**Search Assist**

## Summary

SearchAssist is a conversational search platform that combines **natural language understanding** with **intelligent virtual assistant actions.** The result is a **contextualized** search experience that is relevant and personalized for your users.

SearchAssist applications enable quick and accurate **retrieval** of information from vast datasets, giving your users rapid access to the information they need. SearchAssist applications are highly **customizable** and can be tailored to business needs.

SearchAssist applications offer two different approaches to providing relevant information to their users in response to their queries: **Providing search results** and **Generating Answers.** Search functionality aims to provide users with a set of results and associated documents or web pages in the order of relevance to the user’s query. On the other hand, the Answers functionality aims to provide a precise response to a user’s query as opposed to providing a list of relevant results.

Some of the key features of a SearchAssist application are listed below.

* It supports ingesting data in different formats from different sources, including web pages, files, and most commonly used third-party content management systems.
* It is highly customizable, allowing you to completely control and customize the application’s response per your business requirements.
* It can be seamlessly integrated into any of your existing applications to provide an engaging user experience.
* It can be easily integrated with the most common large language models for answer generation.
* The debug and test features of the application give you complete insights and allow you to optimize the performance of the application for your users.
* SearchAssist application also provides you with analytics data to understand the usage and behavior of the application over time.

## First Application

A typical SearchAssist application development goes through the following behind-the-scenes stages to generate answers or provide search results.

A diagram of a search results

Description automatically generated

To provide relevant results to a user query, a search application first indexes the data as per the index fields identified for the data. Next, it performs a series of manipulations on the data as per the workbench processes defined followed by fine-tuning the relevance of the fields, and then personalizing the results before presenting to the user as the response. For an answer application, SearchAssist prepares an answer index by dividing the input data into relevant chunks. Depending upon the model selected for answer generation, the most relevant, direct, and precise answer is generated and displayed to the user.

Steps involved:

1. Understand Business Requirements
2. create your application
3. import data from sources
4. implement search functionality : Search functionality aims to provide users with a set of results along with the URLs or references to the documents or web pages that are used to derive the results or the source of truth.
   1. indexing data: transformation of ingested data to extract relevant information as per the business requirements and enable efficient retrieval. Add index fields along with default indices, configure the workbench stages which define how the ingested data is to be transformed before indexing.
   2. Finetune relevance: You can fine-tune the search results to fetch and display the most relevant results to the users efficiently. It is the process of adjusting the parameters to improve the quality of search results in terms of accuracy and relevance to the user query.
5. Implement answer functionality: Answer Retrieval refers to providing a specific response to a user’s query as opposed to providing a list of documents or web pages.

* Choose the answer generation model.
* enable answer snippets: go to the **Answer Snippets** under the **Indices**tab and enable one or more of the supported models. Next, click on **Train**to allow the SearchAssist application to prepare its answer index. In this process, the application automatically generates chunks from the source data. These chunks are then used for answer generation.
* optimize chunks: on enabling answer snippets for an application, the application automatically generates chunks from the source data. SearchAssist provides a chunk browser for you to assess and analyze the extracted chunk, giving you an insight into the output from the extraction process and enabling you to undertake subsequent actions if required.

1. Personalize results: process of defining rules based on your business requirements that generate custom responses for each query and position responses in the search results. You can use Results Ranking, Facets, and Business Rules to personalize the search results by selecting appropriate filters based on the field values. Similarly, you can use business rules to personalize answers as per your business needs.
2. Setup search interface.
3. test and debug with preview option. To test workbench stages, use simulate option.

## **Concepts**

### **Sources**

SearchAssist simplifies searching by allowing the ingestion of data of different types and from varied sources to expand the range of information that is available to the users for search.

To add, update or view any type of data in SearchAssist, go to the **Sources** tab.

#### **Files**

SearchAssist allows you to upload and index files hosted on local machines or remote servers. This functionality streamlines the process of locating information, eliminating the need to manually locate specific documents and sift through their contents, thus saving users both time and energy. You have the option to upload single files from a local device or a URL. Alternatively, you can also upload many files simultaneously from a designated directory.

