

Orbit: Open Re-implementation of Biomedical Literature APIs

Shrestha Ghosh
University of Tübingen
Tübingen, Germany

Mert Okur
University of Tübingen
Tübingen, Germany

Harrisen Scells
University of Tübingen
Tübingen, Germany

Abstract

Information retrieval experimentation with biomedical literature, despite an important area of research that can have a high impact, is often gated by a high barrier of entry. This is due to a variety of reasons including the requirement of bespoke document processing pipelines, complex indexing pipelines, and domain-specific query languages. While biomedical literature databases with freely available API access do exist, e.g., the Entrez API for PubMed and the ClinicalTrials.gov API, the use of such APIs is limiting: rate limits, restrictions on the number of documents that can be retrieved per request, and no way to extend or add functionality. The usual option for overcoming these limitations is to scrape or download the data, index it with an off-the-shelf search engine, and use whatever query language is implemented by the search engine. These off-the-shelf options are insufficient for reproducing retrieval in biomedical tasks, such as, systematic reviews, clinical-trial design and evidence-based clinical decision-making, that require retrieval from multiple sources and complex query mechanisms. We provide a drop-in, extendable, and open-source clean room re-implementation of two commonly used biomedical literature APIs that dramatically lowers the barrier of entry to experimentation in retrieval-based biomedical tasks. An overview of the system with example usage and video showcase is available at <https://orbit.health-nlp.com>.

CCS Concepts

• **Information systems** → *Information systems applications; Digital libraries and archives; Information retrieval.*

Keywords

Biomedical Literature, Domain-specific Search, APIs

ACM Reference Format:

Shrestha Ghosh, Mert Okur, and Harrisen Scells. 2026. Orbit: Open Re-implementation of Biomedical Literature APIs. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26)*, November 07–11, 2026, Rome, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3799682.3840238>

1 Introduction

Performing information retrieval experiments with biomedical literature is essential for tasks such as systematic reviews and clinical trial matching. Despite being freely available with API access, their usage remains challenging for numerous reasons. These APIs, such as the Entrez API for PubMed and the ClinicalTrials.gov API impose rate limits to ensure fair access. However, this severely restricts the

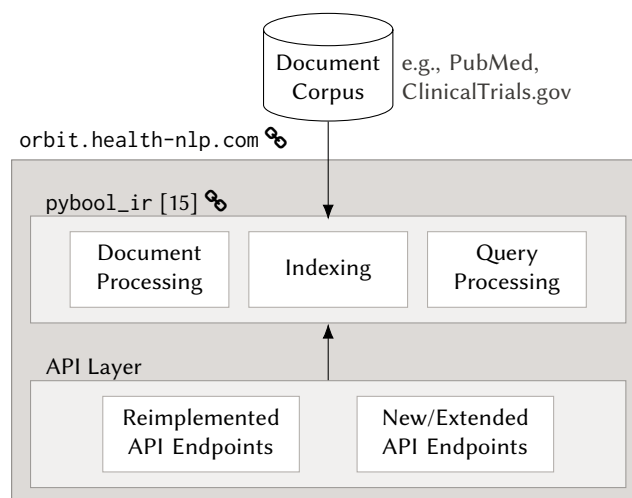


Figure 1: High-level architecture diagram of Orbit. Using pybool_ir as the base, we re-implement biomedical literature APIs, enabling the modification and extension of these highly used services.

number of documents retrieved per request, and there is no way to add new functionality, making the retrieval process extremely time-consuming.

The other option to download and index the data locally is equally challenging. Indexing literature databases require specialised processing of not just the raw data, but also meta-data such as the publication date and domain-specific ontologies such as MeSH, with often little documentation about schema or type information. These indexes additionally require constant update to deal with the rate at which documents are published. Meanwhile on the retrieval side, implementing query languages is error-prone and, depending on the search engine used, not all syntax may be supported (e.g., adjacency operators that work like a more restrictive AND operator) or may have subtle differences compared to the original search engine. Given the complexity of the setup itself, the usual route is to use off-the-shelf tools. However, these off-the-shelf options become insufficient when one wants to have more control over the data processing, such as combining multiple sources, or query mechanisms, such as processing complex Boolean queries.

