Spacerini: Plug-and-play Search Engines with Pyserini and Hugging Face

Christopher Akiki¹,²  Odunayo Ogundepo³  Aleksandra Piktus⁴,⁵
Xinyu Zhang³  Akintunde Oladipo³  Jimmy Lin³  Martin Potthast¹,²

¹ Leipzig University  ² ScaDS.AI  ³ University of Waterloo  ⁴ Hugging Face  ⁵ Sapienza University

Abstract

We present Spacerini, a tool that integrates the Pyserini toolkit for reproducible information retrieval research with Hugging Face to enable the seamless construction and deployment of interactive search engines. Spacerini makes state-of-the-art sparse and dense retrieval models more accessible to non-IR practitioners while minimizing deployment effort. This is useful for NLP researchers who want to better understand and validate their research by performing qualitative analyses of training corpora, for IR researchers who want to demonstrate new retrieval models integrated into the growing Pyserini ecosystem, and for third parties reproducing the work of other researchers. Spacerini is open-source¹ and includes utilities for loading, preprocessing, indexing, and deploying search engines locally and remotely. We demonstrate a portfolio of 13 search engines created with Spacerini for different use cases.²

1 Introduction

The commoditization of data has transformed data-driven computer science in general (Hey et al., 2009) and machine learning (ML) and natural language processing (NLP) in particular (Mitchell et al., 2022). The race to train ever-larger language models depends so much on having access to immense amounts of text (Hoffmann et al., 2022) that the datasets have become, as Bender et al. (2021) claims, both technically (Bender et al., 2021) and methodologically (Jo and Gebru, 2020) “too big to document”. In practice, this often leads to an approach of training first and asking questions later (Akiki et al., 2022), which again is an example of a convenience experiment (Krohs, 2012), a research approach that depends on the availability of a resource and the ease of a method rather than its suitability to the problem at hand. In this sense, Beaulieu and Leonelli (2021) believe it is important to distinguish between the availability of data and its appropriateness, especially in light of the misconception that web data represents all human experience and is immune to the ever-widening digital divide (Leonelli, 2020). They suggest that the divide not only exists, but limits the representativeness of the web, which in turn reinforces the biases of the artifacts that use it (Bender et al., 2021).

Being unable to easily audit large datasets incentivizes researchers to release models trained on data they do not truly understand (Mitchell et al., 2022) leading to model behaviors that are hard to study, predict or trace (Mitchell et al., 2022; Siddiqui et al., 2022; Akyurek et al., 2022). This is especially problematic in light of the potential real-world harms that ensue (Weidinger et al., 2021; Hutchinson et al., 2020; Founta et al., 2018; Fast et al., 2016). Being able to properly understand the limitations of our datasets and qualitatively explore them in an ad-hoc fashion is a necessary first step toward understanding the behavior, harmful biases and failure modes of the artifacts that build upon them. Understanding the training data is therefore a critical step in the process of releasing and auditing large language models (Mökander et al., 2023).

It is from this vantage point that we initially developed Spacerini as an open-source tool for the quick indexing and deployment of shareable search engines, but have also since come to realize its potential in being useful for an even wider audience interested in making their text artifacts searchable. Indeed this includes IR students, Digital Humanists, Shared Tasks organizers, and digital investigative journalists, all of whom seek to quickly deploy ad hoc search engines for their research. We cover these use cases in more detail in Section 4.

Spacerini helps streamline the process of auditing large datasets by allowing users to effortlessly index their text collections and deploy them as interactive search applications that can be easily edited in the browser and shared with all stakeholders. It achieves that by “standing on the shoul-

¹ https://github.com/castorini/hf-spacerini
² https://hf.co/spacerini
Figure 1: Example of one of the many search apps (hf/spacerini/miracl-french) deployed as a Hugging Face Space. The Lucene BM25 index is hosted in the same repository as the frontend using git LFS and the frontend is based on a template which was automatically generated from one of the many Spacerini cookiecutter templates. The loading, preprocessing, indexing, and app deployment were made using an end-to-end workflow similar to the one showcased in Section 3.

As discussed in Section 1, in the face of increased scrutiny of the models trained on datasets in question, the topic of data understanding and governance has been gaining more traction, being accepted as an important part of research. Efforts such as those of Mitchell et al. (2022) contribute frameworks for more standardised and reproducible metrics and measurements of datasets, and we position ourselves as a complementary continuation of their work, focusing on a more curatorial and qualitative assessment that might not readily fit under the umbrella of “measurements”. We therefore aim to fill the gap in the evaluation landscape by facilitating qualitative, rather than quantitative analysis of large scale data collections.

