Lucidworks

Enterprise Search Buyer's Guide

It's not a search tool.
It's a productivity power-saw.

Executive Overview

Enterprise search is to corporate intranets what Google is to the internet at large, where most people no longer use homepages and have largely abandoned bookmarks. Instead, they simply search for what they need. Thanks to the proliferation of powerful platforms like Amazon, users now expect equivalent functionality at work.

Companies need to evolve with the needs and expectations of their employees. A typical knowledge worker needs to search in multiple disconnected applications to find answers to questions, spending 20% of their time just looking for information they need to do their jobs. Bad search is a drain on productivity and a frustrating experience for all, but when deployed correctly, search functionality can dramatically increase business productivity.

These days, companies need more than just another search tool. This buyer's guide will walk you through the key features to consider when choosing an enterprise solution platform. You'll learn the must-have functionality that is core to any search application, as well as more advanced capabilities that many organizations need in order to have an effective enterprise search experience.

The search market consists of leading vendors, cloud solutions, legacy solutions, application-specific search and open-source toolkits.

Comprehensive platforms like Lucidworks Fusion allow for cloud, on-premises and hybrid installations. These platforms lead the way in personalization and machine learning technologies that come as part of a complete solution, including A/B testing as well as user and operational monitoring. These solutions cover a complete set of search needs, including enterprise search, site search, ecommerce search, as well as customer and product 360.

Cloud solutions move operational concerns out of the purview of the customer and vary in comprehensiveness or specificity. Some of these solutions are quite complete, but many require additional systems for data ingestion or user profiling capabilities. Some of these solutions only search CRM tools or are limited to a small set of cloud-only sources. These are a great way to handle specific business problems and may offer a form-fit solution to a given organizational pain point.

Application-specific search includes the search built into services like SharePoint or Salesforce, but also includes third-party solutions such as Coveo or Attivio, which focus strictly on one or a limited set of data sources. Application vendors often do a poor job of providing search, and while third-party solutions can be helpful, application-specific search frequently misses the context and introduces inefficiency where a user has to search multiple systems to put together the "bigger picture" or find exactly what they were looking for. Smaller organizations generally rely on a set of application-specific search solutions. Legacy solutions include those from vendors that no longer exist like FAST or Verity. Some of these solutions are still offered from their acquiring vendor, but few have kept up with recent technology. This problem pervades the solution from providing poor search results, trouble connecting to new data sources, and not scaling for modern use. Most large organizations are replacing legacy solutions.

Open source toolkits like Elasticsearch or Apache Solr are widely deployed. While Solr, provides excellent scalability and performance, it doesn't provide cutting edge AI and personalization features. It also requires a lot of custom scripts and operations in order to deploy and maintain effectively. While some of the largest search deployments in the world use Solr, these are primarily managed and developed by technology firms. Most larger enterprises want complete solutions that handle a variety of search and operational concerns and don't want to have to develop it themselves.