Supported File formats:

* Search Results: PDF, DOCX, PPT, JPEG, PNG, and TXT
* Extractive Answers: PDF
* Generative Answers: PDF, DOCX, PPT and TXT

Scanned PDFs and password-encrypted PDFs are not currently supported.

under **Sources,**click **File Upload.**

#### **Web Pages**

If, the content of the websites is indexed, the data and information can appear in the search results for the user. SearchAssist enables you to ingest web content through web crawling at regular intervals to perform efficient data searches.

 go to the ***Sources***tab and click ***Web Crawl***.

#### **FAQ**

FAQs can be added from files or a URL. Files can be JSON, CSV or PDF. In case of PDF, we can highlight and annotate the question and answer, header, footer. We can even add FAQs manually.

FAQs are first drafted, reviewed and finally approved. Only approved FAQs become part of search content.

#### **Structured Data**

Structured data typically refers to data in a standard format that can be easily organized and is searchable. A product catalog for an e-commerce application is an example of structured data where the fields like name, description, features, cost, etc. for each record or item are standard. You can manually add data to the application or upload structured data from a JSON or CSV file. When you add new structured data to your application, you need to identify index fields from the data and add them to the Index Configuration. This is required for the application to include this data in the search results.

To facilitate indexing and search of the content ingested as structured data and to display it as part of search results, follow the steps listed below.

1. Identify and add Index fields from the structured data.  Use column headers( for data in CSV format) or keys(for data in JSON format) from the data as Index fields.
2. Mark them as searchable and presentable.
3. Assign appropriate weights to the fields as per your business requirements, so that relevant content is displayed in the results. Fine-tune other search settings, if required.
4. Map the field names in the results template so that they are included in the results layout view.

#### **Connectors**

SearchAssist offers seamless integration with various content sources and repositories, including but not limited to ServiceNow, Microsoft SharePoint, Atlassian’s Confluence, and IBM Domino, through the **Connectors**. Connectors facilitate the extraction and search of data from diverse third-party applications, CRM systems, web databases, and cloud-based platforms. SearchAssist can be configured to connect to multiple systems at the same time using a connector for each third-party application allowing you to set up a centralized and efficient data retrieval and indexing process.

#### **Virtual Assistants**

You can integrate an active virtual assistant capable of performing actions related to intents from your SearchAssist app. This integration allows end-users to invoke virtual assistant actions from the search interface. SearchAssist identifies and maps the intent behind user queries, if any, which trigger actions that the linked virtual assistant offers.

### **Indices**

Indexing is the technique by which search engines **organize** the data ingested from the content sources to enable efficient and rapid **retrieval** of relevant information in response to user queries. The indexing process includes the identification of the indices from the content, pre-processing, and structuring of the data in an index. An index is like a data structure which stores data of a particular kind. The indexing process **contributes** to the overall **effectiveness** of the search.

#### **Index configuration**

Index Configuration refers to the set of parameters and rules used for preparing the index for the data.

To **create a new Index Configuration,** click on the drop-down icon and select **+ Create New** option.

##### **Index Fields**

Searching for relevant results in the ingested data can be a slow and inefficient process. But SearchAssist indexes the source data. Indexing is like organizing the data for quick and relevant search retrieval.  It stores the **ingested data as field and value pairs**. Index fields are like the **schema of a database** and define the data stored with SearchAssist.

By default, SearchAssist automatically adds some pre-defined index fields associated with web pages, documents, and FAQs (<https://docs.kore.ai/searchassist/manage-indices/default-index-fields/>)

in this tab, we have options to add, search, edit, delete a field.

while adding, options to look out are **searchable** and **autosuggest** checkboxes.

##### **Traits**

Traits are characteristics that can be identified and extracted from the search user’s input. They provide insight into the search user’s intent.

Traits are specific entities, attributes, or details that users express in their conversations. The utterance may not directly convey any specific intent, but a trait in an utterance can lead to intent detection and drive bot conversation flow.

