A DEVELOPER’S GUIDE TO BUILDING AI-DRIVEN KNOWLEDGE MINING SOLUTIONS

Unlock actionable insights from all your content with Azure Cognitive Search

Microsoft

Executive Summary

Businesses collect a staggering amount of data every day, primarily in unstructured formats. Across virtually all industries, organizations can realize significant benefits by harnessing and refining the information contained within this raw data. To do so, they need a process for extracting structured data from unstructured content, making it more consumable by business systems, from search to analytics.

Knowledge mining is an emerging category in AI, which refers to the orchestration of a series of AI services to uncover latent insights in vast amounts of data. Knowledge mining solutions make it easier for developers to bring the benefits of AI into their organizations, whether it’s enhancing search functionality in business applications or improving business processes through automation.

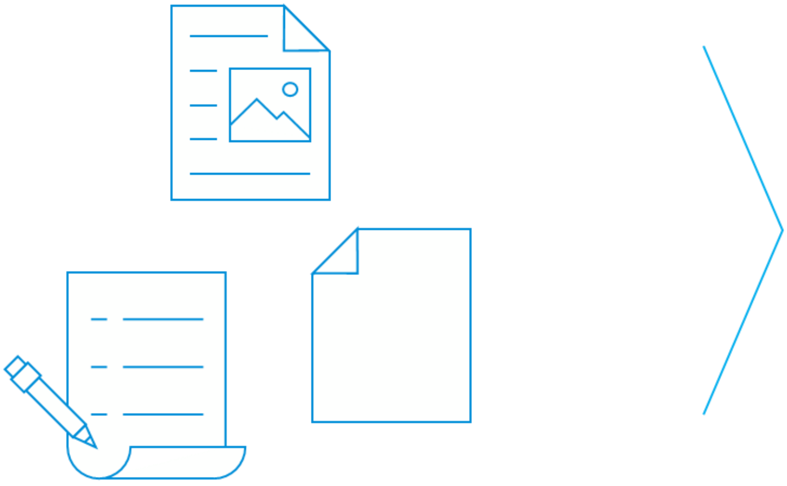
Conceptually, it is easy to see that knowledge mining can help any company needing to explore or surface a large amount of information quickly, repeatedly, and accurately. It can save hundreds of manual hours, enabling users to make informed decisions quickly and turn their attention to higher-value activities. But to get started, developers and business leaders must identify concrete use cases that tie to their business objectives.

In this white paper, we examine how knowledge mining works, describe common ways organizations can use it, and provide an overview of knowledge mining solutions in Azure. By exploring everyday use cases, business leaders and developers alike can find inspiration on how to start using knowledge mining to unlock valuable insights and information hidden within their data.

Contents

Introduction to knowledge mining

Knowledge mining refers to an emerging category of AI designed to simplify the process of accessing the latent insights contained within structured and unstructured data. It defines the process of using an AI pipeline to discover hidden patterns and actionable information from sets of structured and unstructured data in a scalable way.



Using AI to unlock valuable information lying latent in all your content

Your business documents

Knowledge mining solutions improve content understanding by extracting information from siloed structured and unstructured content using a range of pre-trained and custom AI services like computer vision and natural language processing. Using pre-trained models provides developers with access to powerful machine learning capabilities without the need to build and train them. For cases where custom models are needed, knowledge mining solutions also provide the ability to include custom Azure Machine Learning models. This capability provides developers with the flexibility to implement custom AI without having to rethink their whole workflow. Knowledge mining enables users to discover patterns and relationships among previously disparate data points in a variety of channels, including search interfaces, analytics solutions, and other business applications.

Benefits of knowledge mining

For most organizations, a lack of data is no longer a primary challenge. In fact, a recent estimate suggests that society is collectively creating more than 2.5 quintillion bytes of new data every day[1], and the pace of data generation is rapidly accelerating as the adoption of the Internet of Things (IoT) increases. Approximately 80% of the new data produced is unstructured[2], representing things like device telemetry, tweets, Office files, PDFs, images, videos, and audio files, for example.

One of the primary difficulties faced by companies today is how to extract actionable information and business insights from this massive influx of unstructured data. Unlike the structured data traditionally used by organizations for business intelligence, unstructured data does not have a predefined data model, making it more challenging to search and analyze. The volume of data generated compounds this predicament, making it extremely difficult or impossible for human beings to quickly review it to find information or arrive at business decisions. This unstructured data, however, represents an enormous opportunity for companies to gain actionable business insights if it is processed intelligently and promptly.

For organizations to extract value from the immense volume and variety of unstructured content in an acceptable timeframe, it is necessary to rely on the capabilities offered by artificial intelligence (AI). At its core, AI refers to machines mimicking the cognitive functions associated with the human brain. Using AI models, machines can tirelessly perform cognitive tasks like comprehending, perceiving, calculating, organizing, and reasoning, providing the ability to create valuable inferences and insights into vast quantities of data[3]. While pre-trained AI services work well for most use cases, many scenarios require companies to develop custom models tailored to the specific needs of their organization or industry. Creating custom models and stitching those together with pre-training models to thoroughly analyze content is time-consuming and can be prohibitively expensive for all but the largest organizations. Fortunately, the emerging capabilities of knowledge mining simplify the process of accessing the latent insights contained within unstructured data.