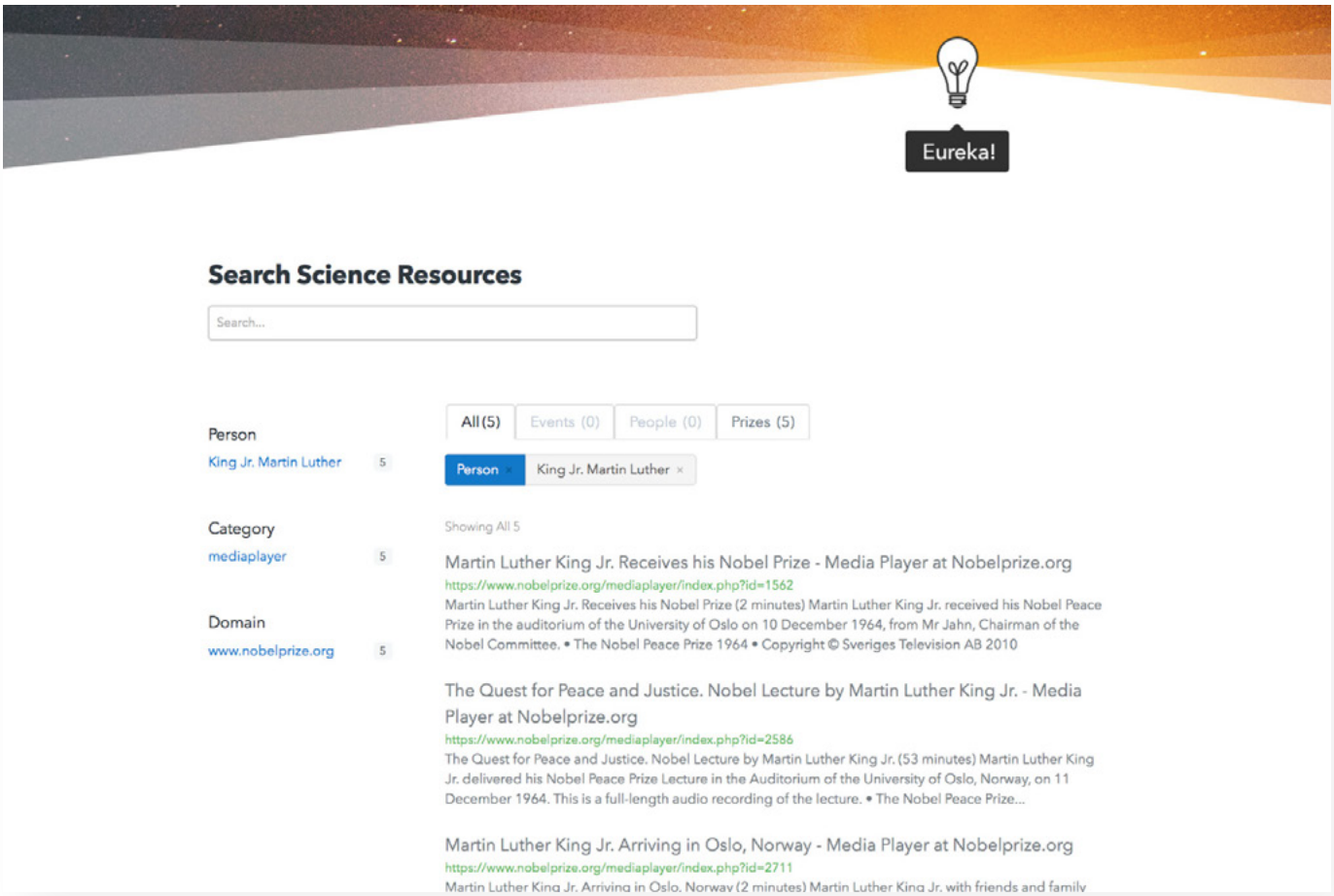
While most search solutions fall into these categories, not all are created equal. Key differentiators include their core architecture and how scalable the solution is. Additionally, some vendors are spending a lot of R&D on personalization and AI technologies, while others are moving to specialize more in specific areas of domain search (i.e. CRM search or log analysis).

