

Documentation

Seqwa AI-Powered Search

Nikhil Kalyankar

Kalyankar Analytics Private Limited

3/28/22

Page intentionally left blank

Table of Contents

- Introduction 4
- About 5
- Features 6
 - Full-Text Search 6
 - Semantic Ranking..... 6
 - Semantic Search 6
 - Entity Extraction..... 7
 - Summarization..... 7
 - Clustering..... 7
- Getting Started 8
 - Data Preparation..... 8
 - Create An Index 9
 - Generate API Keys..... 9
 - Working with No-Code Search UI 10
- Developer API 12
 - Credentials 12
 - Autocomplete 13
 - Request URL..... 13
 - Method 13
 - Headers..... 13
 - Params 13
 - Request URL Structure 14
 - Response 14
 - Response JSON Structure 14
 - Search..... 16
 - Request URL..... 16
 - Method 16

Headers.....	16
Request URL Structure	17
Response	17
Response JSON Structure	17
Recommend	19
Request URL.....	19
Method	19
Headers.....	19
Params	19
Request URL Structure	20
Response	20
Response JSON Structure	20
Index.....	21
View API Usage.....	21
Add Records	21
(Re)generate API Keys	21
Search UI.....	21
Update Information.....	22
View Indexing Logs.....	22
Delete Index.....	22
Account	23
View API Usage.....	23
Add API Credits and Storage	23
View Purchase History	23
Redeem Code	23
Delete Account	24

Introduction

Businesses lose 80% of online sales due to sub-standard search functionality. Customers want speed, accuracy, and relevance. They expect to get to the right product within five to ten keystrokes.

Seqwa is a low-cost, easy-to-use hosted search platform for solving your critical search requirements. It supports three uniquely beneficial functions:

Autocomplete (query suggestions and top full-text search results), Search (Full-text Search with semantic ranking for intent), and Semantic Search (Intent-driven search). Work with one or all the functions based on your needs.

Set up a functional search utility in minutes: upload your data, generate API keys, publish a demo user interface, and search! Enjoy a seamless search experience supported by a highly available and scalable cloud infrastructure.

www.seqwa.com

About

We are Seqwa, and we believe in the power of semantic search.

Semantic search is the future of online search—it allows for a more advanced understanding of queries and the ability to anticipate needs. But implementing semantic search is expensive, and it isn't cost-effective for many companies. We want to change that. Seqwa is an AI-powered, cloud-based search solution accessible to all businesses, big and small. Our goal is to make semantic search available to everyone because we believe that anyone should implement a powerful tool without breaking the bank.

Seqwa allows customers to get the power of scalability while only paying for what they need. We make our platform simple, so you can focus on your business goals instead of getting bogged down by technical details. At Seqwa, we believe that an advanced search is essential for any business. Instead of typing in the exact words you want to find, a semantic search takes away the guesswork and allows for a more natural way of finding what you need.

Features

Full-Text Search

When looking for a word or topic, you can quickly use a sequential search to find it within a single document. However, it would take hours to search it in a large corpus.

Full-text search engines speed up searching by breaking this process into two phases: indexing and searching. Indexing involves reading, processing, and listing search terms. Search engines can now look up the index and then rank the documents. There are many ways of ranking search results, including search term frequency, term distance, or a pre-defined custom formula.

Full-text search is a common and robust approach. The user is presumed to be looking for a specific set of documents.

In edge cases, the user may have to go through several searches and documents. These edge cases get amplified on websites providing e-commerce and similar services. Users are likely to drop off within a few unsuccessful queries resulting in lost sales.

Semantic Ranking

Ranking search results semantically helps better serve users. This approach ranks the results based on the user's intent. You can understand user intent from the search query, previous search queries, user provided-context, user preferences, or demographic profile.

For instance, when two users search for the term: ball, they will get different top results if one is a soccer fan and the other is a tennis fan.

Semantic Search

Finding documents based purely on intent can help discover documents that do not appear in full-text search results but are relevant to the user. Imagine showing product recommendations to users based on viewing and search history or displaying similar products while viewing a product page.

While working on a written document, you can provide semantic search assistance to the user based on the context of the document. For instance, a journalist writing a story on unethical behavior in a specific sport can search relevant news stories that cover either similar reports in the same sport or similar behavior in other sports.

Entity Extraction

Finding documents based purely on intent can help discover documents that do not appear in full-text search results but are relevant to the user. Imagine showing product recommendations to users based on viewing and search history or displaying similar products while viewing a product page.