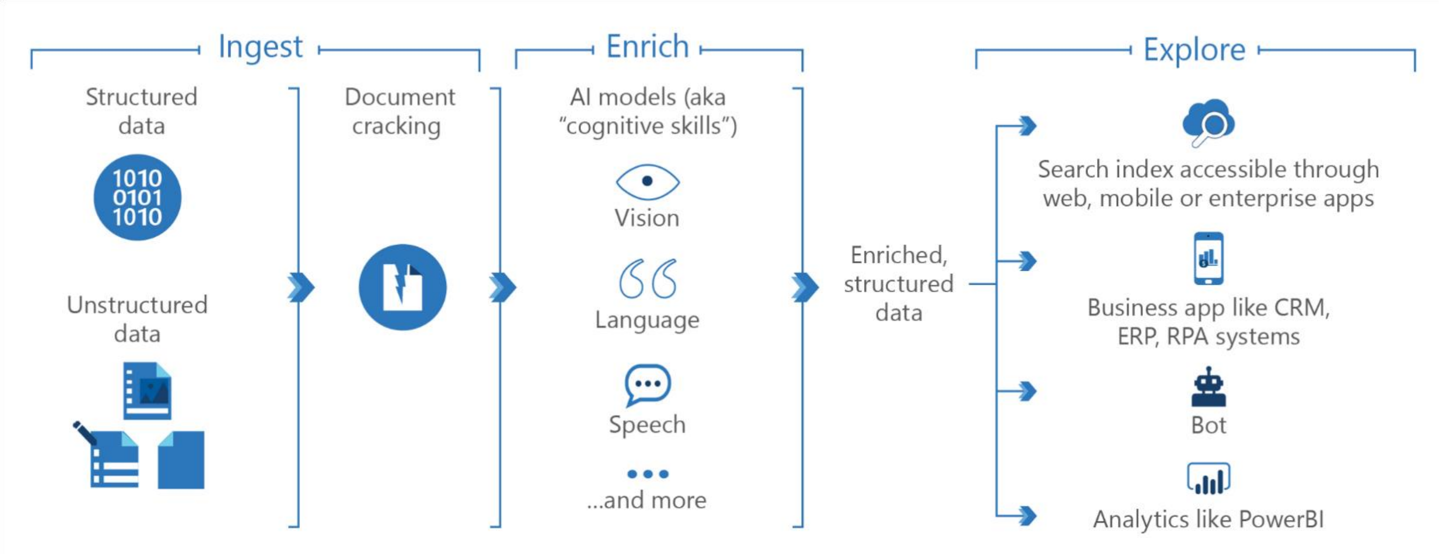
Ultimately, knowledge mining helps stakeholders across organizations and industries find essential needles in haystacks of files, make better-informed decisions, automate redundant business processes, identify risks and opportunities, and much more.

The adoption of AI is changing how business works across all industries. AI is helping companies understand and benefit from their ever-growing stores of data in ways that were not possible in the past. As organizations benefit and profit from customer data, they must ensure the information is used ethically and handled responsibly. However, the increase in the use and collection of unstructured data makes this a challenge for many organizations. Unstructured data, without a defined data model, is inherently more challenging to search and analyze, and therefore, harder to understand and classify.

Implementing knowledge mining solutions can help companies not only gain valuable business insights from their data but also help them identify, classify, and protect sensitive information contained within the data they are collecting. Using AI models, such as the PII Detection cognitive skill offered by Microsoft, allows organizations to automate the process of identifying and protecting sensitive data. The PII Detection skill extracts personally identifiable information from an input text and offers the option to mask it from that text in various ways.

How knowledge mining works

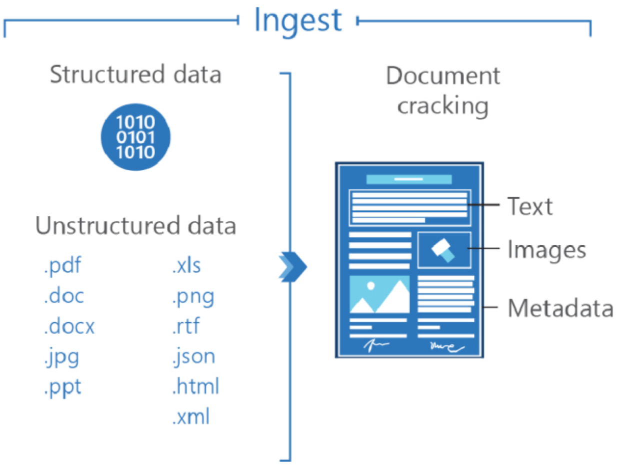
Knowledge mining is the process of applying a series of AI services, in what is known as an enrichment pipeline, to extract information and context from structured and unstructured data. In general, knowledge mining involves the ingestion and enrichment of data, followed by making the newly enriched, structured data available for exploration and analysis.