Biomedical literature has been used in information retrieval research for many decades, and continues to be used in active research. Although several specific instances have been mentioned above, it is also worthwhile mentioning in numbers, the amount of shared tasks alone that use some form of biomedical literature: four editions of TREC Precision Medicine, three editions of TREC Clinical Trials, two editions of TREC BioGen, three editions of CLEF Technology Assisted Reviews, and fourteen editions of BioAsq.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CIKM '26, Rome, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
<https://doi.org/10.1145/3799682.3840238>

Although not an exhaustive list, it does certainly beg the question: *how do we lower the barrier of entry to enable those that do not have the expertise or infrastructure available to process and index the large underlying document collections?*

Orbit is positioned to allow not just individual researchers to more easily access and experiment on these large document collections, but also for shared task organisers to host specific slices or instances of the data to ensure all participants are working on the same collection. Furthermore, due to the use of standard infrastructure, these slices or instances of the data can be packaged, archived, and distributed much more easily. Thus, participants can work directly with the data without worrying about how to best download and/or index rather than via an API.

Related Work. PubMed is a commonly used biomedical literature search engine for information retrieval research.¹ For example, the underlying document collection is used in shared tasks such as TREC Clinical Decision Support,² TREC-BioGen,³ CLEF BioASQ,⁴ and in information retrieval research such as work on automating systematic reviews, such as the CLEF Technology Assisted Reviews tracks [5–7] and a more recent task on systematic review updates [1]. The easiest approach to doing research in this space is to use the API provided by PubMed, called Entrez [14]. Entrez Direct is a command-line toolkit for accessing Entrez [8]. Europe PMC API is another service that provides access to PubMed and other sources.⁵ ClinicalTrials.gov is yet another commonly used search engine for biomedical literature.⁶ For example, the underlying document collection is used in shared tasks such as TREC Clinical Trials,⁷ TREC Precision Medicine,⁸ and in information retrieval research such as trial search [13, 17] and trial structuring [2, 3]. The Aggregate Analysis of ClinicalTrials.gov (AACT) database,⁹ is a publicly available relational database of clinical trials parsed from the ClinicalTrials.gov database accessible via Postgres clients to its cloud-based database and can be downloaded as well. The AACT setup, while useful for data analysis, is not suitable for running retrieval experiments.

However, using such APIs comes with several limitations, namely rate limiting, restrictions on the number of documents able to be retrieved per request, no way to extend or add additional functionality (new ranking models, query processing techniques, etc.), and being subject to arbitrary API changes. Recently the Entrez API for PubMed was changed such that it was no longer possible to request more than 10 000 documents.¹⁰ The ClinicalTrials.gov API also imposes limitations on query endpoints to return a maximum of 10 000 results. The field values endpoint limits a maximum of only 250 values and does not support search expressions. Specifically for PubMed, notices about lapses in government funding is worrying as countless researchers depend on the API for not just information retrieval research but many areas of biomedical research [4].

¹<https://pubmed.ncbi.nlm.nih.gov/>

²<https://www.trec-cds.org/2016.html>

³<https://trec-biogen.github.io>

⁴<https://clef2026.clef-initiative.eu/labs/bioasq/>

⁵<https://europepmc.org/restfulwebservice>

⁶<https://clinicaltrials.gov/>

⁷<https://www.trec-cds.org/2023.html>

⁸<https://www.trec-cds.org/2020.html>

⁹<https://aact.ctti-clinicaltrials.org/>

¹⁰<https://www.ncbi.nlm.nih.gov/books/NBK25499/#chapter4.retmax>

Table 1: Overview of the number of documents indexed across the services, and the API endpoints implemented for them.