While working on a written document, you can provide semantic search assistance to the user based on the context of the document. For instance, a journalist writing a story on unethical behavior in a specific sport can search relevant news stories that cover either similar reports in the same sport or similar behavior in other sports.

Summarization

There are primarily two types of summarization: extractive and abstractive. Both approaches yield summaries that any average reader could make sense of the contents of a large document. Extractive summarization picks the document's key components that effectively explain the whole text. The abstractive approach involves writing a summary with words not part of the original text but similar to a human-written summary.

Clustering

Clustering means grouping similar documents. It helps recommend similar products and topics. Measuring the similarity of a document to others can help identify the semantically nearest ones. This approach requires natural language processing and natural language understanding.

Getting Started

Data Preparation

JSON – JavaScript Object Notation – is a format for data interchange and currently the only accepted format.

Follow these steps to prepare your data:

1. Prepare your data in a spreadsheet
2. Add columns for title, description, pricing, link to the product page, an image link, and any other required field.
3. Create rows for each product or document
4. Save the spreadsheet as a CSV file
5. Search on Google with the query Convert CSV to JSON
6. Pick an online tool to convert your CSV file to JSON format

JSON structure

```
[  
  
  {  
  
    "field1": ".....",  
  
    "field2": ".....",  
  
    "field3": ".....",  
  
    ".....": "....."  
  
  },  
  
  {  
  
    "field1": ".....",  
  
    "field2": ".....",  
  
    "field3": ".....",  
  
    ".....": "....."  
  
  }  
]
```

```
}  
]
```

You can add new records through the Add Records utility.

Create An Index

1. Sign up (or Sign In if you have already signed up)
2. On the App homepage, you will see the Index Catalog with a Create Index button
3. Click the Create Index button
4. On the Create Index page,
5. Choose a JSON file to upload
6. Enter a title
7. Add a description
8. Click Create New Index button when it is active
9. You will be redirected to the Indexing Logs page.
10. The indexing process may take several minutes or more depending on the number of records in the uploaded file.
11. You can stay on the Indexing Logs page or come back to check the progress.
12. On completion, you MUST generate API Keys to search the index and optional create a Demo Search UI.

Generate API Keys

You can generate API keys after the indexing process is complete and the status is ACTIVE. To create API Keys:

1. Navigate to the Index Catalog page
2. Select the Index
3. On the Index page, click on API Keys
4. Click on the Generate button to generate keys
5. You will see API keys assigned to the specific functions
6. Copy and save the keys. You can view and regenerate keys at any time.

Working with No-Code Search UI

You create Demo Search UI pages to test and share.

*Sharing the Demo Search page will expose the API keys to everyone with the link to the page.

1. Navigate to the Index Catalog page
2. Select the Index
3. On the Index page, click on Search UI.
4. On the Search UI page, click the Create button. If you have previously created a UI page, you will be able to edit it from the list.
5. Fill up the details in the Search UI Creation Form:
 - a. Page Title (required)
 - b. Content Details (required) - Enter the source, copyright and other details about your data.
 - c. Show Summary (optional) - Select the checkbox if you want to show the summary of the search record that shows the appearance of the keywords in the search query.
 - d. Semantic Ranking (optional) - Select the checkbox if you want to semantically rank the full-text search results. Selecting this checkbox opens up a text box where you can enter contextual text for personalization.
 - e. Highlight Field (required) - Pick the highlight field from the list of fields in the dataset. This will be used to highlight keywords after the full-text search.
 - f. Title Field (required) - Pick the title field from the list of fields in the dataset.
 - g. Page/Document URL (optional) - Pick the field which has the document link
 - h. Image URL (optional) - In the case of Product layout, pick the field that links to the product image.
 - i. Functions
 - i. Search (Selected by default and cannot be changed)
 - ii. Autocomplete (optional) - Selecting Autocomplete function checkbox enables the Autocomplete AND/OR/NOT buttons.
 - iii. Recommend (optional)
 - j. Visual Settings (optional) - pick the colors and font for custom look-n-feel

- i. Grid Layout
 - ii. Page Background Color
 - iii. Page Font
 - iv. Tag Background Color
 - v. Tag Title Color
 - vi. Summary Text Color
 - vii. Button Background Color
 - viii. Button Text Color
6. Entering the required fields: Page Title, Highlight field, Content Details and Title field will activate the demo utility.
7. You can change the look-n-feel and perform all the functions.
8. Click Submit to create the Search UI
9. On successful submission, you will be redirected to the Search UI page, and the newly created page is listed in the pages list.
10. You can edit the page, open it in a new window, copy the URL, or delete it with the Actions dropdown.

