter conditions: Some can be evaluated *locally*, while others are inherently *global*. Given just one n-gram and its frequencies, one can decide whether the n-gram is contained in a dictionary, using only this *local* information. However, it is not possible to decide whether this n-gram is one of the k most frequent ones in the full corpus, *global* information about the corpus is required. Our case study in Section 5 will examine one study employing a dictionary-based filter and another one utilizing a top-k filter: (1) In [7], the authors estimate the prominence of psychoanalysis over time. They use a list of 55 English and 48 French terms related to the topic, to quantify the development in different countries. They then sum these frequencies and compare the resulting trajectories for French and English. (2) In [19], the 5,000 most frequent English and Spanish n-grams are used to track the development of 'rationality' in language. *Principal Component Analysis* is then used to find patterns and trends in the frequencies.

3 Related Work: Query Systems

Several systems exist to interact with the GBNC, with different levels of expressivity in their query languages. In this section, we review these systems.

GBNV. The GBNV debuted alongside the initial release of the GBNC, making it the first tool designed for GBNC analysis [14]. It lets users search for n-grams, retrieve plots of the frequencies, and perform simple computations – for example addition and multiplication – on these frequencies. While the GBNV is widely used, its functionality is limited, and some query types are not possible: (a) While the GBNV offers the wildcard character '*', its use has limitations. For example, users can search for '* science' and retrieve up to 10 results like 'natural science' and 'political science'. However, the GBNV rejects queries with more than one wildcard. (b) The GBNV can only plot 12 n-gram frequency series at a time. This makes the extraction of frequencies for many n-grams difficult. (c) The GBNV does not support filtering based on characteristics of the frequency series, for example the ten most frequent n-grams in a given year.

Other Query Systems. Other systems target at different weaknesses of the GBNV. *Google Books Advanced* [5] significantly extends the flexibility of wildcard search for the GBNC. *NgramQuery* [1] combines the *Web1T5* corpus [2], a predecessor of the GBNC, and *WordNet* [6]. This combination allows for queries along semantic dimensions, for example finding synonyms of a given n-gram. The *Conceptual History Query Language* (CHQL) [28] introduces an operator for nearest-neighbor search within the n-gram frequencies. *Slash/A* [23] enables researchers to analyze custom corpora in a format similar to the GBNC. It also introduces queries for similarity between n-gram frequencies [24]. None of these systems solve all the challenges we described in Section 1, falling short in either expressiveness (C1) or scalability (C2).

General Data Analysis Frameworks. While TNCs like the GBNC have a very specific structure and might profit from dedicated query systems, more general data analysis frameworks and tools are also applicable. It is possible to store TNCs inside relational database systems — e.g., as shown by Davies [5] — making them accessible to all analysis tools built on top of those. For the full GBNC, this approach is still prohibitive due to its sheer size, which would increase even more when including index structures for fast query execution. Similar considerations hold true for other analytical processing approaches as well: Storing the full GBNC locally is prohibitive for end users, independent of the concrete database or file format; analyzing the data without downloading it is impossible, as it is stored on servers not accessible for custom code execution; many researchers are only interested in small subsets

Table 1: Examples of research works using the GBNC

		List Generation	Frequency Extraction	Analysis	# of n-grams
medicine	[8]	✓	✓	simple arithmetic operations	< 50
	[13]	✓	✓	time series similarity	5
	[21]	✓	✓	correlation between topics	354 + inflections
philosophy	[7]	✓	✓	sum and mean	108
	[9]	✓	✓	peak, nadir, change	60
	[19]	✓	✓	principal component analysis, sum	10,000
	[26]	✓	✓	descriptive statistics, curve fitting	304
social sciences	[3]	✓	✓	correlation with historic events	6
	[11]	✓	–	–	10
	[27]	✓	–	–	50
	[30]	✓	✓	correlation with time	60 + inflections

of the GBNC, so there is no need for *Big Data* techniques once the relevant subcorpus has been constructed.

The *Archives Unleashed* project [18] has investigated how researchers from the humanities interact with web archives in general. The authors identified four main activities, which they dubbed the *Filter, Extract, Aggregate, Visualize* (FEAV) cycle. Their results coincide with ours, we refine them to the specific case of TNCs.

4 Workflow

Building on information needs identified in Section 2, Figure 1 presents our workflow to address them. We provide a brief overview, followed by details on file formats and the two workflow phases.