Service	Corpus Size (docs)	Original APIs	Orbit APIs
PubMed (Entrez)	40 131 104	🔗 ESearch	🔗
		🔗 EInfo	🔗
		🔗 ESummary	🔗
		🔗 EFetch	🔗
ClinicalTrials.gov	570 377	🔗 Studies	🔗
		🔗 Single Study	🔗
		🔗 Data Model Fields	🔗
		🔗 Search Areas	🔗

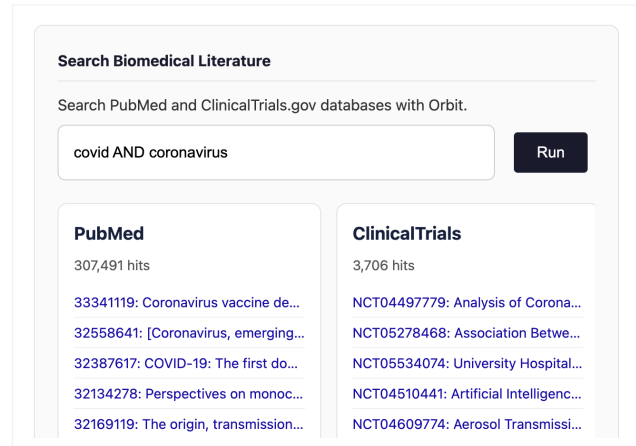


Figure 2: User interface of Orbit with the search results on covid AND coronavirus from the PubMed and ClinicalTrials.gov indexes linked to the source documents.

Our Contribution. Given the widespread use of biomedical literature APIs for conducting research, it is important that access to these APIs remains open and modifiable. As such, in this paper, we present, **Orbit**, our open re-implementation of the PubMed Entrez and ClinicalTrials.gov APIs. Figure 1 provides an overview of the components in Orbit. Crucially, Orbit provides a drop-in replacement for these APIs in a single command-line program. We provide a docker-based pipeline that downloads, indexes, maintains, and enables replication and extension of existing APIs without any limitations imposed. We publicly release the code with a permissive MIT license.¹¹ In addition to being fully self-hostable, we also provide an instance of our API accessible directly at <https://orbit-api.health-nlp.com> and via a user-interface at <https://orbit.health-nlp.com>, see Figure 2, for testing and demonstration purposes. Orbit, supports the (1) re-implementation of new APIs (including those outside of or relating to biomedicine) and (2) the extension of existing APIs with new features that are not available in the original versions.

¹¹<https://github.com/health-nlp/orbit>

2 Overview of Orbit

Table 1 provides an overview of the biomedical literature APIs that have been implemented, the corpus size of each of these services, and individual links to a description of the original API endpoints and our re-implementations.

Target Users. The main problem that Orbit addresses is to provide unrestricted access to APIs that are commonly used in biomedical literature research. These APIs are used extensively, not just for information retrieval and natural language processing research, but in many areas of bioinformatics, where researchers need to retrieve biomedical studies in bulk for analysis. Limitations or restrictions with these APIs can be a barrier to retrieval tasks and their functionality is not flexible to changes or extensions.

We envision the two main groups as the target users of Orbit. The first group is information retrieval and natural language processing researchers that are investigating how to improve information access in biomedical literature. The second group is end-user biomedical researchers who consume biomedical APIs for their own research.

Implementation Details. Figure 1 provides a high-level architecture of Orbit. The re-implementation pipeline involves document processing, indexing, and query processing setup using `pybool_ir` [15]. This library was demonstrated to reproduce the underlying PubMed search engine and has been used successfully for retrieval experiments [16]. However, productionising new methods using this library is difficult, especially for putting it into the hands of end-users. Orbit provides a way to allow end users of existing biomedical APIs to access state-of-the-art methods without needing to adapt to new tools or workflows.

Once the local indexes are up, we use the `fastapi` library to re-implement the API endpoints [12]. The resulting API is fully documented using the OpenAPI specification, providing a way for users to try the API interactively.¹² As the `ClinicalTrials.gov` API also supports this specification, it is easily possible to compare it to our re-implementation using the links in Table 1.