You can extract traits from documents. SearchAssist can then map those traits to user queries and use the information to help to increase the relevance score. Use the traits added to the SearchAssist app to fine-tune an index workbench and configure business rules.

For example, instead of a simple query, “*How do I pay my bill”*, the user might say, “*I was wondering if it would be possible to make payments.”*  To ensure the SearchAssist app generates the same result for either query, manually define a common trait for these two utterances. Tag the corresponding content source or answer with that trait so that future queries generate the most effective search result.

##### **Workbench**

Workbench is a SearchAssist tool that converts content into objectively indexed documents. It processes the ingested content in a series of stages known as the Index Pipeline. Collectively, the Index pipeline converts the ingested content into a document ready for indexing. You can use the ***simulator***to test the configuration at each step individually or the cumulative effect of the various stages.

A diagram of a stage

Description automatically generated

SearchAssist supports the following Index Pipeline stages.

* **Field Mapping** maps fields in an indexing pipeline document to a target field, sets values, copies values, removes fields, renames fields, and more.
* **Entity Extraction**uses NLP techniques to identify named entities from the source field. For example, finding dates or geographic locations from a document.
* **Traits Extraction**extracts specific attributes that search users might express in their conversations. When traits are extracted from the source documents, SearchAssist can find more relevant results for the users.
* **Custom Script** stage allows you to enter customized scripts to perform any operations on the fields like deleting or renaming fields.
* **Keyword Extraction**automatically detects and extracts important words stored in a field.
* **Exclude Document** stage drops all the documents that match the specified condition.
* **Semantic Meaning** is a technique to understand the meaning and interpretation of words, signs, and sentence structure. This stage currently supports web page-related sources only.
* **Snippet Extraction** stage allows you to configure the answer snippets generation from the documents.
* **Custom LLM prompt** stage allows you to configure the prompts to be used for data enrichment using the third-party LLMs.

A screenshot of a computer

Description automatically generated

Example of a prompt: Write a summary of {{file\_content}}. Try to use only the information provided. Here file\_content is an index field.

We can add a stage, order them, activate or deactivate a stage, or delete it.

The Workbench comes with a built-in simulator that has an interactive preview of how the stage rules affect a document before it’s indexed. Click the ***Simulate***button.

##### **Index Settings**

Index Languages: SearchAssist offers multilingual support that allows you to enhance your customer experience. SearchAssist not only allows you to add and index content in any of the supported languages but can also interpret user queries in the supported and enabled index languages, search through content in those languages, and even, present both search results and answer snippets in the same language as that of the user query.

You can only enable a maximum of four supported languages at a time in a given index configuration.

Note that when German or Spanish languages are added and the Generative snippets are enabled, the embeddings model is changed internally and new vector embeddings are generated. This will modify any previously generated answers for the English language. Hence, it is recommended to verify the answers generated for previously added content again.

#### **Search Configuration**

##### **Search Settings:**

Search Relevance refers to the accuracy of the results provided by the Search application in response to a user query. It is a measure of the performance of the search application. SearchAssist provides various configuration properties like weightage, synonyms, search relevance, etc. that you can fine-tune as per your business needs and return the most relevant results to your users.

* **Weights**: configure weights for the searchable fields. The more the weightage, the more important the field. (0-10). By default, weights are defined only for the pre-defined fields.
* **Highlighting**: This feature allows you to highlight and emphasize keywords matching to the search query in the search results. It improves readability and helps users quickly identify query matches in the search results enhancing the search experience of the users. We can define the style of highlighting and the index fields that are to be highlighted. highlighting a field makes it presentable.
* **Presentable**: used to configure fields that are used in displaying the search results. The fields added as Presentable are included in the search results. Fields that can be used as search results should be added as Presentable fields like page\_title, faq\_answer, etc. Some of the standard pre-defined fields are added as presentable fields by default.
* **Synonyms**: We can define synonyms and one-way synonyms and can be used to compute search relevance.
* **Stop Words**: Stop words are the set of commonly used words in a language that do not really provide any keywords useful for search but are used in sentences. We can choose to add stop words to a default set or create an entire set of stop words.
* **Search Relevance:** we can configure threshold for a match to be considered as a search result. Search Relevance can be enabled if you want to ensure that the search results contain the subject, object, verb, or a combination of these language components same as that in the search query. After the feature is enabled, select the language components like subject, object and verb of the search query to be matched in the results.
* **Spell Correction:** enable or disable spell correction while searching. Add the fields on which spell correction needs to be applied from the query. There are some other parameters that are intuitive.
* **Prefix Search:**