Ingest
Structured data
1010 0101 1010
Unstructured data
Document cracking
Enrich
AI models (aka "cognitive skills")
Vision
Language
Speech
...and more
Enriched, structured data
Explore
Search index accessible through web, mobile or enterprise apps
Business app like CRM, ERP, RPA systems
Bot
Analytics like PowerBI

Knowledge mining works by orchestrating the overall enrichment pipeline. The first step of a knowledge mining enrichment pipeline is to **ingest** structured and unstructured data from various sources, potentially including internal and third-party data. As part of the ingestion process, the pipeline "cracks" the documents to extract any data contained within them and create a simple structure for the information. Next, the pipeline uses AI to **enrich** the data by analyzing and acting on the extracted information, applying additional structure, finding patterns, and gaining understanding. Finally, the pipeline publishes or exposes the enriched data, allowing search tools, existing business applications, and business intelligence and analytics solutions to **explore and analyze** the newly structured data.

Ingest

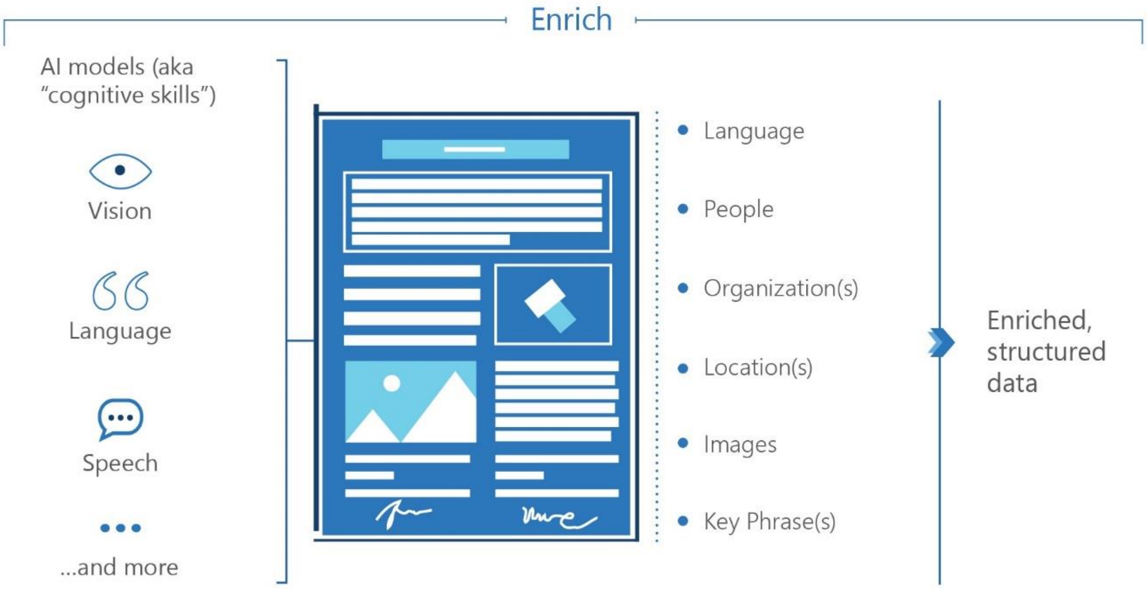
Knowledge mining begins with ingesting data. Ingestion of both structured and unstructured data is possible. Structured data has a defined data model and typically resides in relational databases, such as SQL Server or MySQL. Unstructured data does not have a predefined data model and can come from sources such as NoSQL databases, APIs, blob storage, file stores, and many others. The rapid growth in the amount of unstructured data is what has driven the development of knowledge mining, as its lack of a defined structure makes it more challenging to search and analyze. Many business documents qualify as unstructured data, including images, videos, audio files, PDFs, Word documents, PowerPoint presentations, Excel spreadsheets, emails, web files, rich text formats, and JSON files, among many others.



Data ingestion is the process of aggregating raw data, whether structured or unstructured, from various siloed sources and locations into a persistent, centralized data store. In knowledge mining, the ingested data is typically given a standard structure as part of the import process. This structure allows for the effective use of enriched documents when the enrichment pipeline is complete. This structure is determined based on the information extracted via "document cracking." Document cracking is the process of extracting or creating text content from non-text sources, often using optical character recognition (OCR).

Enrich

Once data has been ingested and cracked, the next step is to enrich the data contained within each document using artificial intelligence. AI enrichment acts on the raw data extracted during cracking to identify patterns, obtain information, and gain understanding from the text contained within images, blobs, and other unstructured data sources.



