Site Search Engines

A Comparative Survey

Aumüller Thomas, Liegl Daniel, Platzer Fabian

Site Search Engines

- Site Search allows a user to search a website's content.
- Made possible by a Site Search Engine (SSE) in backend, which processes query and returns results.

- Two main setups:
  - **Self-Hosted:** Engine hosted on own infrastructure.
  - **Cloud-Based:** API calls to engine hosted by someone else.

Source: https://www.algolia.com/blog/product/what-is-site-search/

Self-Hosted SSE

Pros:

- Lower price or open-source
- More control

Cons:

- Own server
- Setup
- Maintenance

Cloud-Based SSE

Pros:

- No hardware
- Fewer limitations
  - (e.g. computational power)
- Less maintenance
- Hidden Complexity

Cons:

- Lack of control
- Data privacy
- Network bandwidth limits

Survey Structure

- Self-hosted SSE setup.
- Indexing.
- Dataset.

- Toolchain used.
- JavaScript integration.
- Live demo.

- Criteria.
- Assessment.

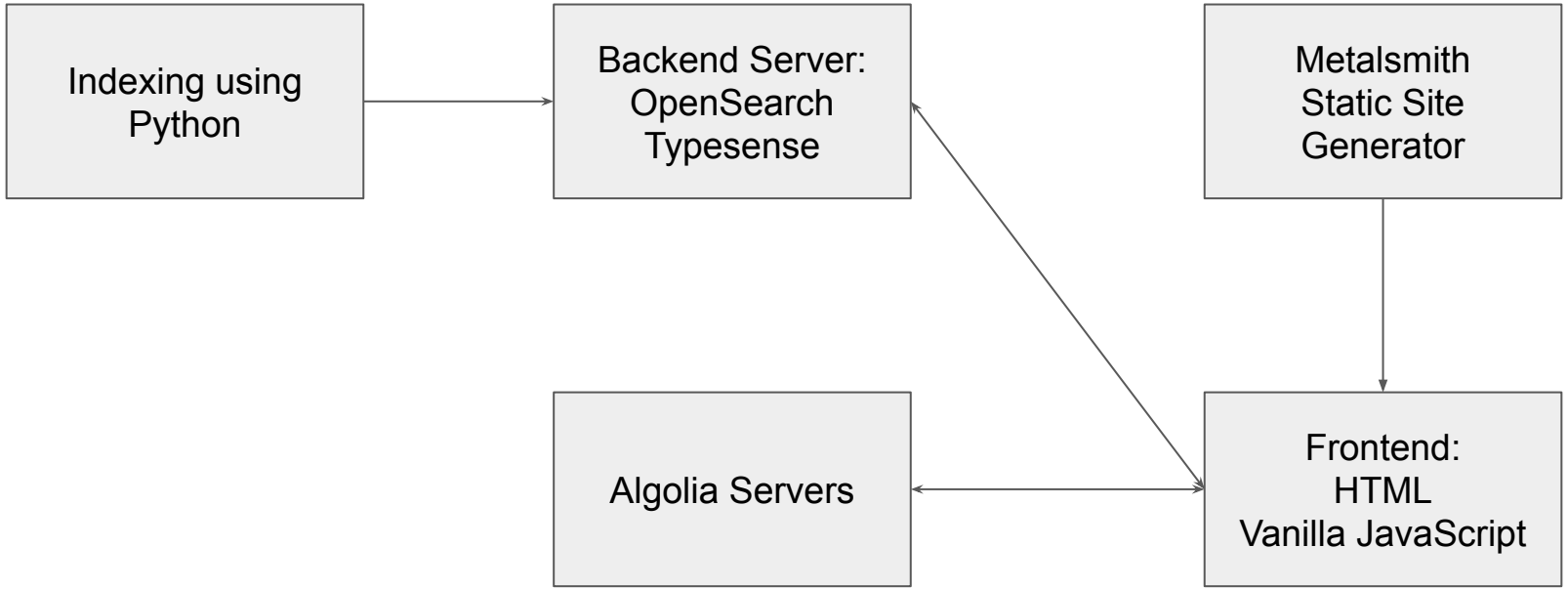
Evaluated Three SSEs

- **OpenSearch** (Self-Hosted)
  - Secure search and analytics.
  - Machine learning support (k-NN search)

- **Typesense** (Self-Hosted)
  - Easy setup.
  - Lightweight yet powerful & scalable alternative.
  - Clean well-documented API.

