elastic

Implementing search and generative AI for your knowledge base

Implementing powerful search for your knowledge base is easy to understand in theory, but can be intimidating in practice, especially if you're considering building a state of the art generative AI experience for your end-users.

elastic.co

Table of contents

Introduction

Congratulations on making the decision to implement search for your knowledge base! Search is on the critical path to a winning self-service support strategy since customers and agents can navigate directly to the search bar for answers. You're likely already aware of the benefits that highly relevant search brings to your online support content and documentation: fewer inbound support tickets, streamlined customer support, and a better overall customer experience, just to name a few.

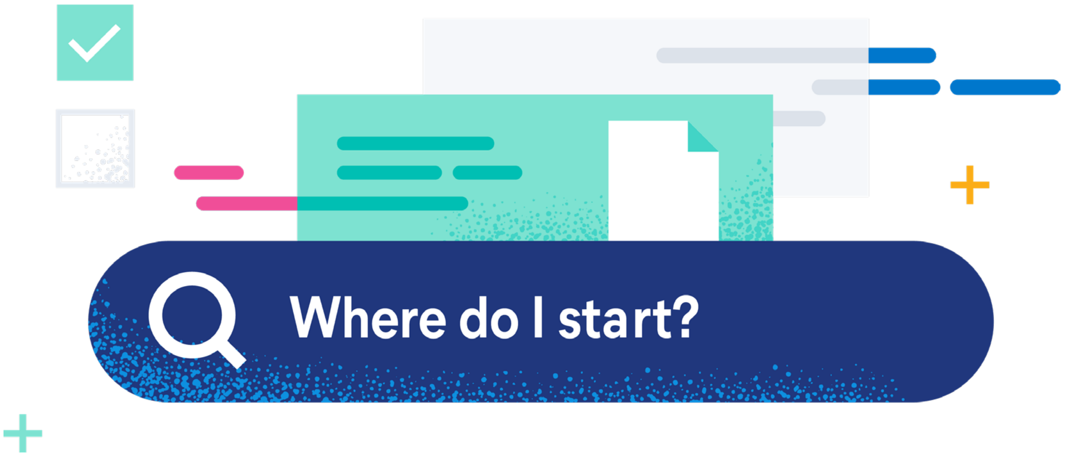
> 70% of organizations are currently exploring generative AI and 19% are already in pilot or production mode. The primary focus of AI investments? Customer experience and retention.[1]

However, when assessing options for implementing search, you may encounter a whole set of issues you've never had to deal with before. For example, you're probably well aware of all the buzz — and the dizzying array of options — surrounding artificial intelligence and generative AI technologies like ChatGPT. Fortunately, you've come to the right place for help.

This ebook explores the different search implementation options available for your knowledge base and takes you through the four phases of search implementation:

- Planning
- Implementation
- Testing
- Maintenance

And in the process, we'll give high-level guidance on the what, when, and why of AI.



[1] | Gartner poll of 2,500+ executive leaders, May 2023

Build vs. buy: Which implementation is right for you?

The type of search implementation you choose will depend on the goals you've defined for your search, the resources you have available, and your unique working environment.

Let's take a quick look at two different types of search implementation: out-of-the-box content management system (CMS) search and cloud-based search. The basic differences are outlined below, with more in-depth explanation on the pages that follow.

Out of the box

- No work required
- One size fits all
- Cannot be customized
- Keyword-based search
- Minimal tools for customizing search results and tuning relevance
- Limited search analytics
- Limited and/or or black-box artificial intelligence capabilities

Custom cloud search

- Some development work required
- Fully managed on cloud
- Data ingestion flexibility, with capabilities for unifying disparate sources of content
- Built-in relevance and language recognition technology
- Customization, relevance tuning, and performance analysis via dashboard built directly into product
- ML-enabled capabilities: natural language processing, semantic search, sentiment analysis, and personalization
- Integration using retrieval augmented generation (RAG) with third-party large language models (LLMs) and/or generative AI
- Flexibility to bring your own domain-specific ML model (proprietary or third-party)

Out-of-the-box CMS

During the planning provcess, some companies may feel that the out-of-the-box search built into their CMS is adequate for their customers and support agents. While in some cases this may be true, many companies soon discover that out-of-the-box search is inherently inflexible and cannot be customized for the specific needs or desires of your support team.