Similarly to the authors of Gradio (Abid et al., 2019), a Python package for fast development of Machine Learning demos, we believe that the accessibility of data and model analysis tools is crucial to building both the understanding of and the trust in the underlying resources. The potential of relevance-based interfaces to massive textual corpora, the creation of which can be facilitated by leveraging toolkits such as Pyserini (Lin et al., 2021), has previously been tapped into by the researchers at the Allen Institute of AI who propose...
a C4 (Raffel et al., 2022) search engine. Similar interfaces have also been found useful in more specialised domains, e.g. in COVID-related datasets (Zhang et al., 2020), news quotes (Vuković et al., 2022), or medical literature (Niezni et al., 2022). However, while these solutions are undeniably useful, they remain very contextual: dataset-specific, and project-specific. We believe Spacerini to be the first generalizable tool which proposes an end-to-end pipeline automating the route from raw text to qualitative analysis.

3 Spacerini

Spacerini is a modular framework that integrates Pyserini with the Hugging Face ecosystem to streamline the process of going from any Hugging Face text dataset (or indeed any text dataset)—either local or hosted on the Hugging Face Hub—to a search interface driven by a Pyserini index that can be deployed for free on the Hugging Face Hub. In what follows, we deconstruct an example script to showcase the different features enabled by Spacerini. When run end-to-end, the script pulls a dataset from Hugging Face, pre-processes it, indexes it, creates a gradio-based search interface and deploys that as a Hugging Face Spaces demo. This is only meant as a feature-complete demo, and we don’t expect most people to want to integrate every step into their workflows, but rather to cherry-pick and decide what best to use depending on context.

3.1 Loading Data

All our workflows are backed by the Hugging Face datasets library (Lhoest et al., 2021), itself based on the extremely efficient Apache Arrow format. Datasets is a mature library which provides a standardized interface to any tabular dataset, in particular, to tens of thousands of community datasets hosted on the Hugging Face Hub. The datasets library gives fine-grained control over the lifecycle of tabular datasets, which we choose to abstract away through a set of opinionated data loading functions that cover the use cases we deem relevant to information retrieval. We also add new functionality, such as the ability to load any document dataset from the ir_datasets library using for example the following one-liner to load MS MARCO (Nguyen et al., 2016) as a Hugging Face datasets.Dataset object using a function from the data subpackage:

```python
from spacerini.data import load_ir_dataset
hf_dset = load_ir_dataset("msmarco-passage")
```

We include wrappers to load database tables, pandas DataFrames (pandas development team, 2020; Wes McKinney, 2010), and text datasets on disk, as well as the ability to load any dataset either in memory-mapped mode or in streaming mode: the former makes it possible to handle larger-than-memory datasets, and the latter larger-than-disk datasets that can be streamed from a remote location such as the Hugging Face Hub.

3.2 Pre-processing

Spacerini also provides a preprocess subpackage which offers a range of customizable pre-processing options for preparing datasets. This module includes a sharding utility that enables the partitioning of large datasets into smaller, more manageable chunks for efficient parallel processing.

```python
from spacerini.preprocess import shard_dataset
shard_dataset(
    hf_dataset=hf_dset,
    shard_size="1GB",
    column_to_index="text",
    shards_paths="msmarco-shards",
)
```

3.3 Indexing

Spacerini’s index subpackage leverages Pyserini to provide very efficient Lucene indexing and allow users to easily and quickly index large datasets, either sharded in the pre-processing step, or any text format accepted by Pyserini, and streaming text datasets, such as those returned by Spacerini’s data subpackage. This subpackage also exposes several tokenization options using existing language-specific analyzers as well as Hugging Face subword tokenizers (MOI et al., 2022).

```python
from spacerini.index import index_json_shards
index_json_shards(
    shards_path="msmarco-shards",
    index_path="app/index",
)
```

---

1https://c4-search.apps.allenai.org/
3https://hf.co/datasets
4https://lucene.apache.org/core/9_5_0/analysis/common/
We also provide wrappers to Pyserini’s dense and hybrid retrieval functionality through the spacerini.index.encode subpackage.

3.4 Template-based Search Interfaces
Having indexed a collection, one can easily spin up a frontend using the frontend subpackage and one of many provided templates\(^7\). These are built using cookiecutter,\(^8\) a Python templating library for software projects. We provide a few batteries-included frontend templates based both on the Gradio\(^9\) and Streamlit\(^10\) demo app frameworks, both of which are natively supported by Hugging Face Spaces. Figure 1 showcases a search engine built using one of our Streamlit templates.

```python
from spacerini.frontend import create_app
cookiecutter_vars = {
    "dataset_text_field": "text",
    "metadata_field": "docid",
    "space_title": "MS MARCO Search",
    "local_app": "app"
}
create_app(templates=gradio-vanilla, extra_context_dict=cookiecutter_vars, output_dir='.')
```