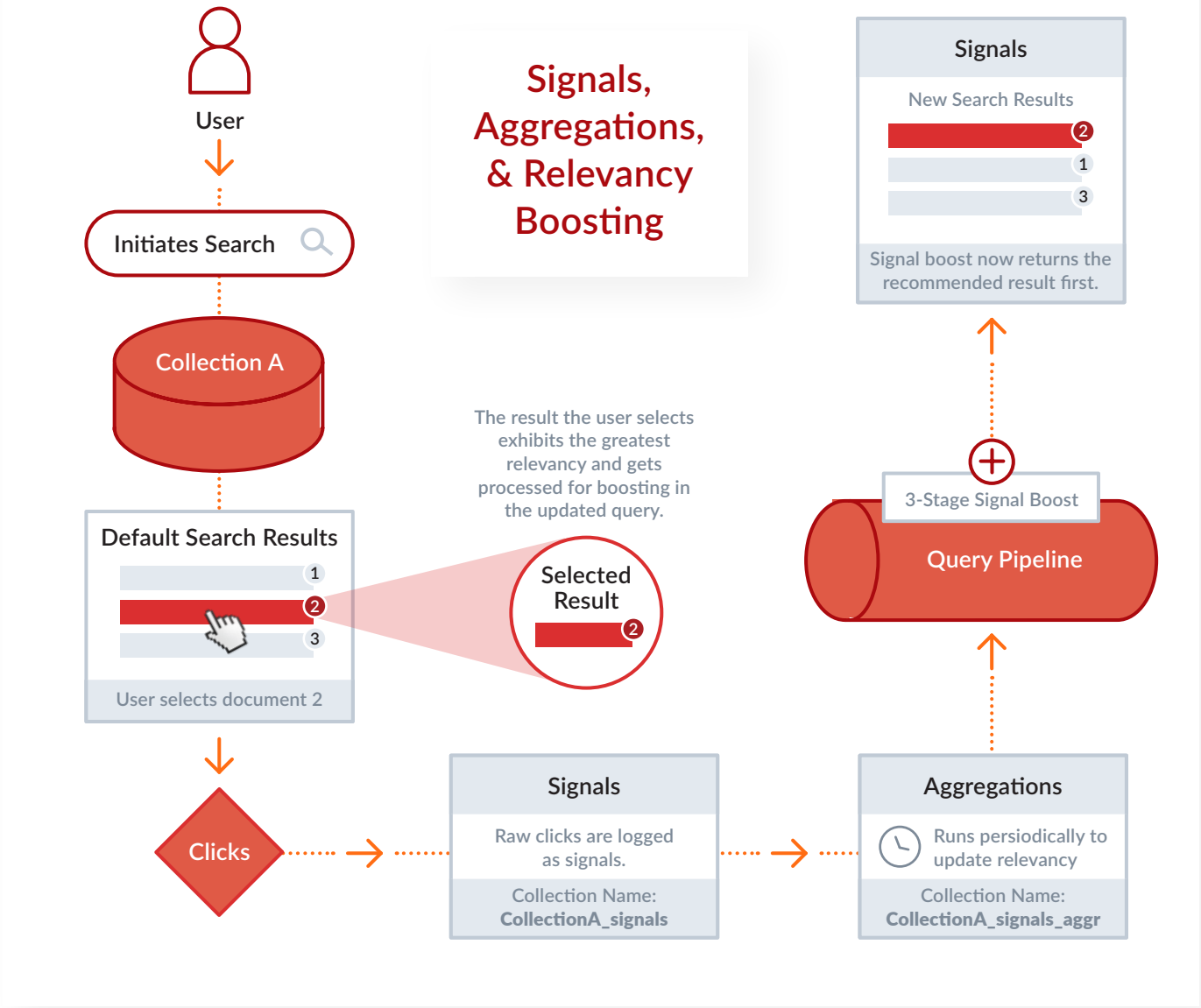
Search Capabilities

Keywords and key phrases are one of the most familiar components of the search experience. A query parser receives a query and translates it into parameters for a search algorithm, which is executed across a specialized database. Older search technologies required a specific syntax where a user would explain in a specialized language exactly what they wanted. Modern search parsers such as Solr's DisMax parser work without this specificity. Advanced queries such as geospatial search still require specialized syntax. Most of the time, this specialized syntax is computer-generated by a search application built to speak to the search engine.

Faceting is a critical capability that allows the user to filter results efficiently based on a specific field. This is especially useful for limiting a search to a category or department. Faceting can also be used for range filtering like date or price. Most search solutions allow you to define synonyms. This capability enables implementers to adapt user queries to their corpus of data. For example, implementers may define "lawyer" as a synonym for "barrister." That will allow searches for "lawyer" to return documents that also might contain the term "barrister."