Eg: let us assume that there are two data records:

1: “Check out these fighter jets flying over my town!”

2: “The air show featured a variety of fighter jets, including the F-16 and the F-35.”

And the search query is “fly fighter je”.

If the prefix search type is set to ‘**Prefix All’,**  the search result would return only the first record since it has all the words in the prefix, irrespective of the order of the words.

However, if the prefix type is set to ‘**Prefix Any**’, the search result would return both the records since both the records contain at least one of the words from the search query.

We need to select the fields to be used for the Prefix search in the search results.

* **Small Talk:** enable or disable small talk.
* **Bot Actions:** Choose the behavior when query qualifies a dialogue task.

##### **Personalizing Results**

Personalization allows you to show different results to different users for the same query depending on their user profiles or search context.

SearchAssist personalization uses the NLP approach in identifying the user’s context, detecting facets, intent, localization needs, and objective behind the query. The machine learning (ML) capabilities of SearchAssist can adapt to the priorities and preferences of the end-user prompt with ready queries and position results.

Personalizing can be done by 3 features. **Facets**, **Business Rules**, and **Result Ranking.**

* Facets allow users to define filters to show search results.Two types: Range and Value Facet.
* Result Ranking: As a business user, you might need to promote certain products based on context or a mandate. Perhaps you prefer to position results in a particular order. SearchAssist allows you to fine-tune the search results based on your business priorities. You can hide/unhide, boost or lower the position of certain results.
* Business Rules: ?????????????????????????

##### **Answers**

Answers are specific pieces of information extracted or generated by a search application in response to a user query.

Search results usually present a list of documents, web pages, or content retrieved in response to a user query, ranked based on their relevance to the query. On the other hand, Answers aim to directly address user queries and provide an exact, precise piece of information as the response.

1. **Extractive Answers:** Extractive answers involve selecting and presenting relevant chunks of text directly from the source documents that contain the answer to the user’s query. Extractive answers preserve the original wording and structure of the content
2. **Generative Answers:** Generative answers involve using the retrieved chunks to generate answers to the query based on the understanding of the question and the relevant information in the source documents.
   1. Answers are usually paraphrased to reply to the exact user query.
   2. LLMs are used to generate answers from retrieved chunks.

The answer-generation process mainly consists of the following steps:

1. **Content Ingestion**: Involves processing of source documents that will be used for generating answers.
2. **Chunking**: Involves breaking down the source documents into smaller, meaningful units called chunks.
3. **Generating Vector embeddings**: Involves converting chunks into multi-dimensional vectors representing the chunks.
4. **Chunk Retrieval**: Involves selecting the most relevant chunks of text from the vector space based on their similarity to the user query.
5. **Answer Generation**: Involves generating a response to the user query based on the retrieved chunks.

For extractive answers, the chunks are extracted using a**rule-based chunking strategy** that uses headers and paragraphs under the header to identify chunks.

Generative model leverages the power of large language models to generate an answer snippet from the most relevant chunks. Currently, SearchAssist supports using OpenAI and Azure OpenAI LLMs to generate the answer snippets. For Generative Answers, a plain text extraction model is used. This extraction model treats all text the same way, i.e., title, sub-title, content, everything would be read as content only. Chunks are generated as per the number of tokens defined. By default, this value is set to 400. While retrieval, the top n chunks as configured matching to the user query are retrieved and sent to the LLM for answer generation.

##### **Search Experience**

customize how the UI of search assistant and display of results.