The `pybool_ir` library uses Lucene indexes to back the API implementations in Orbit. This makes it possible to easily extend existing and implement new Orbit APIs by using existing information retrieval frameworks like `pyserini` [10] and `pyterrier` [11] (as demonstrated by Scells and Potthast [15]). Orbit also supports the ongoing update of these indexes, and can be configured to periodically download and index newly published studies.

Finally, all re-implemented services in Orbit are built into docker containers. We have implemented each step of the data acquisition, processing, indexing, and hosting pipeline to depend on each other. This allows the entire pipeline to run from a single command. Taking inspiration from the *two-click* experiments from `pyserini` [9], this allows researchers to easily reproduce and extend complex data processing pipelines that depend on the re-implemented APIs.

3 End-User Experience

Installing Orbit. Once the repository has been cloned, all of the re-implemented APIs can be run at once using Docker compose.

¹²<https://swagger.io/specification/>

```
from Bio import Entrez
from Bio.Entrez import _open, _build_request

db = "pubmed"
term = "headache[tiab]"

request = _build_request(
    "https://eutils.ncbi.nlm.nih.gov/entrez/.fcgi",
    {"db": db, "term": term})
entrez_record = Entrez.read(_open(request))

request = _build_request(
    "https://orbit.ncbi.nlm.nih.gov/entrez/.fcgi",
    {"db": db, "term": term})
orbit_record = Entrez.read(_open(request))

entrez_record["Count"], orbit_record["Count"]
```

Entrez API	Orbit API
102 573	97 010

Figure 3: Querying for all PubMed articles having to do with headache in the title or abstract using the BioPython library. The high-level helper functions have hard-coded URLs, but the methods listed above correspond to code in the helper functions. Our user-interface provides a live interactive comparison. The code can also be run in an online notebook.¹⁵

```
$ git clone git@github.com:health-nlp/orbit.git
[...]
$ docker-compose -f compose.all.yml up --build
[...]
```

We also provide alternative scripts for running individual APIs. To complete the entire acquisition, processing, and indexing pipelines for the two implemented APIs in this paper, we found it took approximately one day.

Use with Existing Systems. There already exist many libraries for dealing with biomedical literature APIs. We show how simply by changing the base URL in existing python libraries, Orbit can be used in existing pipelines.

Firstly, we highlight an adapted example of the API from the `biopython` tutorial on Entrez.¹³ Figure 3 illustrates the code and results of the API usage with the default Entrez URL and the Orbit URL. Next, we experiment with the `pytrials` library.¹⁴ Figure 4 illustrates the code and results of these examples using the default `ClinicalTrials.gov` URL and the Orbit URL.

User Interface. Since existing libraries may themselves come with their own restrictions, such as, the maximum 1000 trials returned by `pytrials`, our user interface allows users to directly compare the search results of Orbit with the original APIs via live queries. We demonstrate this feature in Figure 6. In this figure we compare Orbit to the `ClinicalTrials.gov` studies endpoint.

¹³https://biopython.org/docs/latest/Tutorial/chapter_entrez.html#pubmed-and-medicine

¹⁴<https://pytrials.readthedocs.io/>

¹⁵https://colab.research.google.com/drive/1-i6-hOlxujftA4K_4VtHh8IsTzWZB7oF

```

from pytrials.client import ClinicalTrials
ct = ClinicalTrials()
ct_records = ct.get_full_studies(
    search_expr="Coronavirus_OR_COVID")

ct._BASE_URL = "https://orbit.[...]/api/v2/"
orbit_records = ct.get_full_studies(
    search_expr="Coronavirus_OR_COVID")

len(ct_records), orbit_records["totalCount"]

```

ClinicalTrials.gov API	Orbit API
1000	13071

Figure 4: Querying for full studies related to Coronavirus or COVID. Note that the Python library pytrials does not support paging and caps the number of studies to be returned at 1000. The same footnote from Figure 3 can be used to run this code interactively.

```

$ curl -X GET \
  https://orbit.health-nlp.com/pubmed/ \
  /entrez/efetch/efetch.fcgi?term=cancer&
                                     retmode=trec&
                                     trecqid=1 \

1 Q0 41588292 0 1.0 orbit
1 Q0 41602872 1 0.9997 orbit
1 Q0 41602737 2 0.9994 orbit
1 Q0 41607415 3 0.9992 orbit
1 Q0 41555907 4 0.9989 orbit
1 Q0 41551008 5 0.9987 orbit
[...]
```

Figure 5: Returning a TREC run file from the Orbit Entrez API endpoint. The same is possible for the clinical trials endpoint.

