

Algolia: Overview on Search and Recommendation

Introduction

Search has become a task that people do in daily life. People search for contacts when they need to call someone, search for past messages when they need to check the messages they have received or sent, search for an address, search for items when they shop online, and many other tasks need to be done by search. This makes search engines play an important role in almost all applications. However, building a good search functionality is not that simple and easy, especially with large data sets. There are a lot of aspects that need to be handled when building search such as tokenizing, indexing, ranking, and feedback to improve search. This paper explores an overview of Algolia, a search-as-a-service which provides a set of tools that simplify the process of making and integrating a full search experience into websites and applications.

Algolia Search

Algolia search provides many built-in features that are available out-of-the-box and easy to integrate into applications. Some of the features are global language support, typo-tolerance, and synonyms. It supports not only alphabetical languages, but also symbol-based languages like Chinese, Japanese, Hebrew, and Arabic. Typo-tolerance helps users to still find search results they are looking for even if they make mistakes while typing the query. For example, a user types “crpet” when it should be “carpet”. It also handles a few types of synonyms, one of them is regular synonyms like “pants” and “trousers”.

Algolia also supports common text retrieval features like removing stop words and performing lemmatization. Not only that, it also supports compounding for German, Dutch, Finnish, Danish, Swedish, and Norwegian Bokmål languages. Compounding is a process of cutting a compound word into its meaningful parts. For example, in German language, the compound word *Hundehütte* (dog house) is split into *Hunde* + *hütte*. Behind the scene, the compounding process includes the following steps: find a large text data for the target language, do lexical analysis (part-of-speech tagging), trim the lexicon by keeping only words that could not be compounded by other words with respect to a given score, and enrich the lexicon.

Ranking Search Results

Showing relevant search results is very important in search engines. In Algolia, relevance is defined by finding records that match a query, then ranking the records based on some criteria. Ranking is essentially sorting the results. There are two types of ranking supported in Algolia, which can be selected based on needs. First, exhaustive sorting, which sorts results strictly based on the sorting attributes. In exhaustive sorting, relevance is not the main key. The second type is relevant sorting. The two types of ranking formula depend on the index replica types. These replicas are used when there's a need to use different rankings for the same data. Each index has a unique sorting strategy and can't be changed at query time. This is because Algolia does pre-sorting at indexing time, which leads to performance boost.

Algolia has its own approach in ranking search results. It uses a tie-breaking algorithm that is applied to 8 different ranking criteria. These criteria are, in order: number of typos, geolocation, number of words in the query matching in the result, filters, distance between words, best matching attribute in the record, number of words matching exactly (without typo), and custom ranking. Algolia recommends to not change the order of the criteria since it works for the majority of use cases.

The tie-breaking algorithm is applied after matching records are found. The way it works is by ordering the results based on each criteria, in order. If there are records that are the same in a criteria (tie), it will reorder the records with the next criteria (break the tie). The last criteria, custom ranking, is based on setting custom ranking attributes. These attributes usually are based on metrics that are appropriate to the business goal, such as popularity, number of sales, views, likes, release dates, etc.

There are certain advantages of using tie-breaking algorithm in ranking results. First, ranking is clear and easy to understand. It is defined by the 8 criteria. Second, it's easy to modify the criteria order and to add custom ranking.

Implementation Process

There are 4 big steps to integrate Algolia:

1. Indexing data

Data needs to be sent to the Algolia server to be indexed. Developer needs to fetch the data from its original source (database, XML files, spreadsheets, or anything) and transform them into a JSON format. This can be done by writing a script that is run locally or on a server. Then send the transformed data to Algolia server via the provided

API. The records in the JSON file should contain information that is useful for the purpose of search i.e the text that the users will search on, filter attributes, and ranking attributes. It is discouraged to include unnecessary attributes in the record as it will consume more space and make noise.

2. Configuring relevance

This step includes selecting searchable attributes and custom ranking attributes. Searchable attributes are attributes that will be used for searching. Some examples of useful searchable attributes are name, description, color, brand, tags, etc. Custom ranking attributes are usually related to the business metrics like number of sales, views, likes, ratings, etc. This can be configured right from the Algolia dashboard.

3. Building the search user interface

Algolia offers a UI/UX library called InstantSearch that developers can use on web, mobile, or voice applications. It is also customizable. This way, developers don't need to implement the search UI from scratch, thus reducing development time.

4. Iterating using Algolia Analytics

Building an application is not a once-and-done event. It is very important to analyze and iterate to improve the implementation. Algolia also provides out-of-the-box analytics and Click and Conversion Analytics. By using these, developers can analyze data to make decisions to improve on many aspects of the search functionality, from how to format and structure the data, configurations to index, layout, etc.

Algolia Recommend

Algolia Recommend is a pretty new feature introduced in 2021. It offers two recommendation models, Frequently Bought Together and Related Products. Under the hood, it relies on supervised machine learning models. The data from the past 30 days are collected and transformed into a matrix of `userToken` and `objectID`. Then, it applies a collaborative filtering algorithm.

Conclusion

Algolia as a search-as-a-service provides many out-of-the-box features that are essentials to building search functionality. Although it still requires some knowledge about search engine features, developers will spend less time on development because Algolia simplifies the process of indexing, relevance tuning, building search UI, and analytics. Business teams can also play a

part in tweaking the ranking rules since it is available from the dashboard. Teams can focus more on fine tuning the relevance to improve the search functionality as a whole.

References

<https://www.algolia.com/doc/guides/getting-started/how-algolia-works/in-depth/features/>
<https://www.algolia.com/blog/engineering/increase-decompounding-accuracy-by-generating-a-language-specific-lexicon/>
<https://www.algolia.com/doc/guides/managing-results/refine-results/sorting/in-depth/replicas/>
<https://www.algolia.com/doc/guides/managing-results/must-do/custom-ranking/>
<https://www.algolia.com/doc/guides/sending-and-managing-data/prepare-your-data/in-depth/prepare-data-in-depth/>
<https://www.algolia.com/doc/guides/managing-results/must-do/searchable-attributes/>