Knowledge mining performs enrichment on individual documents as a sequence of calls to AI models. Each AI enrichment step can act on the raw text data extracted from the document, as well as on the enrichments added by previous actions in the pipeline. This capability to build upon previous AI enrichments adds powerful capabilities to build upon insights with each subsequent step in a pipeline. AI enrichment models can be either pre-trained or custom models, and pipelines frequently include both. Most enrichment pipelines start by leveraging pre-trained natural language processing and computer vision AI services to uncover valuable information without the need for developers to build custom models.

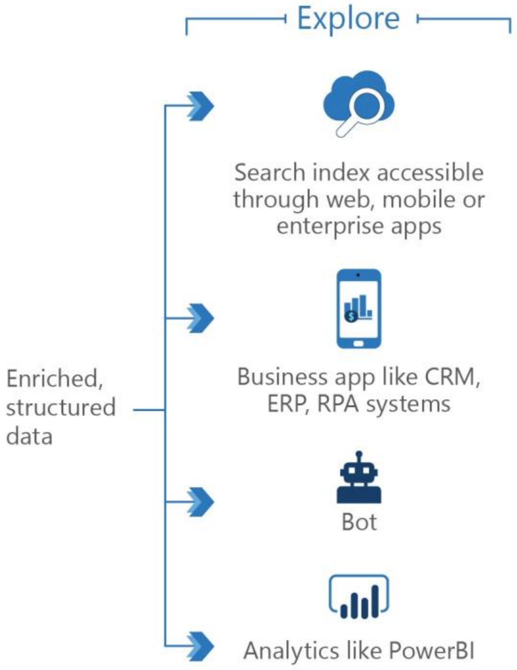
- **Natural language processing** services can understand written and spoken human language. These AI services can interpret sentiment, detect and translate languages, and extract words, key phrases, and the names of people, locations, and organizations.
- **Computer vision** services can analyze images or videos to detect and classify faces, landmarks, celebrities, or other objects. They can also caption images and transcribe handwriting.

While pre-built AI services from technology providers typically provide the basic functionality required to gain valuable insights, many organizations develop custom AI models to get the most out of knowledge mining. Custom models allow developers to integrate customized rules and logic specific to their industry or organization.

The AI enrichment step results in the creation of a collection of enriched documents, with data from each data source conforming to the structure defined by the pipeline and containing the information, insights, and other details added by every AI model in the enrichment pipeline.

Explore & Analyze

When enrichment is complete, the final step is to expose the newly enriched, structured documents, so they are accessible for exploration and analysis. This step might mean adding the documents to a search index or writing them out to a storage location, or often, both.


Explore
Enriched, structured data
Search index accessible through web, mobile or enterprise apps
Business app like CRM, ERP, RPA systems
Bot
Analytics like PowerBI

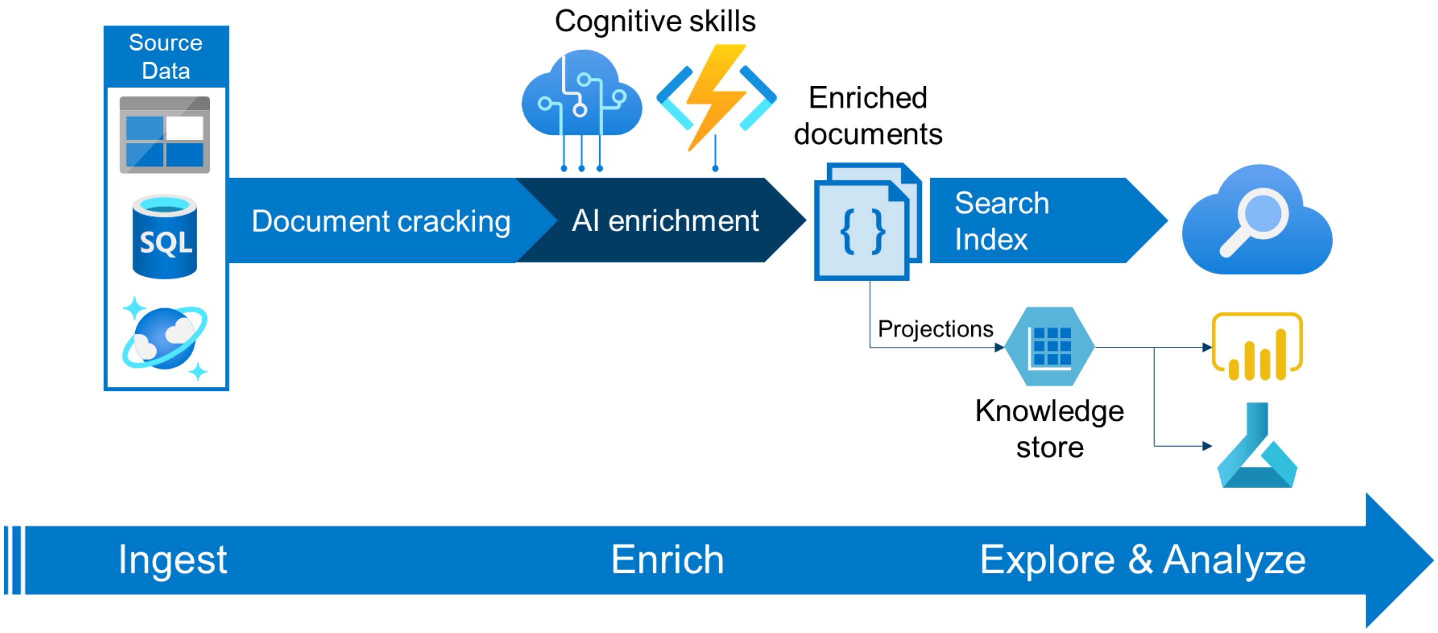
Exploration is the process of reviewing the added enrichments to learn more about your data. To facilitate exploration, organizations typically make the results of enrichment available via search indexes or end-user and line-of-business applications, such as customer relationship management (CRM) or enterprise resource planning (ERP) systems, to name a few. Exploration ordinarily involves a human user searching for and exploring the enriched documents, perhaps looking for relationships between the data in documents or linking documents by keywords.