The difference between the results returned by Orbit vs the original APIs (5.4% less for PubMed in Figure 3 and 19.2% more for ClinicalTrials.gov in Figure 6) arises due to query translation differences. Orbit does not perform the same automatic query expansion on terms, which accounts for lower recall in PubMed data. ClinicalTrials.gov performs a weighted match of limited data fields as opposed to an unweighted and all fields match by Orbit resulting a higher number of results returned.

Extending APIs. We demonstrate one extension we have made to the two APIs that benefit information retrieval researchers in Figure 5. Both PubMed and ClinicalTrials.gov are commonly used to obtain a set of documents that is re-ranked using more advanced ranking models. To better support this common use-case of the APIs, we add another format parameter to the search endpoints of our re-implementations to allow results to be returned in TREC run format.

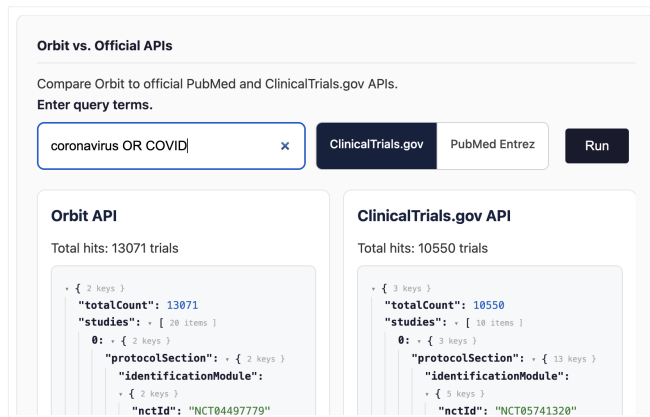


Figure 6: Comparison of Orbit results with the ClinicalTrials.gov API using the compare functionality.

4 Conclusion

We present Orbit, an open re-implementation of biomedical literature APIs, which offers a low barrier of entry to doing research with biomedical literature and opening new avenues for accessing biomedical literature. Our system enables anyone, whether bioinformatician or information retrieval researcher, to get started working with these large scale, complex document collections, where all the steps from acquiring the data to running the API have been pipelined to be run from a single docker command. Orbit enables anyone to run the same API services locally bypassing rate limitations and to extend API functionalities.

We have plans to extend the number of services beyond the two re-implementations of the PubMed Entrez API and the ClinicalTrials.gov API. Future work involves systematic testing of query translation and ranking functions of Orbit and the original APIs and extending our re-implemented collections to include the open access subset of PubMed Central, PMC, (i.e., PubMed articles with a license that permits full-text distribution; 11 756 316 documents at the time of writing),¹⁶ the open access subset of Europe PMC (approximately 7.6 million documents at the time of writing),¹⁷ and bioRxiv (a biomedical pre-print service; 435 078 documents at the time of writing).¹⁸ We intend to acquire, process, and index the full-text of these documents as part of their API re-implementations, significantly increasing the resources needed to host and distribute this data which was not feasible in the scope of this paper. A hosted version of Orbit is available at <https://orbit.health-nlp.com> and the source code is available at <https://github.com/health-nlp/orbit>.

Acknowledgments

Shrestha Ghosh is supported by the AI & Data Science Fellowship Program between the University of Tübingen and Boehringer Ingelheim

¹⁶<https://pmc.ncbi.nlm.nih.gov/>

¹⁷<https://europepmc.org/>

¹⁸<https://www.biorxiv.org/>

GenAI Usage Disclosure

The authors used Claude Sonnet 4.6 for writing and refactoring the code for building the website. Gemini Flash 3.5 was used for server-side code generation and refactoring and debugging. All code was reviewed and verified and the authors take full responsibility for the publication's content.