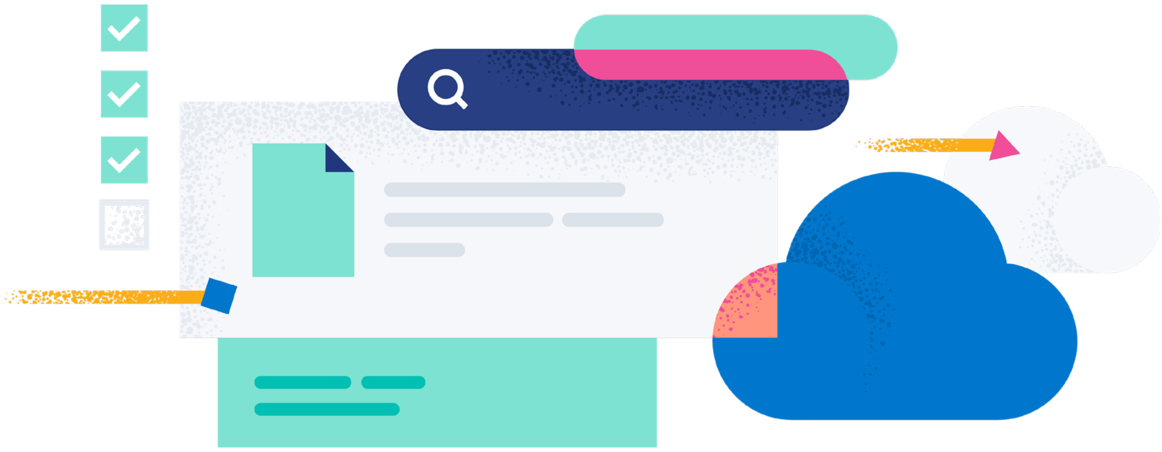
As a one-size-fits-all solution, CMS search relevance models cannot be changed, leaving support teams guessing about how they might influence which search results appear at the top of the list for a given query. Perhaps more importantly, out-of-the-box search is often built with little to no support for common language problems, such as spelling correction (also known as typo tolerance), synonym recognition, or basic language parsing technology. Companies that want a customized search solution with more advanced relevance, language modeling, natural language processing, and/or the ability to integrate their proprietary data and content with generative AI experiences will need to invest in building new search on their own or look to third-party search solutions with advanced capabilities built in.

Custom search fully managed on cloud

If you want the benefits of an advanced search engine with language modeling technology and machine learning capabilities and can spare a few development cycles, then custom search is probably the best option for you.

Cloud-hosted or serverless search solutions allow complete control over your search experience, but free up your engineers from highly complex tasks such as building advanced relevance models or developing natural language processing for your search engine. You get the highly relevant search results and natural language understanding boosted by machine learning that these custom tools can provide.

Cloud-hosted or serverless search also saves your engineers from dealing with recurring software updates and ongoing server maintenance, and often comes with advanced analytics and optimization dashboards that non-technical staff can take advantage of.



Next steps: The key phases of implementing search

Now that you understand which implementation options are available, you can begin to scope out the four phases of search implementation: planning, implementation, testing, and maintenance:

Planning

Gathering stakeholders, designing search, setting goals, proposing timeline, preparing a migration plan.

Implementation

Indexing, frontend coding, backend server configuration

Testing

Checking search result quality, testing common language problems

Maintenance

Tuning and optimizing search results, maintaining software

Planning

What is the goal?

The first step in the planning phase is to look at your own people, processes, and systems and develop a set of best practices for your own knowledge base to ensure a smooth implementation. At the end of planning, you should know how your new search experience will work, who will be responsible for executing what, and how long the project will take.

What advantages does AI-powered search bring?

High relevance is the name of the game when it comes to AI in the search space. Advanced retrieval like vector, hybrid, and semantic search can help your end-users find accurate responses faster using queries composed in natural language — even if they don't have domain expertise. Retrieval augmented generation provides highly relevant search results via context window to large language models. These results supplement generative AI experiences with proprietary, real-time data to give customers even better human-like responses (in question-answering or chatbot experiences, for example) at a lower cost.