Analysis usually refers to the application of analytics tools, such as Power BI, Azure Machine Learning, or Azure Databricks, for exploring and gaining a deeper understanding of the enriched data. Analytics tools provide more robust capabilities for gaining insights from the enhanced data. Power BI allows companies to create rich reports and dashboards, which enable consumers to explore the data visually. Azure Machine Learning and Azure Databricks are powerful analytics platforms for doing machine learning, extracting actionable insights, and performing anomaly detection, for example.

Knowledge mining with Azure Cognitive Search

Azure Cognitive Search is a search-as-a-service cloud solution that gives developers APIs and tools for adding a rich search experience over private, heterogeneous content in web, mobile, and enterprise applications. It offers capabilities such as scoring, faceting, suggestions, synonyms, and geo-search to provide a rich user experience. Azure Cognitive Search is also the only cloud search service with built-in knowledge mining capabilities.

Azure Cognitive Search acts as the orchestrator for your knowledge mining enrichment pipeline, following the **ingest**, **enrich**, and **explore and analyze** pattern described above.



Ingest

Data ingestion, or indexing, in Azure Cognitive Search is the process of populating a search index from an external data source. An index is a persistent store of documents and other constructs used for filtered and full-text search on an

Azure Cognitive Search service. When creating an index, you provide a schema that defines the components of the index. Indexes are composed of the following elements:

- The *fields collection* specifies the name, type, and attributes of each field in your index. The fields collection is typically the most substantial part of an index.
- *Suggesters* define which fields in an index are used to support auto-complete or type-ahead queries in searches.
- *Scoring profiles* define custom scoring behaviors that let you influence which items appear higher in search results.
- The *analyzers* element is an optional collection of language analyzers that can be assigned to fields in the index. Analyzers are responsible for processing text in query strings and indexed documents.
- The *CORS* (Cross-Origin Resource Sharing) component specifies a list of sites from which cross-origin queries are allowed. Client-side JavaScript cannot call any APIs by default since the browser prevents all cross-origin requests.
- The *encryption key* element provides the ability for a search index to be encrypted with **customer-managed keys** in Key Vault. By default, all indexes are using Microsoft-managed keys.

Azure Cognitive Search creates physical structures based on the schema you provide. For example, if your index has a field marked as searchable, an inverted index is created for that field. Because physical structures are created in the service, dropping and recreating indexes is necessary whenever you make material changes to an existing field definition. You can create search indexes, and entire search pipelines, using the Azure portal, the .NET SDK, or REST API calls using Postman. Code, rather than a portal approach, is recommended for iterative design. For developers just getting started, however, it can be informative to create a knowledge mining pipeline using the Azure portal and then use Postman to retrieve and inspect each component of the pipeline to understand the structure and relationships between components better. Using Postman and the REST API is the recommended approach for index development. Using Postman allows you to parameterize the API calls, edit the API calls to customize your pipeline, and to share and preserve your changes easily.

Documents in an index are, conceptually, a single unit of searchable data in your index. For example, an e-commerce retailer might have a document for each item they sell, a news organization might have a document for each article, and so forth. Mapping these concepts to more familiar database equivalents: an index is conceptually like a table, and documents are roughly equivalent to rows in a table. All documents in an Azure Cognitive Search index must be in JSON (JavaScript Object Notation) format.