Our proposed workflow is based on a key observation: Analyzing the GBNC requires two distinct phases. In the first phase, the full corpus is restricted to a subcorpus relevant to the specific research questions. This corresponds to the first two steps described in Section 2. This subcorpus is then subjected to deeper analysis in a second phase, the third step in Section 2. The two phases involve vastly different computational demands. The first processes the entire GBNC, which spans several terabytes. Constructing a subcorpus entails downloading and extracting this massive dataset. In contrast, the second phase only works with the much smaller, filtered subcorpus. Though Phase 2 often involves more complex operations, the significantly smaller data size makes its computational cost negligible. This renders Phase 2 interactive, facilitating researcher experimentation and evaluation of diverse information needs. Conversely, the time-consuming Phase 1, processing the entire GBNC, only needs occasional reruns when the subcorpus of interest changes.

File Formats. The GBNC is available for download as a collection of compressed plain-text *csv*-files. These are human-readable, but not suitable for efficient automatic analysis. Calculating the total frequency of an n-gram, for instance, requires scanning an entire line, splitting it by *tab* characters, and then iterating through the year entries (up to 500). For a single line or a small subcorpus, this is feasible – but when parsing a larger number of lines, this operation causes notable overhead. To optimize performance, our workflow includes the conversion of the GBNC's plain-text files into the binary *Apache Parquet* format [25]. Then, only one expensive pass over the original files is necessary, and further processing can take place on the more efficient binary representation.

Subcorpus Construction. To construct a subcorpus, one first has to define a filter condition. These can take many different forms, for example those described in Section 2. To carry out the filtering, three distinct operations have to be performed: download of the compressed data, decompression, and the filtering itself. In practice, these three operations cannot be performed sequentially, as the size of the corpus and the resulting memory consumption would become prohibitive very quickly. Instead, we propose to download only a part of the corpus, extract and filter it, then continue with the next part in a streaming fashion.

Depending on the filter condition, the filtering process is of different complexity. A core distinction is between *local* and *global* filter conditions, as described in Section 2. Local filtering readily benefits from parallelization. However, global filters necessitate a more complex approach, with two main options available: (1) Process n-grams and their frequencies in a streaming fashion, fully sequentially. At all times, keep a set of all previously seen candidates that can potentially satisfy the filter condition. Remove n-grams from this set as soon as they do not fulfill the condition any more. (2) Process n-grams and their frequencies in batches. For each batch, evaluate the filter condition as if the batch were the full corpus. Create new batches from the result and repeat until only the final result remains. This is possible for *algebraic* and *distributive* filter conditions (e.g, top-k, dictionary, ...), while the evaluation of *holistic* conditions (e.g., top-x%) would require information exchange between batches. In our literature review, we only found filter conditions of the first kind.

Considering the size of the corpus, the first, fully sequential approach is infeasible. We therefore propose to follow the second approach, filtering batches in parallel. In our case, the batches are purely structural and defined by the way Google split the GBNC into smaller files for download. We have observed that a single batching step suffices for all filter conditions in Section 2. The intermediate

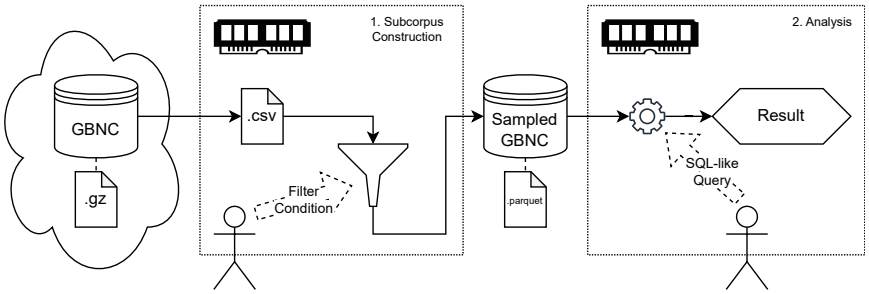

Figure 1: Our proposed workflow

results are then small enough to be combined into a single batch for the final iteration.

Our solution hinges on two assumptions: Batch-wise filtering requires the batch and intermediate results to fit in main memory. This limits parallelism, as the number of concurrent filter operations depends on both the number of CPU cores and available memory (*Memory Constraint*). Batch-wise filtering must significantly reduce the number of n-grams. Otherwise, local filtering will not sufficiently shrink subsequent batches, preventing a single-step global filtering evaluation (*Effective Filtering*). We demonstrate the validity of both assumptions in the following. As batches, we use the GBNC files as distributed by Google. The files, when downloaded, usually range between 400 and 800 MiB, and up to 4 GiB when extracted. All filter conditions identified in Section 2 can be computed with constant memory overhead per n-gram. Existing literature employs highly selective filtering conditions. In both dictionary-based and top-k filtering, only a tiny fraction of the millions of n-grams per file remains. In our experiments, the resulting filtered set is usually smaller than one unfiltered file.