References

- [1] Pierre Achkar, Tim Gollub, Martin Potthast, Carsten Eickhoff, and Harrison Scells. 2026. UP2DATE: Shared Task on Systematic Review Updates.. In *SCOLIA*. 78–83. <https://ceur-ws.org/Vol-4187/shared-08.pdf>
- [2] Ziqi Chen, Bo Peng, Vassilis N Ioannidis, Mufei Li, George Karypis, and Xia Ning. 2022. A knowledge graph of clinical trials (CTKG). *Scientific reports* 12, 1 (2022), 4724.
- [3] Surabhi Datta, Kyeryoung Lee, Hunki Paek, Frank J Manion, Nneka Ofoegbu, Jingcheng Du, Ying Li, Liang-Chin Huang, Jingqi Wang, Bin Lin, et al. 2024. AutoCriteria: a generalizable clinical trial eligibility criteria extraction system powered by large language models. *Journal of the American Medical Informatics Association* 31, 2 (2024), 375–385.
- [4] Owen Dyer. 2025. PubMed is running on autopilot during shut-down, but key independent committee has been abolished. *BMJ* 391 (2025). arXiv:<https://www.bmj.com/content/391/bmj.r2158.full.pdf> doi:10.1136/bmj.r2158
- [5] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and René Spijker. 2017. CLEF 2017 Technologically Assisted Reviews in Empirical Medicine Overview. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017 (CEUR Workshop Proceedings, Vol. 1866)*, Linda Cappellato, Nicola Ferro, Lorraine Goeuriot, and Thomas Mandl (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-1866/invited_paper_12.pdf
- [6] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and René Spijker. 2018. CLEF 2018 Technologically Assisted Reviews in Empirical Medicine Overview. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018 (CEUR Workshop Proceedings, Vol. 2125)*, Linda Cappellato, Nicola Ferro, Jian-Yun Nie, and Laure Soulier (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-2125/invited_paper_6.pdf
- [7] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and René Spijker. 2019. CLEF 2019 Technology Assisted Reviews in Empirical Medicine Overview. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019 (CEUR Workshop Proceedings, Vol. 2380)*, Linda Cappellato, Nicola Ferro, David E. Losada, and Henning Müller (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-2380/paper_250.pdf
- [8] Jonathan Kans. 2010. Entrez® Direct: E-utilities on the Unix Command Line. *Entrez® Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US)* (2010).
- [9] Jimmy Lin. 2022. Building a Culture of Reproducibility in Academic Research. *CoRR* abs/2212.13534 (2022). arXiv:2212.13534 doi:10.48550/ARXIV.2212.13534
- [10] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations.. In *SIGIR*. 2356–2362. doi:10.1145/3404835.3463238
- [11] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval.. In *CIKM*. 4526–4533. doi:10.1145/3459637.3482013
- [12] Sebastián Ramírez. [n. d.]. *FastAPI*. <https://github.com/fastapi/fastapi>
- [13] Maciej Rybinski, Jerry Xu, and Sarvnaz Karimi. 2020. Clinical trial search: Using biomedical language understanding models for re-ranking. *Journal of Biomedical Informatics* 109 (2020), 103530.
- [14] Eric Sayers. 2010. A General Introduction to the E-utilities. *Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US)* (2010).
- [15] Harrison Scells and Martin Potthast. 2023. pybool_ir: A Toolkit for Domain-Specific Search Experiments.. In *SIGIR*. 3190–3194. doi:10.1145/3539618.3591819
- [16] Harrison Scells, Ferdinand Schlatt, and Martin Potthast. 2023. Smooth Operators for Effective Systematic Review Queries.. In *SIGIR*. 580–590. doi:10.1145/3539618.3591768
- [17] Zifeng Wang and Jimeng Sun. 2022. Trial2Vec: Zero-Shot Clinical Trial Document Similarity Search using Self-Supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). 6377–6390. doi:10.18653/v1/2022.findings-emnlp.476