

Apache Lucene is an open-source software library written in Java. Lucene works well with other popular programming languages as well such as C++ and Python. At a high level, Apache Lucene is extremely helpful for applications that require a full-text search. Users can index documents at any length and search for text within those documents. Lucene will return a ranked list of those documents (similar to how the Google search engine works). Lucene is being used by large corporations such as Comcast and IBM.

Some of the features of Lucene include scalable indexing. Lucene can index 150GB/hour with modern hardware and minimal RAM requirements (from <https://www.happiestminds.com/insights/apache-lucene/>). Lucene offers users the best search results based on priority. Users can utilize the spell-checking proximity function as well as the multi-lingual search capability feature. Lucene provides users with results that process the entire text corpus and uses relevancy sorting (such as sorting by date or by a given field). Since Lucene is open source, it can be used cross-platform, supporting both commercial and other open-source applications.

Now that we have a high-level understanding of Apache Lucene, I want to talk about Apache Solr, an application that is built with Apache Lucene. Apache Solr, like Lucene, is built in Java and is open source. Solr is primarily used as an information retrieval library but can also be used as a NoSQL database. Because of this, it shares a lot of the same terminologies with NoSQL databases. Below is a list of some of the common terminologies.

- Document: Basic unit of information
- Field: Key-value pair in document
- Collection: Group of cores that form an index

- Node: JVM instance running Solr

Solr works with all kinds of data: structured, semi-structured, and unstructured. Solr is so optimal that data gets stored and indexed and is available for search in real-time. It does all this by following the master-slave architecture. On top of all these performance benefits, Solr is also stable and fault-tolerant. It also provides built-in security such as SSL encryption between Solr clients and Solr nodes.

In terms of functionality, Solr has a lot to offer. Users can search using Boolean queries, phrase queries, fuzzy queries, wildcards, auto-complete, and spell-check just to name a few (from <https://sematext.com/guides/solr/>). Solr also allows you to manage administrative tasks such as logging, adding, updating, deleting, and searching for documents all in a provided built-in user interface. Lastly, Solr is simple to use across multiple programming languages including Java, C++, and Python. On top of this, users can search using RestFul XML/HTTP, JSON APIs, and client libraries to perform searches. Once again, since Solr is open-sourced, it can be used across commercial and open-source software with its license.

With all of the benefits listed above, it is easy to see why many companies will opt for Solr. Many applications nowadays require powerful search capabilities and Solr provides a lot of the requirements such as processing and searching in real-time at a massive scale, fault tolerance to provide stability for their clients, and built-in security in the event of handling sensitive data. Some of the most recognizable companies that use Solr are Apple, NASA, Goldman Sachs, and Boeing.

Despite all of these benefits, there is one major drawback to Solr. Since Solr is open source, there is no formal support if a company decides to use this software. Also, there are no formal trainings for Solr however many platforms, such as Udemy, offer courses on Solr.

## References

<https://sematext.com/guides/solr/>

<https://www.happiestminds.com/insights/apache-lucene/>

[https://solr.apache.org/guide/7\\_2/a-quick-overview.html](https://solr.apache.org/guide/7_2/a-quick-overview.html)

<https://lucidworks.com/post/who-uses-lucenesolr/>

<https://lucene.apache.org/>