Analysis. Expressing the diverse information requirements from literature requires a sufficiently flexible query language. For our workflow, we have chosen not to limit the choice of query languages and analysis systems, but instead opted to present the results of the filtering step as an *Apache Parquet* [25] file. This flexible format can be read and analyzed by different analysis frameworks, for example *pandas* [22], *Apache Spark* [31] or *DuckDB* [15].

5 Evaluation

In this section, we present a case study implementing existing research using our approach. We use our workflow to reproduce the results of two existing articles with two different filter conditions. The research questions of these studies are described in Section 2, we now focus on the technical aspects. The case study was run on a consumer-grade *Lenovo Thinkpad P14s*, with an *AMD Ryzen 7 PRO 5850U* CPU, and 16 GiB of RAM. The computer's internal SSD achieves write speeds of around 3 GiB/s. The maximum available download speed was around 100 MiB/s.

Haslam et al. [7]. In this study, the authors define a list of n-grams related to psychoanalysis in two different languages and consider the summed frequencies. Since this filter condition is simple, it does not require any programming effort – the user only has to supply a list of n-grams. Our implementation of the workflow downloads, extracts and filters the data necessary for this study in 36 minutes. This is possible because, when using the dictionary-based filter, one does not need to download the entire corpus: The n-grams are ordered alphabetically, and the boundaries per batch are known. The subsequent analysis and recreation of the figures in [7] takes less than 2 seconds, with the resulting *Parquet*-files being roughly 85 KiB in size.

Scheffer et al. [19]. In this study, a top-k filter condition is used, with $k = 5,000$. The top-k filter condition is implemented within our workflow, so no further programming is required to obtain the relevant data. Our workflow downloads, extracts and filters the relevant parts of the GBNC in 42 minutes, the resulting files take up 29.9 MiB of disk space. For this case study, one needs to download all 1-grams of the English and Spanish GBNC, respectively — but not the longer n-grams. In total, this leads to the download and extraction of 15.5 GiB, which takes around 38 minutes, the vast majority of computation time, while the actual filter operation takes only 4 minutes. To analyze the data, the authors of [19] use *Principal Component Analysis* (PCA). Our workflow focuses on streamlining data extraction, it currently excludes implementations of PCA or other complex statistical methods. Libraries for these methods can be connected to our workflow or applied to the resulting files.

6 Conclusions

The Google Books Ngram Corpus (GBNC) is a fascinating resource for researchers from the humanities, but analyzing it poses a number of technical challenges. In this paper, we presented an end-to-end workflow for such analyses. We reviewed existing studies that use the GBNC and derived a set of abstract information requirements needed for these studies. We devised a workflow that is able to satisfy these information requirements on standard consumer hardware. We replicated existing studies on the GBNC within our workflow, demonstrating that it solves the challenges from Section 1: Our solution is applicable in practice, even on standard consumer-grade hardware. The case study has shown that filter conditions from existing literature can be evaluated in less than 1 hour, with subsequent analyses taking mere seconds.

Acknowledgements

This work was supported by the pilot program Core-Informatics of the Helmholtz Association (HGF).

References

[1] Martin Aleksandrov and Carlo Strapparava. 2012. NgramQuery – Smart Information Extraction from Google N-gram using External Resources.. In *LREC*. 563–568.

[2] Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Ver. 1. *LDC2006T13, Linguistic Data Consortium, Philadelphia* (2006).

[3] Paul Caruana-Galizia. 2016. Politics and the German language: Testing Orwell's hypothesis using the Google N-Gram corpus. *Digital Scholarship in the Humanities* 31, 3 (2016), 441–456.

[4] Jason Chumtong and David Kaldewey. 2017. Beyond the Google Ngram Viewer. Forum Internationale Wissenschaft.

[5] Mark Davies. 2014. Making Google Books n-grams useful for a wide range of research on language change. *International Journal of Corpus Linguistics* 19, 3 (2014), 401–416.

[6] Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

[7] Nick Haslam and Lotus Ye. 2019. Freudian slip? The changing cultural fortunes of psychoanalytic concepts. *Frontiers in Psychology* 10 (2019), 468468.

[8] Tado Jurić. 2022. Using digital humanities for understanding COVID-19: lessons from digital history about earlier coronavirus pandemic. *medRxiv* (2022), 2022–02.

[9] Pelin Kesebir and Selin Kesebir. 2012. The cultural salience of moral character and virtue declined in twentieth century America. *The Journal of Positive Psychology* 7, 6 (2012), 471–480.