Signal capture is the capture of user behavior data or "signals." In most search applications, signals include events such as user queries, clicks, adds, purchases and other similar clickstream data. However, in advanced uses, this may include user location, vector, altitude or any number of event type data. Signal capture is a collaboration between the search application, the back-end search solution and the front-end search application. The application captures the actual click and sends it to the search solution, which processes the event and stores it for later use. Newer features, such as recommendations, depend on signal capture.



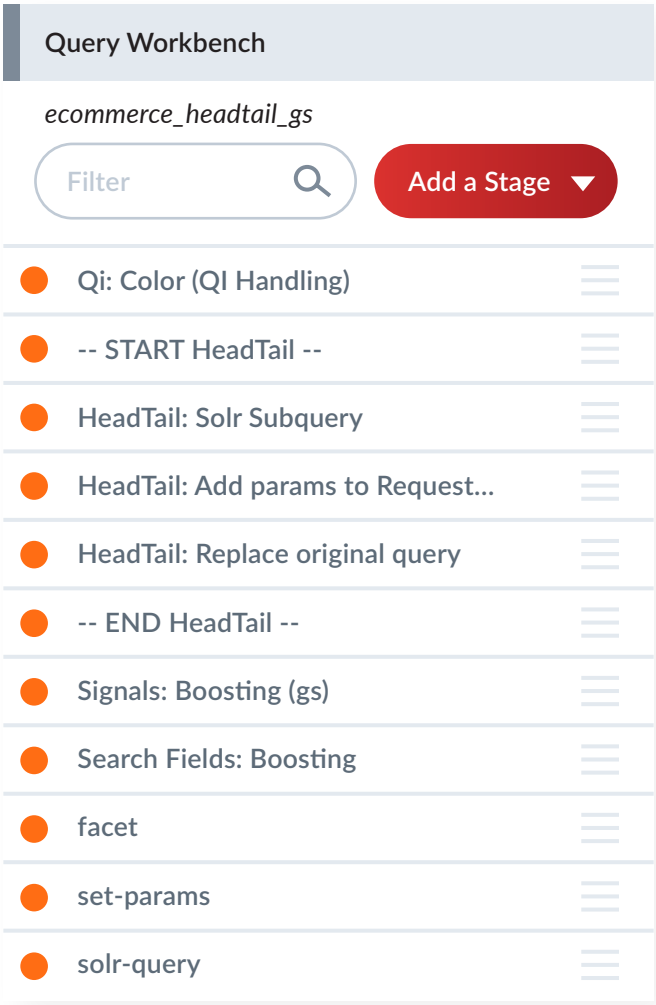


Artificial intelligence and **recommendations** are newer techniques for finding users' relevant data. Not all data lends itself perfectly to keyword or key phrase search. For example, the most relevant result for a query may not be the one that contains the matching keyword. One way of solving this problem is by looking at what users clicked on most often for that query. However, that may not be enough. Context is important, and the user is important. Instead of showing results that most users clicked on, it may be more relevant to show what similar users clicked on. There are numerous algorithms and techniques for using user behavior data to help users find exactly what they need, even before they know they need it. This is the next frontier of enterprise search.

Query pipelines allow implementers to change queries in stages and to change the data that is returned. The stages approach is a critical tool to use in order to handle complex data or complex queries or to provide behavior profiling functionality. Some solutions offer prepackaged functionality as well as extensibility using JavaScript or other programming languages.

UI Capabilities

A powerful search back-end is important, but users only see the front-end user interface (UI). Previously, some search solutions left this entirely to the implementer, but now some offerings are providing UI functionality that allows implementers to import or compose their UI instead of having to create it from scratch. This makes a lot of sense given most search UIs do similar things. Moreover, given that AI and personalization functionality often collaborate with UI, modern UIs are too complex to write and maintain by hand without a larger staff of experts.



WYSIWYG embedding is the most advanced functionality in the marketplace. This allows implementers to configure a search UI in a web-based administration tool and then "include" it on their site using HTML or JavaScript statements.