There are two basic approaches used for ingesting data and populating an index in Azure Cognitive Search:

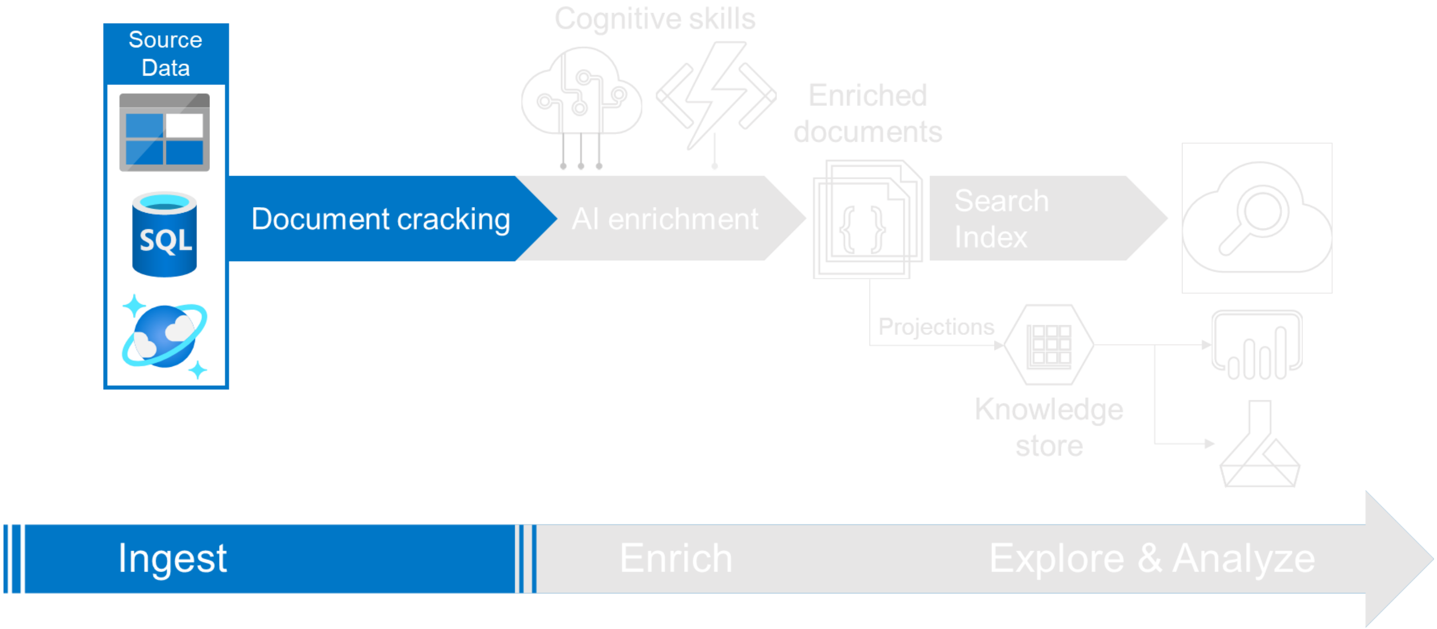
1. *Pull* your data into the index using an Azure Cognitive Search indexer.
2. *Push* data into the index programmatically.

Pull model

The *pull model* uses an indexer to crawl a supported Azure data source and automatically upload data from the source into an index. This approach is sometimes referred to as a *pull model* because the indexer pulls data into the index without a developer having to write any code that adds data to an index.

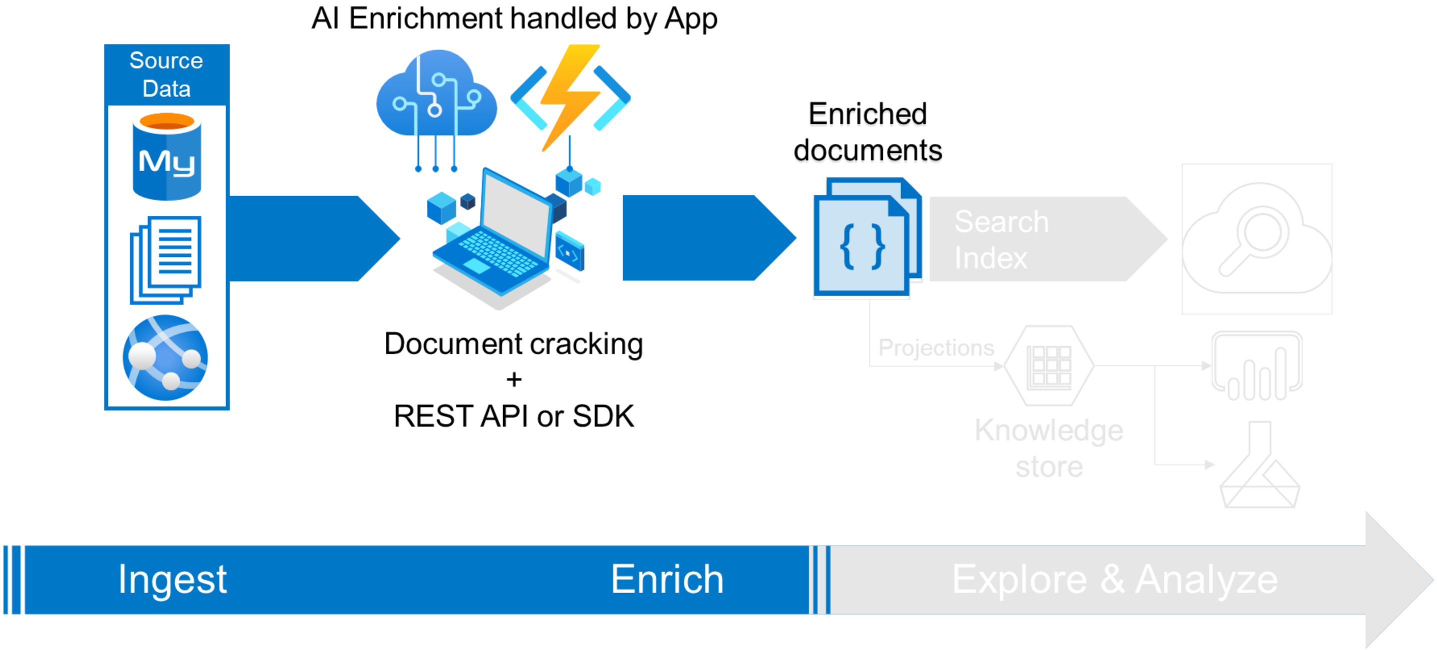
Indexers connect an index to a supported data source and crawl the data source to extract searchable data and metadata to populate the index. Indexers can run on-demand or on a recurring schedule capable of running as often as every five minutes.