3.5 Deployment to Hugging Face Spaces
The local apps developed in the previous subsection can then be pushed to Hugging Face Spaces and hosted there for free. One can then further customize the running app from the browser, for example to add functionality not provided by the chosen template. The goal of the templates is to provide a useful starting point in the form of a running app that users can further customize with interface features useful for their own workflows.

```python
from spacerini.frontend import create_space_from_local
create_space_from_local(
    space_slug="msmarco-passage-search",
    organization="spacerini",
    space_sdk="gradio",
    local_dir=LOCAL_APP,
    delete_after_push=True,
)
```

3.6 Sharing Indexes as Hugging Face Datasets
Orthogonal to the workflow presented so far, is the ability to upload Lucene indexes to the Hugging Face Hub using shareable dataset repositories and enabling reproducible retrieval experiments.

```python
from spacerini.index import push_index_to_hub
push_index_to_hub(
    dataset_slug="msmarco-english-analyzer-mmspace",
    index_path="index",
)
```

Any hosted index can then just as easily be downloaded for local use:

```python
from spacerini.index import load_index_from_hub
index_path = load_index_from_hub("msmarco-fr-analyzer")
```

3.7 Search and Pagination
Search features are provided by the search subpackage and leverage the memory-mapping feature of Arrow tables to load the entire table of results—no matter how big—only materializing the specific shard that corresponds to the requested result page.

```python
from spacerini.search import result_indices, result_page
ix = result_indices("
Lorem Ipsum",
num_results=1_000,
INDEX_PATH,
)
last_results_page = result_page("
hf_dset",
i,
page=-1,
results_per_page=20,
).to_pandas()
```

4 Use Cases and Demonstrations
We envision Spacerini to be useful primarily to NLP researchers, students, shared task organizers, data scientists, and data annotators, as well as tech-adjacent and -proficient professionals and laypeople. In what follows, we overview a series of 13 use-cases that we implemented and how they might benefit their respective targeted audience. An overview of all 13 search engines can be found

---

\(^7\) https://github.com/castorini/hf-spacerini/tree/main/templates
\(^8\) https://github.com/cookiecutter/cookiecutter
\(^9\) https://gradio.app/
\(^10\) https://streamlit.io/
at https://huggingface.co/spacerini; in following inline links to the selected engines, this part of their URL prefix is shortened to ‘hf’ for brevity.

4.1 NLP researchers
Spacerini is designed to enable qualitative analysis of large-scale textual corpora without the need for extensive engineering work. It can be used in dataset auditing campaigns, such as those carried out by Kreutzer et al. (2022) or in data annotation efforts. Also relevant here are generative text models whose outputs can be better understood by better understanding the datasets they were trained on. We refer the reader to Section 5 of Piktus et al. (2023) for a detailed exposé of potential use scenarios which include: PII detection, problematic content detection, social representation, benchmark and language contamination detection, as well as plagiarism and memorization detection. An example demo for this context is the index of the XSUM (Narayan et al., 2018) dataset which is indexed and can be explored with the hf/xsum-search demo.

4.2 IR researchers
Given its tight integration with Pyserini, Spacerini can also be leveraged by IR researchers to experiment with modifications of their retrieval pipelines in user studies or to deploy demos of their working prototypes. Reproducibility for IR experiments is further enhanced thanks to the index sharing abilities showcased in Section 3.6. As a practical example, Spacerini was used in the context of the BigCode project (Li et al., 2023) to quickly experiment with multiple n-gram tokenization schemes for BM25-based code retrieval; this corresponds to the hf/spacerini/code-search demo.

4.3 Linguists
Corpus linguistics relies on qualitative analyses of text corpora to understand language and its many varieties by studying the way it is used (McEnery and Hardie, 2011; Piktus et al., 2023). Some of these empirical analyses can be facilitated by the usage of an inverted index, coupled with the correct querying patterns and frontend elements, both of which are easy to achieve using Spacerini.

4.4 Digital Humanists
Spacerini can also be leveraged by Digital and Computational Humanists, Archivists and Librarians looking to index their collections. Indeed, GLAM (Galleries, libraries, archives, and museums) collections are increasingly being made available as datasets. Furthermore, there is a growing interest in the digital humanities in training and using languages models, as demonstrated by the success of projects such as the AI for Humanists Project. In this context, indexing data relevant to these efforts is a difficult task; often project-based and contingent upon precarious funding arrangements. Having a project-agnostic tool like Spacerini could prove valuable to this community and a useful addition to toolkits such as the GLAM Workbench (Sherratt, 2021).

4.5 IR Students
Given its low barrier of entry, Spacerini can be a good tool for IR courses, where students could be tasked with developing search engines, by providing an easy-to-deploy frontend interface for their developed retrieval systems which does not even have to be deployed within the same application, as demonstrated by the hf/chat-noir, a frontend wrapper for ChatNoir (Bevendorff et al., 2018).