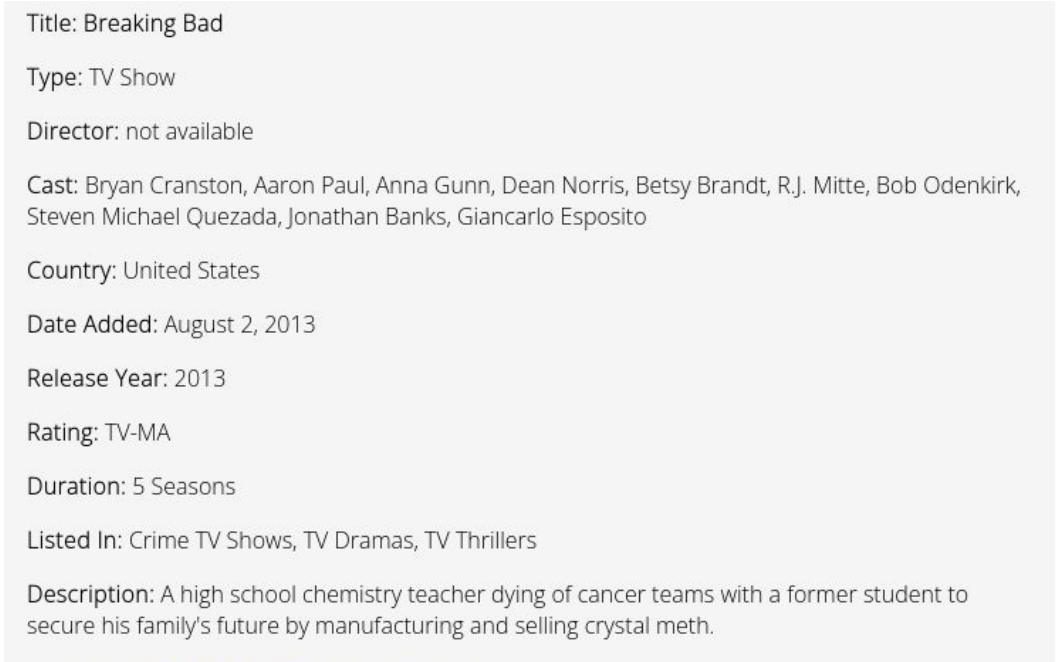
- **Algolia** (Cloud-Hosted)
  - Web interface for managing.
  - Easy to implement.
  - API Monitoring.

Visual Breakdown of the Toolchain



Dataset

- Listings of movies and TV shows on Netflix
- Details such as - cast, directors, ratings, release year, duration, etc.
- 8807 entries.
- netflix_titles.csv (3.4 MB)



Source: https://www.kaggle.com/datasets/shivamb/netflix-shows

Self-Hosted Backend Setup

- Easy setup using Docker.
- Docker compose to build infrastructure.
- No major problems with setup for Typesense and OpenSearch:
  - Docker compose templates available.
  - Did not manipulate the templates; See what works with setup recommended by developers.
- OpenSearch needed some more attention, due to CORS.

Indexing Typesense

- Used the ‘typesense’ Python library.
- Easily created the index, with our data schema.
- Imported the CSV with the Netflix data.
- Converted it to the JSONLines format.
- Passed the whole file into the import_() function with few problems.

Indexing OpenSearch

- Created initial index using OpenSearch dashboard.
- Used ‘opensearch-py’ Python library.
- OpenSearch needs two JSON objects for each document to be indexed.
- Had to manually build a really large JSON string.

_bulk request example:

```
{"index" : { "_index" : "netflix", "_id" : "5940" } }
{
        "title":"Breaking Bad",
        "description":"A high school chemistry
        teacher dying of cancer teams with a former
        student to secure his family's future by
        manufacturing and selling crystal meth.",
        "cast": "Bryan Cranston, Aaron Paul, Anna
        Gunn, Dean Norris, Betsy Brandt, R.J. Mitte,
        Bob Odenkirk, Steven Michael Quezada,
        Jonathan Banks, Giancarlo Esposito",
        "listed_in":"Crime TV Shows, TV Dramas, TV
        Thrillers"
}
```

Indexing Algolia

- Used the web interface of Algolia.
- Supports records as JSON, CSV, and TSV.
- JSON files exceeded limit of free trial’s API calls.
  - Used CSV format.
- Easy to index, manage and add records to indices via web interface.