Smart panels combine back-end functionality, like recommendations or similarity search (aka "More Like This"), with UI components. These allow implementers to include this functionality in their UI without having to write the underlying UI or back-end handling code. In cloud-based solutions, these come with preconfigured, back-end implementations.

Component libraries provide common UI functionality, such as typeahead. These exist in many forms from tag libraries or JavaScript APIs.

Although technically a back-end component, **REST connectivity** is critical to any modern search UI. Most mobile and web UIs connect via JSON over a REST interface.

Typeahead and other forms of auto-suggest are now standard user expectations in search UI. As users type, the UI suggests what they're likely to be interested in.

Auto-classification is a more advanced form of typeahead. For example, when a user types "speaker," and then "audio electronics" is automatically selected as a category.

Data Import

Data source connectors are an important piece of most search solutions. While REST APIs allow users to import from nearly any data source, having to write a connector to every common data source (i.e., Oracle, SQL Server or SharePoint) is a taxing endeavor for implementers. The most important question isn't whether the solution supports the most connectors but whether the solution supports your data sources and any you are likely to deploy.

Parsers work in conjunction with data source connectors to process the data that comes back and turn it into documents. For example, if you're scanning a local disk and pulling back files, should each file be loaded as a document or each row in the file? Is the document an XML or a CSV or a ZIP file? Parsers interpret the data into documents so that they can be further processed or indexed.

Pipelines are used to connect data sources, parsers and stages of logic used to manipulate data into well-formed documents.

JavaScript is the modern scripting language that serves as a kind of "language of trade" for most developers. Because of that familiarity, some search solutions allow manipulating data with JavaScript.

A REST API allows operation control of the search solution as well as importing and exporting data.

Native libraries allow a search solution to bind into a system language like Java or Python without the implementer having to write REST API glue code.

```
HttpSolrServer solrServer + new HttpSolrServe(url, client);
solrServer.setParser(new XMLResponseParser());
ModifiableSolrParams solrParams = new ModifiableSolrParams();
solrParams.add(*"q", "*:*");
QueryResponse out = solrServer.query(solrParams);
System.out.println("QTime is " + out.getQTime());
System.out.println("RH is " + out.getResponseHeader().toString());
```

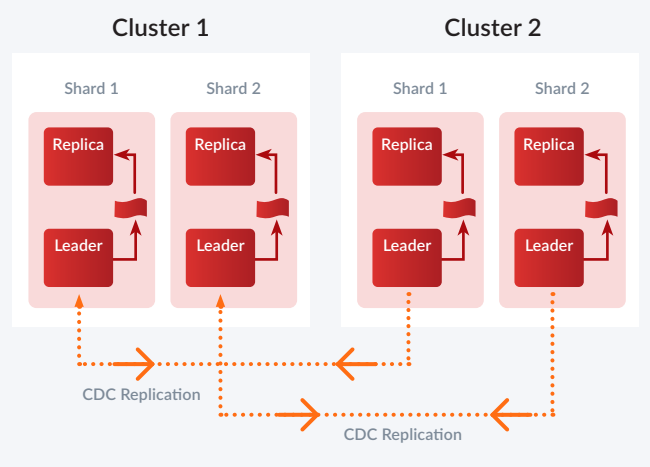
Operational Capabilities

Scalability and **capacity** are important differentiators among search solutions. Can the system scale to the number of documents and users your system needs to support? How hard is it to add additional capacity and can that be done without significant downtime? Some solutions still use client-server architecture or rely on older computing technologies like shared file systems (NAS) instead of modern clustering topologies.