Developer API

The Seqwa API is organized around REST. Our API has predictable, resource-oriented URLs, and uses HTTP response codes to indicate API errors. We use built-in HTTP features, like HTTP verbs, which are understood by off-the-shelf HTTP clients. We support cross-origin resource sharing, allowing you to interact securely with our API from a client-side web application (though you should never expose your secret API key in any public website's client-side code). JSON will be returned in all responses from the API, including errors.

Credentials

Before accessing the REST API, you need to note the Index ID and API Keys. You can do with these steps:

1. Navigate to the Index Catalog page
2. Select the Index
3. On the Index page, click on API Keys.
4. Copy the Index ID and the API Keys.
5. Seqwa offers function-specific API keys so you can choose the function that best suits your needs.

Autocomplete

Autocomplete performs a fundamental and essential to help users complete a search. Autocomplete offers query suggestions, related named entities, and top full-text search results.

Implementing a user interface must include an API call for any change in the query, like adding or removing a character. Having a caching mechanism can considerably reduce the number of API calls.

Request URL

<https://www.seqwa.com/api/v1/autocomplete>

Method

GET

Headers

content-type	application/json
seqwa-api-key	API KEY generated for the Autocomplete Function

Params

index	Index Id	required
query	Query text entered by the user. Supports Lucene query parser syntax which includes AND, OR, and NOT operands, term query like fieldName:fieldValue, and range query like fieldName:[minFieldValue TO maxFieldValue]	required
highlightfield	Field for highlighting the query text	optional
fields	Array of fields	optional

maxresults	Maximum results to be returned. (Default: 10, Limit: 25)	optional
-------------------	--	----------

Request URL Structure

`https://www.seqwa.com/api/v1/autocomplete?index=`

`INDEX_ID&query=QUERY_TEXT&highlightField=SEARCH_FIELD`

`&fields=FIELDS&maxResults=MAX_RESULTS`

Response

suggestions	Contains an array of suggestions for queries based on the query text. Each element of the array is an object with suggest and displaySuggest parameters. The suggest param is plain text and displaySuggest has highlighted words with the html <code></code> element.
relatedItems	Contains an array of top tags from the search results.
records	Top search results based on the search query and up to maximum number of results requested.
error	Returns an error message if any error occurs on the server.

Response JSON Structure

```
{
  "suggestions": [
    {
      "suggest": "...",
      "displaySuggest": "<b>....</b> ..."
    },
    {
      "suggest": "...",
      "displaySuggest": "<b>....</b> ..."
    }
  ],
}
```

```
"relatedItems": [  
  "...",  
  "..."  
],  
"records": [  
  {  
    "recordId": "##",  
    "title...": ".....",  
    "link...": ".....",  
    "image.....": "....."  
  },  
  {  
    "...": ".....",  
    "...": ".....",  
    "...": "....."  
  }  
],  
"error": "....."  
}
```


Search

Search function integrates full-text search with semantic ranking to provide better search results based on context or user preferences. Semantic ranking personalizes full-text search results for products based on user browsing or purchase history. It includes support for double-quoted phrases for specificity and AND/OR/NOT operators for filtering.

Request URL

<https://www.seqwa.com/api/v1/search>

Method

GET

Headers

Content-Type	application/json
seqwa-api-key	API KEY generated for the Search Function

Params

index	Index Id	required
query	Query text entered by the user. Supports Lucene query parser syntax which includes AND, OR, and NOT operands, term query like fieldName:fieldValue, and range query like fieldName:[minFieldValue TO maxFieldValue]	required
highlightField	Field for highlighting the query text	optional
fields	Array of fields	optional
type	Takes the options: ranked and unranked. (Default: unranked)	optional

context	Contextual text for personalization with semantic ranking. The type MUST be set as ranked to apply the context.	optional
maxResults	Maximum results to be returned. (Default: 10, Limit: 25)	optional

Request URL Structure

<https://www.seqwa.com/api/v1/search>

?index=INDEX_ID&query=QUERY_TEXT

&highlightField=SEARCH_FIELD&fields=FIELDS&type=RANKING_TYPE

&context=CONTEXT&maxResults=MAX_RESULTS

Response

searchResults	Contains an array of search results for queries based on the query text. Each element of the array is an object with recordId, summary, displaySummary, and requested fields as parameters. The summary param is plain text and displaySummary has highlighted words with the html element.
topTags	Contains an array of top tags from the search results.
error	Returns an error message if any error occurs on the server.