When you use the Import data wizard in the Azure portal to create an Azure Cognitive Search index, the pull model is employed. The Import data wizard creates an indexer and uses that to populate the index from the Azure data sources you defined in the wizard. Data must exist in a supported Azure data service that can be accessed by an indexer. The indexer "cracks" source files to extract text and metadata and populates the associated fields within documents in the index.



Push model

The *push model* relies on custom applications to push documents directly into a search index programmatically. Applications can use either the [Azure Cognitive Search REST API]() or the [Azure Search SDK for .NET]() to send data into the index. This model does not use indexers or Azure Cognitive Search data sources, so the application pushing data into the index must perform document cracking, if necessary, and AI enrichment before calling the API or SDK methods to add documents to an index. AI enrichment via the application is typically handled by calling Azure Functions or other endpoints configured to comply with the custom skill interface. The documents passed to the index must be in a JSON format that conforms to the structure defined for the index. The push model also allows you to upload documents to Azure Cognitive Search regardless of where the source data resides since it does not rely on an indexer's ability to access it.



The push model provides a higher level of flexibility when compared to the pull model. First, there are no restrictions on the data source type. Any dataset composed of JSON documents can be pushed to an Azure Cognitive Search index, assuming each document in the dataset has fields mapping to fields defined in your index schema. Second, it has no restrictions on the frequency of execution. You can push changes to an index as often as you like, and are not limited by the five-minute minimum when using indexer scheduling. The push approach allows you to upload documents individually or in batches (up to 1000 per batch or 16 MB, whichever limit comes first). Using the push model, you also have control over the type of indexing action on a per-document basis. You can indicate whether a document should be uploaded in full, merged with existing document content, or deleted.

You can use the following APIs to load documents into an index:

- REST API: Add, Update, or Delete Documents
- .NET SDK: indexAction class or indexBatch class