Even without data science expertise or machine learning engineers — or the need to spend a ton of money on training machine learning models — you have options for riding this wave of exciting technology. Search solutions that understand user intent and have flexible AI capabilities built in offer the best of both worlds: AI-enhanced search without a great deal of implementation difficulty.

Who is involved?

Once these best practices are in place, you can move on to gathering your project team members. Your planning team may vary based on your org structure, but here's a sample list of folks you'll need to involve:

- VP of support to champion and guide
- Software engineer for development work
- Data science or machine learning personnel, if you have them, to weigh in on and assist with artificial intelligence implementation
- UX designer for frontend work
- Customer service representatives and support content creation team to provide input on requirements

How do you want the search experience to work?

With your team in place, you can move on to one of the most important steps in the planning process — deciding how you want your search to work. Why is this step so critical? Because what you define in this step determines how visitors will experience your knowledge base search.

You also might want customers to be able to search by topic, content type, or date/time frame, as well as have the ability to filter results by topic, tags (such as "getting started" or "troubleshooting"), content type, or reviews.

To help customers find your content faster, you may want to implement an autocomplete experience that features personalized results (such as links to blog posts, articles, or technical documentation) rather than suggested queries. This can help get visitors to the content they need immediately to solve their issue.

Since end-users may not have the exact vocabulary to describe their support issues, you might want to have advanced search retrieval options like vector, hybrid, or semantic search, which goes beyond basic, keyword-based search and uses embeddings to understand the context and meaning behind search queries.

This is when your development team — and data science team, if you have one — will decide what role machine learning or generative AI will play in improving the search experience. Your customer and support team needs and business goals are the driving factor.

Once you determine and document how you want your search to work, it's time to think about how to index your knowledge base content — a topic discussed in depth in the implementation section.

Who will do what, and when?

In the last step of the planning process, you'll estimate project timelines, as well as develop training and communication schedules. Timeframes are variable depending on the complexity of your project and the scale of data in your knowledge base. Use the best practices you created, the specs you outlined for the search experience, and the schedules and bandwidth of key project stakeholders to build out a timeline.

> Looking for inspiration and best practices? Solutions with active developer communities can help your development teams advance faster and provide answers to common questions on all phases of implementation.

You'll also want to allocate time for training your customer service representatives and your support content creation team. They'll need to understand customer search analytics and the process they'll need to follow when adding new content to your knowledge base.

Finally, you'll need to develop a communications and change management strategy in order to let people in your company know that new search is coming, and how much easier it will be for them and their customers to quickly find the information they need.

LLMs: To train? Or not to train?

Perhaps you've been hearing about large language models, or LLMs, and the role they play in artificial intelligence. These models are trained on massive datasets, and they help build the "intelligence" that AI uses to operate. But as you might guess, the enormous computing resources needed to train an LLM (and fine-tune it over time) are definitely not cheap — as in millions of dollars.

So for the most part, building or training an LLM on proprietary data is out of reach for all but the largest organizations with deep data science expertise. But on the bright side, you can select search technology that can provide context to generative AI technology to produce highly relevant search results based on your domain-specific data.

The key is to choose an AI solution that gives you a spectrum of AI options — from the flexibility of semantic search out-of-the-box to pretrained language models that do a lot of the heavy lifting for you.

What's happening?	DESIGN ▶	INDEXING ▶	FRONTEND ▶	TRAINING
Who's involved?	UX, Support	Engineering, Data Science	Engineering	Support

Implementation

Data preparation and enrichment

For any search experience, but especially for AI-enhanced search, the quality of the data is ever-important. As you're preparing the data to ingest in your search solution, think about how this data will be queried and who will be able to access it. To improve search quality, you might want to:

- Generate vector embeddings
- Enrich data with geo location of an IP address
- Label data using a ML zero-shot model (a model trained on a set of labeled examples)

To increase safety, you can redact personally identifiable information (PII). All of these operations can be done prior to ingesting the data or during the ingest process

How can I index my data?