Response JSON Structure

```
{
  "searchResults": [
    {
      "recordId": "##",
      "summary": "...",
      "displaySummary": "<b>....</b> ...",
      "field1": ".....",
      "field2": ".....",
    }
  ]
}
```

```
    "field3": "....."
  },
  {
    "recordId": "##",
    "summary": "...",
    "displaySummary": "<b>....</b> ...",
    "field1": ".....",
    "field2": ".....",
    "field3": "....."
  }
],
"topTags": [
  "...",
  "..."
],
"error": "....."
}
```

Recommend

Recommend function searches based on semantic similarity, also called semantic search, including finding similar documents for a given document or recommending documents based on the context. For instance, you can suggest similar products on the current product page or create a list of recommended products based on user browsing or purchase history.

Request URL

<https://www.seqwa.com/api/v1/recommend>

Method

GET

Headers

Content-Type	application/json
seqwa-api-key	API KEY generated for the Recommend Function

Params

index	Index Id	required
recordId	Id of the record to search for its recommendations.	conditional
context	Contextual text for finding the recommendations.	conditional
fields	Array of fields	optional
maxResults	Maximum results to be returned. (Default: 10, Limit: 25)	optional

Either recordId or context is required.

Request URL Structure

```
https://www.seqwa.com/api/v1/recommend?index=
INDEX_ID&recordId=RECORD_ID&context=CONTEXT
&fields=FIELDS&maxResults=MAX_RESULTS
```

Response

recommendations	Contains an array of records for queries based on the semantic search query or the selected item. Each element of the array is an object with recordId and requested fields as parameters.
error	Returns an error message if any error occurs on the server.

Response JSON Structure

```
{
  "recommendations": [
    {
      "recordId": "##",
      "field1": ".....",
      "field2": ".....",
      "field3": "....."
    },
    {
      "recordId": "##",
      "field1": ".....",
      "field2": ".....",
      "field3": "....."
    }
  ],
  "error": "....."
}
```

Index

You can view information and manage an Index with seven operations: API Usage, API Keys, Search UI, Add Records, Indexing Logs, Update Information, and Delete Index.

View API Usage

View monthly usage of the index with a breakdown for each REST API function.

1. Select the month from the Date Picker field
2. Click the Get button.
3. You will see the API usage for each function for the current index.

Add Records

Add new records to your index.

1. Prepare a JSON data file with the same structure as the one used to create the index.
2. Upload the file with the Choose File field
3. Click the Update Index button to process the upload file
4. You will be forwarded to the Indexing logs page.

(Re)generate API Keys

View, generate, and regenerate API keys for all the functions.

Search UI

Quickly build and manage Search UI with simple custom configuration. You can create a demo Search UI by clicking the Create button. Existing Search UIs are listed with Actions dropdown. You may edit the design, copy the link, open the link in a new window, or delete the UI configuration. *Sharing the Demo Search UI page will expose the API keys to everyone with the link to the page.

Update Information

You can view and update the title and description of the Index.

View Indexing Logs

View the logs to check the progress of the indexing process when you have created a new index or added new records.

Delete Index

Deleting the index will permanently remove all the data and Search UIs associated with the index. This operation is irreversible.

Account

View API Usage

View monthly usage of the whole account with a breakdown for each REST API function.

1. Select the month from the Date Picker field
2. Click the Get button.
3. You will see the API usage for each function for your account.

Add API Credits and Storage

You MUST purchase API Credits and Storage Units separately. Please view the Pricing page for details.

View Purchase History

View past purchases made for both API credits and Storage units. You can view the purchase receipts hosted on the Stripe website.

Redeem Code

You can redeem your code acquired from third-party purchases and promotions. To redeem:

1. Enter the redemption code
2. Click Submit
3. If the code was accepted, then you will receive an email notification and web notification.
4. You view the newly added API Credits and Storage Units in the Purchase History and the Account page.

Delete Account

Deleting the account will disable logging into your account and permanently remove all the indexes, data, and Search UIs associated with each index. This operation is irreversible.