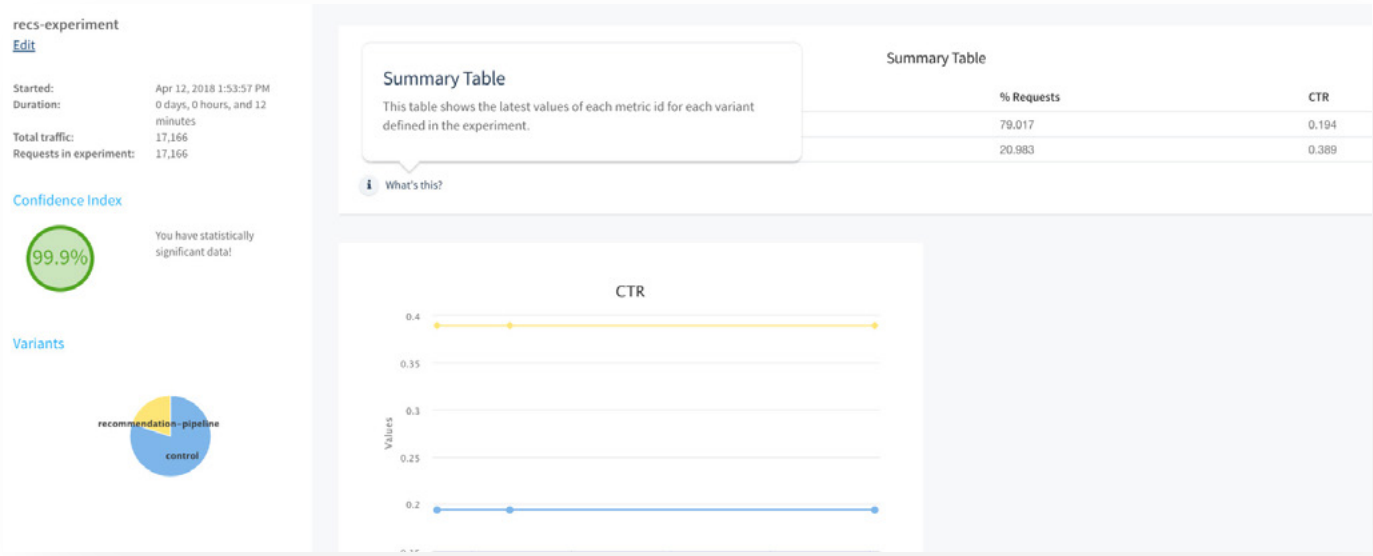
High availability is the capability of the system to suffer a hardware failure without data loss or downtime. This is a feature of a modern cluster topography. Modern search solutions should be resilient against network outages and multiple nodes of hardware faults.

Disaster recovery is the capability of the system to suffer the loss of a complete cluster or data center and failover to a backup site. This requires cross-data-center replication (also called WAN replication). This is important to deal with fiber cuts, weather, earthquakes or other unforeseen catastrophes without a major impact to business operations.

System monitoring is provided by modern search solutions in the form of REST APIs that provide statistics about system performance and uptime, including graphical displays and dashboards.



A/B testing shows admins whether changes to search pipelines or other functionalities improve search performance for users. When applying personalization, recommendations or other AI techniques, it is important to determine whether the changes actually improved click-through rates, purchases or any specified measure of success. A/B testing works by directing some traffic to the new pipeline and comparing it to the original configuration.



Security

Connectivity to major security technologies like Active Directory, LDAP, Kerberos and SAML or other single sign-on systems is critical to a search solution's security capabilities.

Role-based security authorization to determine which users are allowed to delete, read, modify or create documents as well as enact system changes.

Document-level security methods like security trimming to allow for fine-grained control to ensure that users don't see documents they don't have access to as a query result.

Analytics Capabilities

Usage analytics allow implementers to inspect how users interact with search and may even allow inspecting the actions of an individual customer. This capability allows implementers to understand how well they're achieving their conversion goals and see changes in these metrics over time.

SQL connectivity allows analysts to chart and use data with common SQL tools. Solutions may make data as well as user behavioral data available via SQL.