The first step to implementing search is indexing your knowledge base content in a search engine. Once completed, this information can be queried by users and surfaced to them on your website. While the options and requirements for indexing will vary based on the specifics of your implementation, the process generally takes place with a data feed, an API integration, or a web crawler.

· Data feed: Establish an XML, CSV, or RSS file transfer protocol

If you're working with a third-party provider, they may ask you to pass on all of the information you want to index via a data feed. This process involves setting up an FTP transfer of your information in XML, CSV, or RSS format every 24 hours.
Aside from being cumbersome to configure and maintain, data feeds often leave your search index out of sync with your live website offerings, since website changes are only communicated to the provider when the FTP transfer goes through.

· API integration: Build database connection between website and search engine

An API will accomplish the same as a data feed, but without the annoying lag - information is passed on instantaneously to your search engine. API integrations also allow complete control over your engine schema, which allows you to more precisely control how your search experience will work. If someone on your team is experienced with API integrations, then this might be the way to go.

· Web crawler: Extract website information automatically from HTML

If you're looking for the simplest way to index your data, you can get the same results using a web crawler.

When working with a web crawler, you simply provide a URL and the web crawler automatically extracts website information from the HTML. A robust web crawler will allow for crawl scheduling and partial crawls, rules definition for crawling flexibility (such as including or ignoring certain content), and data enrichment.

How long will the frontend work take?

As is usually the case, it depends. In this step, you'll take what you've planned and work with designers and frontend engineers to put everything in place for the user interface. You'll need to build in ample time for design, implementation, integration with your search engine, and deployment.

Your engineering team can take advantage of frameworks that ease the process of building and customizing the user interface. As with all phases, be sure to add in some buffer, as tasks often end up taking longer than anticipated.

The right eyes on the right documents

Another common concern among organizations implementing generative AI experiences for their knowledge bases is having access control over their data. The key is to choose an AI solution with native document-level controls so that only appropriate access to data is granted.

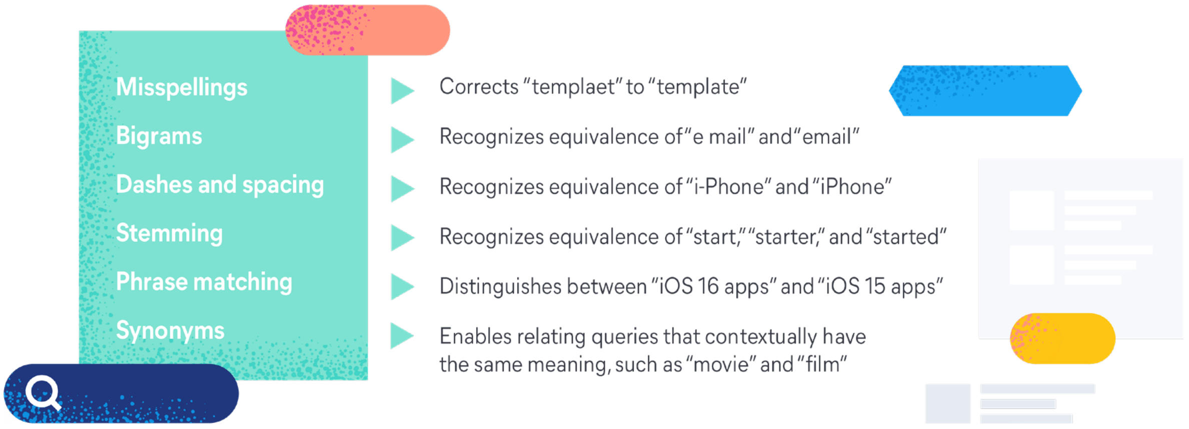
Testing

How do search results look for my most important queries?

In order to ensure your search is working properly, run tests to see how your search handles your top 50 to 100 queries. The key here is analyzing relevance and asking yourself: Do these search results meet my expectations? Are these the most helpful answers for this search? Is the experience able to capture the intent of user queries, even if they don't match specific keywords? If not, you should customize the results for these queries until they match expectations. Your relevance tuning options may be limited with out-of-the-box search, but with custom search you'll have a variety of methods for fine-tuning and optimizing result relevance.

How does my search handle common language issues?