[10] Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the Google Books Ngram Corpus. In *Proceedings of the ACL 2012 System Demonstrations*. 169–174.

[11] Dag Øivind Madsen and Kåre Slåtten. 2022. The possibilities and limitations of using Google Books Ngram Viewer in research on management fashions. *Societies* 12, 6 (2022), 171.

[12] Jason Mann, David Zhang, Lu Yang, Dipanjan Das, and Slav Petrov. 2014. Enhanced search with wildcards and morphological inflections in the Google Books Ngram Viewer. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 115–120.

[13] Christopher B. Menadue. 2020. Pandemics, epidemics, viruses, plagues, and disease: comparative frequency analysis of a cultural pathology reflected in science fiction magazines from 1926 to 2015. *Social Sciences & Humanities Open* 2, 1 (2020), 100048.

[14] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science* 331, 6014 (2011), 176–182.

[15] Mark Raasveldt and Hannes Mühleisen. 2019. DuckDB: an embeddable analytical database. In *Proceedings of the 2019 International Conference on Management of Data*. 1981–1984.

[16] Steffen Roth. 2016. Fashionable Functions: A Google Ngram View of Trends in Functional Differentiation (1800-2000). In *Politics and Social Activism: Concepts, Methodologies, Tools, and Applications*. IGI Global, 177–203.

[17] Steffen Roth, Carlton Clark, and Jan Berkel. 2017. The fashionable functions reloaded: an updated Google Ngram view of trends in functional differentiation (1800-2000). In *Research Paradigms and Contemporary Perspectives on Human-Technology Interaction*. IGI Global, 236–265.

[18] Nick Ruest, Jimmy Lin, Ian Milligan, and Samantha Fritz. 2020. The Archives Unleashed project: technology, process, and community to improve scholarly access to web archives. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. 157–166.

[19] Marten Scheffer, Ingrid van de Leemput, Els Weinans, and Johan Bollen. 2021. The rise and fall of rationality in language. *Proceedings of the National Academy of Sciences* 118, 51 (2021), e2107848118.

[20] Amelia C. Sparavigna and Roberto Marazzato. 2015. Using Google Ngram Viewer for scientific referencing and history of science. *arXiv preprint arXiv:1512.01364* (2015).

[21] Gisbert Wilhelm Teepe, Edda Magareta Glase, and Ulf-Dietrich Reips. 2023. Increasing digitalization is associated with anxiety and depression: A Google Ngram analysis. *PLOS One* 18, 4 (2023), e0284091.

[22] The Pandas development team. 2024. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.10957263

[23] Velislava Todorova and Maria Chinkina. 2014. Slash/A n-gram tendency viewer – Visual exploration of n-gram frequencies in correspondence corpora. In *Proceedings of the ESSLLI*. 229–239.

[24] Velislava Todorova and Maria Chinkina. 2018. Significance Filters for N-gram Viewer. In *Visualisierung sprachlicher Daten*. Heidelberg University Publishing, 301–314.

[25] Deepak Vohra. 2016. Apache Parquet. *Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools* (2016), 325–335.

[26] Melissa A. Wheeler, Melanie J. McGrath, and Nick Haslam. 2019. Twentieth century morality: The rise and fall of moral concepts from 1900 to 2007. *PLOS One* 14, 2 (2019), e0212267.

[27] Klaas Willems. 2013. 'Culturomics' and the representation of the language of the Third Reich in digitized German books. *Interdisciplinary Journal for Germanic Linguistics and Semiotic Analysis* 18, 1 (2013), 87–99.

[28] Jens Willkomm, Christoph Schmidt-Petri, Martin Schäler, Michael Schefczyk, and Klemens Böhm. 2018. A Query Algebra for Temporal Text Corpora. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. 183–192.

[29] Nadja Younes and Ulf-Dietrich Reips. 2018. The changing psychology of culture in German-speaking countries: A Google Ngram study. *International Journal of Psychology* 53 (2018), 53–62.

[30] Nadja Younes and Ulf-Dietrich Reips. 2019. Guideline for improving the reliability of Google Ngram studies: Evidence from religious terms. *PLOS One* 14, 3 (2019).

[31] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, et al. 2016. Apache Spark: a unified engine for big data processing. *Commun. ACM* 59, 11 (2016), 56–65.

[32] Rong Zeng and Patricia M. Greenfield. 2015. Cultural Evolution over the Last 40 Years in China: Using the Google Ngram Viewer to Study Implications of Social and Political Change for Cultural Values. *International Journal of Psychology* 50, 1 (2015), 47–55.