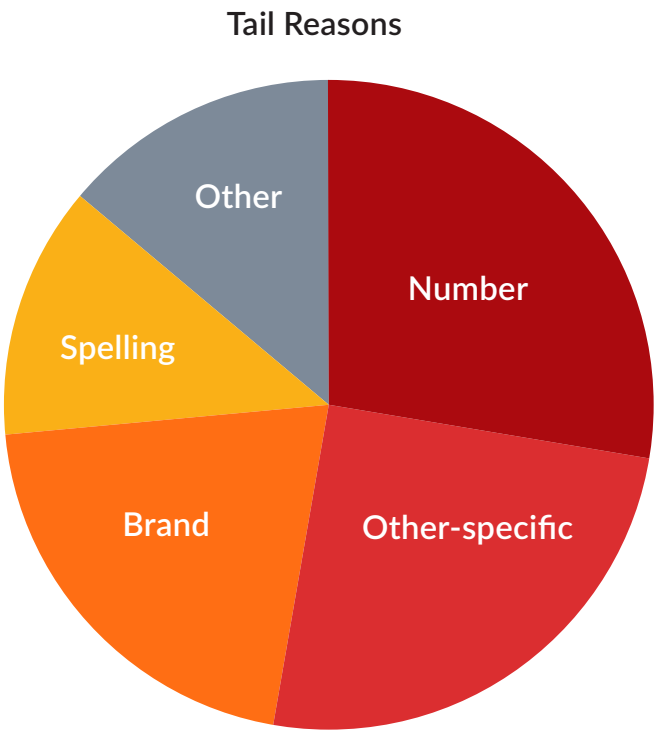
Advanced Functionality

Streaming allows the solution to operate differently than normal. Usually, search solutions return the most relevant items first, but computing this is memory and resource intensive. Streaming allows results to be returned in the order they are retrieved and can return results based on conditions.

Named entity recognition (NER) is the use of natural language processing (NLP) to recognize the names of companies, individuals or other proper nouns. This can be useful for various types of filtered or faceted search.

Clustering and **classification** are machine learning techniques that allow data or queries to be grouped or labeled automatically.

Head/tail analysis is a machine learning technique that identifies and rewrites underperforming queries to be more like similar well-performing queries.



Selection Criteria

We've reviewed many of the criteria to consider when choosing an enterprise search solution for your organization. You've learned what to ask when considering your search solution's storage and scalability needs and how to ensure it works with your company's unique blend of data and documents. You've also become familiar with advanced capabilities like entity recognition, clustering and classification. You're ready to start the vendor selection process for an enterprise search solution.

On the following page, we've included a comparison worksheet. We've left the first column editable to compare solutions to Lucidworks' offerings and help you with your selection. Happy searching!

Get Started or Learn More →

It's time to replace hit-or-miss search with an all-in-one answer platform for data diggers, fact finders and edge seekers everywhere. More than anything, it's time to find out what's possible when information workers have all the insights they need, whenever they need them.

For more information about how Lucidworks Fusion is more than just another search tool, contact us today at **lucidworks.com/contact** or call 415-329-6515.

Enterprise Search Buyer's Guide Comparison Tool

		Lucidworks Site Search	Lucidworks Fusion
Platform Type		Cloud Solution Application-specific search	Comprehensive
Keywords/Key phrase Search		yes	yes
Faceting		yes	yes
Synonyms		yes	yes
Signal Capture		yes	yes
Recommendations		yes	yes
AI		yes	yes
Query Pipelines		yes	yes
WYSIWYG embedding		yes	no
Smart Panels		yes	no
Component Library		yes	yes
REST Connectivity		yes	yes
Auto-classification		no	yes
Data Sources		4	hundreds
Parsers		hundreds	hundreds
JavaScript		no	yes
REST API		yes	yes
Native Bindings		no	yes
Scalability/Capacity		billions of documents millions of users	billions of documents millions of users
High Availability		yes	
Disaster Recovery		no	
System Monitoring		yes	
A/B Testing		no	
Security Sources		n/a	Active Directory, LDAP, Kerberos, SAML, Proxy SSO
Role-based Authorization		no	yes
Document-level/Security Trimming		no	yes
Usage Analytics		no	yes
SQL Connectivity		no	yes
Streaming		no	yes
NER/NLP		no	yes
Clustering/Classification		no	yes
Head-n-Tail Analysis		no	yes