In addition to relevance, your team should investigate how your search handles common language issues for user queries, such as the following:



Misspellings ▶ Corrects "templaet" to "template"

Bigrams ▶ Recognizes equivalence of "e mail" and "email"

Dashes and spacing ▶ Recognizes equivalence of "i-Phone" and "iPhone"

Stemming ▶ Recognizes equivalence of "start," "starter," and "started"

Phrase matching ▶ Distinguishes between "iOS 16 apps" and "iOS 15 apps"

Synonyms ▶ Enables relating queries that contextually have the same meaning, such as "movie" and "film"

Users have come to expect these common features, but out-of-the-box search may not include them. Custom search solutions offer much of this functionality by default, with plentiful options for fine-tuning.

Maintenance

What is required to maintain my index structure?

Once you've determined how your index will be structured, each new piece of content your teams create will need to conform to that indexing structure.

If you're using a web crawler, maintenance will be simpler. Once your site is recrawled, that content is automatically indexed. Finally, ensure that engineers have mapped your CMS to the correct indexing structure.

How can we optimize our search over time?

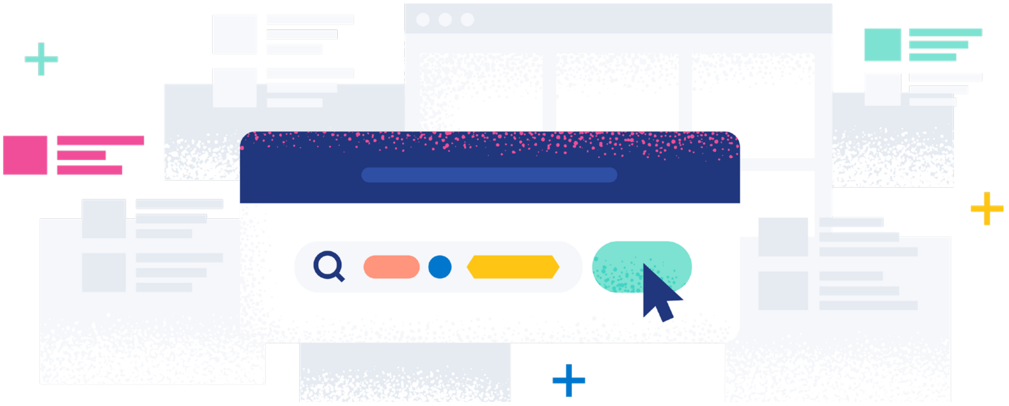
To optimize your search and enable most customers to self-serve over time, be sure to keep track of key performance indicators that matter most. Some key data points to monitor:

- **Clickthrough rate** - Users clicking on top results, indicating they've found what they're looking for.
- **Search exit rate** - Users abandoning search, potentially indicating they didn't find what they wanted.
- **Percentage of searches with no results** - May indicate content gaps in your knowledge base.

The key is not to get bogged down with metrics. Choose carefully those that demonstrate the most impact and continually monitor how they affect your inbound ticket volume.

Finally, search relevance is paramount, so plan to optimize your search by continuously customizing your search engine:

- Monitor and tune your algorithm to control what impacts the order of search results.
- Customize results for individual queries to feature the most important content.



About Elastic Search

With the Elastic Search toolkit, development teams get everything they need to build, tune, and manage search for any application, including knowledge bases. Millions of developers trust the Elasticsearch platform to confidently build with speed, scale, and relevance.

- API clients for the most widely used programming languages
- Flexible ingestion for unifying all of your data sources, including a comprehensive web crawler
- A framework to build search interfaces with just a few lines of code
- Data enrichment with ML inference pipelines, enhancing natural language processing tasks
- Advanced retrieval including lexical and vector search, hybrid ranking with RRF, and out-of-the-box semantic search
- Machine learning model flexibility to bring your own proprietary model or a third-party model that can be managed within Elastic
- Integration with generative AI for more accurate, domain-specific results based on real-time proprietary data
- Built-in search analytics and visualization tools for creating dashboards and doing deeper analysis
- An extensive global developer community
- Hybrid, multi-cloud, and serverless deployment flexibility

See how search solutions from Elastic can help you drive customer success and provide stellar customer support.

Learn more

elastic.co/enterprise-search/customer-support →