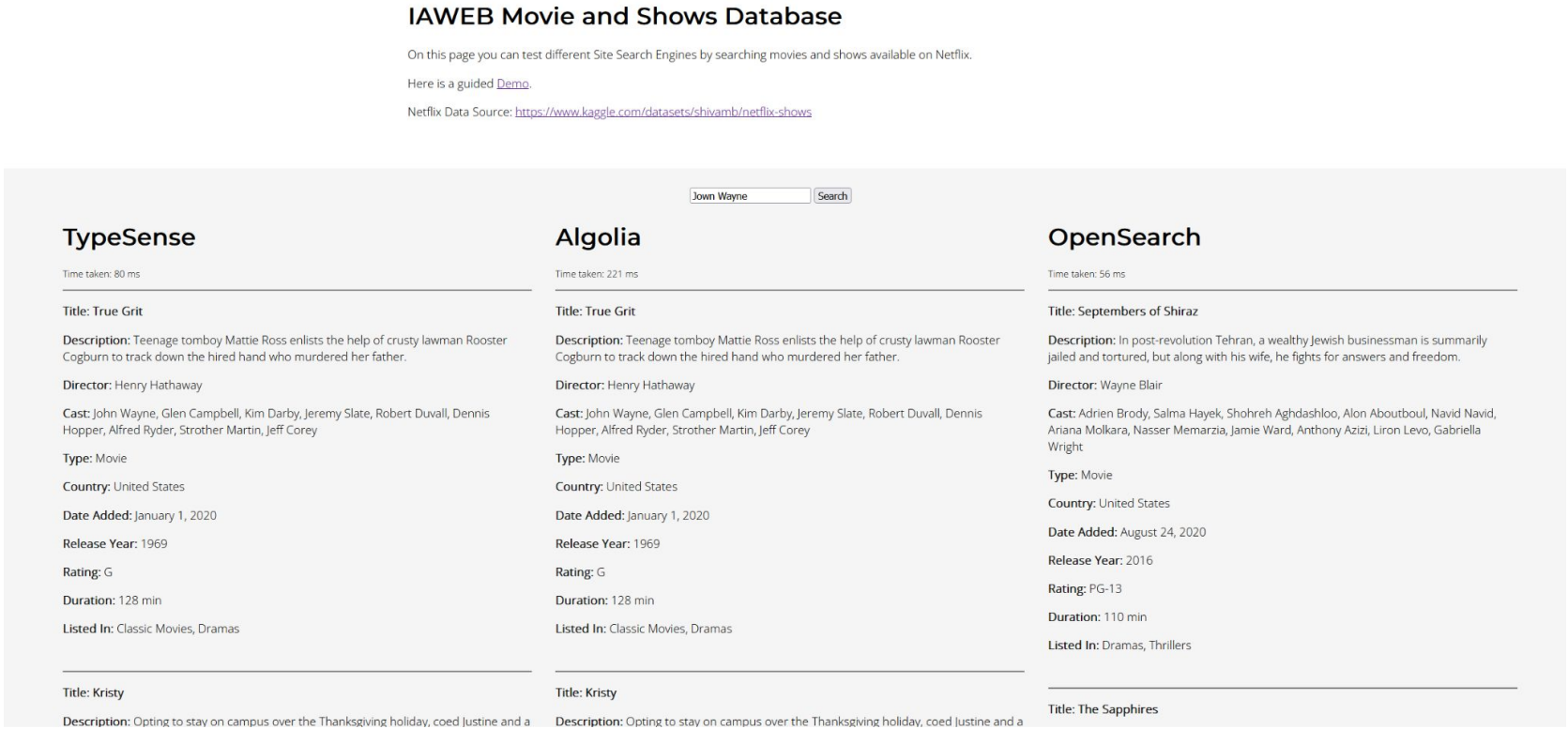
Frontend Toolchain

- Running on an Apache 2.0 web server
- Built using the Metalsmith static site generator.
  - Used the Barebones Starter by Werner Glinka to get started.
- Nunjucks templating engine.
- HTML and Vanilla Javascript.

Frontend Integration of the SSEs

- Single search bar searches all 3 SSEs at the same time.
- Algolia and Typesense have Vanilla JavaScript API clients.
  - Easy import using <script> tags.
  - Queried the backend using the respective request function.
- Opensearch only has a Node.js library.
  - No Vanilla JavaScript support.
  - Manual fetch() querying required.

Showcase



Showcase Videos: https://youtube.com/playlist?list=PLsp4BtuSXH9nkw7SUff_0SsnASeRGWmM1&si=IneqAidjlpZhedUH

Comparison

	Typesense	OpenSearch	Algolia
Paid	No	No	Yes
Faceted Search	Yes (parameter)	Yes (plugin)	Yes
Advanced Search	Yes	Yes	Yes
Query Suggestion	Yes	Yes	Yes
Fuzzy Search	Yes	Yes (in query)	Yes
Taxonomies	Yes (plugin)	Yes	Yes
Dictionaries	No	Yes	Yes
Out of the Box Security	Search only and admin API keys	Manual setup	Search only and admin API keys
Personalized Results	Yes (plugin)	Yes (plugin)	Yes (premium)

A more detailed comparison can be found here: https://docs.google.com/spreadsheets/d/1G1I-4-Oi-zswkyT_IvtEECdnMQlUqo92hm6VAxsr2-k/edit#gid=0.

Conclusion

typesense

Our preferred Site Search Engine: Typesense

- Open Source.
- Very active Community.
- Easy to setup and use.
- Support for many programming languages and frameworks.
- Good developer experience.