4.6 Shared task organizers
Spacerini can also be leveraged by organizers of shared tasks such as MIRACL (Zhang et al., 2022) and Touché (Bondarenko et al., 2022), who want to help participants explore the datasets without forcing them to download large volumes of data or giving participants full access to the data: it is indeed possible to host the index privately on the Hugging Face Hub and only expose access to it through a search interface. Spacerini can also be used as a platform for participants to deploy working prototypes of their submissions with a unified interface provided by the organizer as a cookiecutter template. Example demos for this use case include hf/miracl-bengali, hf/miracl-arabic, and hf/miracl-swahili.

4.7 Tech journalists
Spacerini can help data journalists and digital investigative journalists index, explore, and understand open data, in a similar vein to the functionality provided by the Aleph suite. Providing technical tools to data journalists is a crucial in uncovering matters of public interest, as was evident by role played by the collaborative use of the Neo4j graph

11https://www.bertforhumanists.org/
12https://docs.alephdata.org/
database in unraveling corrupt networks surrounding tax havens (Díaz-Struck and Cabra, 2018).

4.8 Additional usage patterns

Finally, three features of Hugging Face Spaces make them especially attractive for users: (1) they can leverage private datasets, meaning that one can provide search access to a dataset without sharing the underlying data, (2) they can be seamlessly embedded into HTML, specifically Gradio-based Spaces which can be embedded as Web Components\(^\text{13}\) so that users can easily integrate a Spacerini-based search feature into their own websites\(^\text{14}\), and (3) Gradio-based Spaces expose a FastAPI\(^\text{15}\) endpoint that can be queried to access the functionality of the space, making deployed search engines accessible through HTTP calls.

5 Limitations and Future Plans

The main limitation of the off-the-shelf variant of Spacerini is the disk space limit imposed by Hugging Face Spaces, which is currently set to 50 GB for the free tier.\(^\text{16}\) While not enough to accommodate entire corpora such as ROOTS or The Pile, such datasets are typically amalgamations of constituent datasets which can each be studied independently. This limit has no bearing on Spacerini search apps deployed locally. Should users still want to get more disk space for their Spaces-hosted indexes, they can either pay for an upgrade to a more appropriate tier or see whether they qualify for a free hardware upgrade through the community grants offered by Hugging Face, in the Settings pane of the relevant space.

Planned improvements include automating the creation of dataset cards (or rather “index cards”) when pushing an index to the Hugging Face Hub, better documentation, as well as more fine-grained tokenization support.

Please also note that Spacerini is currently in active development and that the stability of its current API and subpackages isn’t guaranteed not to involve breaking changes as we converge toward the first stable version release. We also look forward to community contributions both to the codebase and to the frontend templates, as well as in the form of actual use cases of the library that culminate in the deployment of search apps.

6 Conclusion

We presented Spacerini, a modular framework that enables the quick and free deployment and serving of template-based search indexes as interactive applications for ad-hoc exploration of text datasets. The need for such a tool is especially pressing as large language models have come to consume inordinate amounts of text data, reinforcing the need for a qualitative exploration and understanding of datasets to assess them in a way that is impenetrable to quantitative analyses alone.

Spacerini leverages features from both the Pyserini toolkit and the Hugging Face ecosystem to facilitate the creation and hosting of user-friendly search systems for text datasets. Users can easily index their collections and deploy them as ad-hoc search interfaces, making the retrieval of relevant data points a quick and efficient process. The user-friendly interface enables non-technical users to effectively search massive datasets, making Spacerini a valuable tool for anyone looking to audit their text collections qualitatively. The framework is open-source and available on GitHub under gh/castorini/hf-spacerini and demo search apps can be found under hf/spacerini.

The key advantage of Spacerini is its ability to simplify the search process, allowing researchers to conduct quick and efficient audits, while abstracting away all the minutiae of indexing data or hosting services. We believe that this provides an opportunity for collaboration and transparency in IR and NLP research. With the creation and sharing of search indexes publicly, practitioners, researchers and the general public can work together to pinpoint problematic content, find duplicates, and identify biases in datasets.

Finally, we emphasize that Spacerini is a first step in the direction of systematic dataset auditing, and more work is still needed to create standardized structures that leverage tools such as ours to properly document the different axes of interest.

Acknowledgments

We are grateful to Lukas Gienapp, Daniel van Strien, Yuvaraj Sharma, Omar Sanseviero, Julien Chaumond, Lucain Pouget, Abubakar Abid, and Leandro von Werra for their tireless support and invaluable advice throughout this project.
References


Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7), Cardiff, United Kingdom. Leibniz-Institut für Deutsche Sprache.


Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a miracl: Multilingual information retrieval across a continuum of languages.