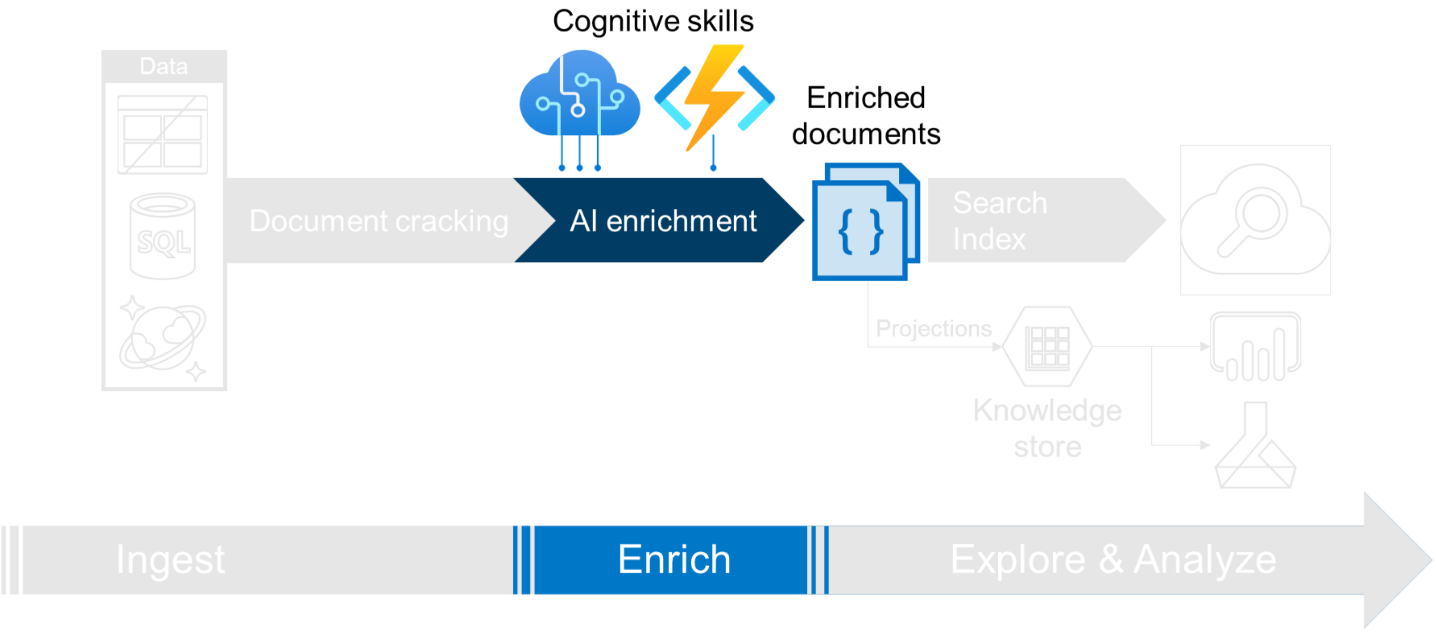
Additional data ingestion resources

- What is an index in Azure Cognitive Search
- Create an index that supports multiple languages
- Indexers in Azure Cognitive Search
- Loading documents into an index with the REST API
- Loading documents with the SDK for .NET
- Indexing large data sets in Azure Cognitive Search

Enrich

AI enrichment defines the built-in knowledge mining capability of Azure Cognitive Search indexing. AI enrichment extracts and enriches content to make it more searchable in an index or knowledge store. The implementation of AI enrichment in Azure Cognitive Search is handled through the attachment of cognitive skills to an indexing pipeline.

A cognitive skill is a module or operation that transforms content in some way. Often, it is a component that extracts data or infers structure, and therefore augments understanding of the input data. Once you have defined a skillset, you must map the output fields of any skill that directly contributes values to a given field in your search index. These *Output Field Mappings* move content from enriched documents into the index.



The collection of cognitive skills included in a pipeline is defined in a skillset, which is a reusable resource in Azure Cognitive Search. Creating a skillset lets you attach text, image, and custom AI enrichment services in the data ingestion phase, extracting and creating new information and structures from raw content. The skillset definition includes an unordered collection of skills, as well as the connection details for your Cognitive Services and Knowledge Store storage accounts. The Azure Cognitive Services platform determines the sequence of skill execution based on the inputs required for each skill.

Skillsets are authored in JSON, and adding advanced functionality, like custom skills, must be done through the REST API or SDK for .NET. You can build sophisticated skillsets with looping and branching using the expression language. It is often easiest to get started by using the import data workflow. You can then view the skillset definition make a REST API call to get the skillset.

The Azure Cognitive Search architecture is extensible, allowing you to assemble an enrichment pipeline from both predefined and custom cognitive skills.

Predefined skills

The predefined cognitive skills in Azure Cognitive Search leverage pre-trained Computer Vision and Text Analytics machine learning models, accessible through the Microsoft Cognitive Services APIs. The cognitive skills built into Azure Cognitive Search fall into two categories:

- **Natural language processing**: These skills include entity recognition, language detection, text translation, key phrase extraction, text manipulation, sentiment detection, and PII detection. With these skills, unstructured text can assume new forms, mapped as searchable and filterable fields in an index.
- **Image processing**: These skills include Optical Character Recognition (OCR) and identification of visual features, such as facial detection, image interpretation, image recognition (famous people and landmarks), or attributes like colors or image orientation. You can create text-representations of image content, searchable using all the query capabilities of Azure Cognitive Search.

The predefined skills, if included, are applied during data ingestion. The results become part of a document's composition in the search index.

Custom skills

Custom skills provide a way to insert transformations unique to your content. A custom skill executes independently, applying whatever enrichment step you require. For example, you could define field-specific custom entities, build custom classification models to differentiate between business and financial contracts and documents, or add a speech recognition skill to reach deeper into audio files for relevant content.

To provide a concrete example, suppose you want to create a custom skill that extracts the first date mentioned in the text of contract documents. The skill accepts a single input *contractText* and returns a single output *contractDate*.

Sou

There is a simple and clear Web API custom skill interface for connecting custom skills to an enrichment pipeline. The only requirement for inclusion in a skillset is the ability to accept inputs and emit outputs in ways that are consumable within the skillset as a whole. The Web API input format must accept an array of records to be processed, with each record containing a record ID and a “property bag” that is the input provided to your Web API. You define the array of records as a *values* array, with each member representing the input for a particular record. Each record in the *values* array is required to have the following elements:

- A *recordId* member serves as the unique identifier for a particular record. When your enricher returns the results, it must provide this *recordId* to allow the caller to match the record results to their input.
- A *data* member that is the “property bag” of input fields for each record.

In our contract date enrichment example, your Web API should expect request input to look similar to this:

```
{
  "values": [
    {
      "recordId": "a1",
      "data": {
        "contractText": "This contract was issued November 3, 2017 and involves..."
      }
    },
    {
      "recordId": "b5",
      "data": {
        "contractText": "In the City of Seattle, WA on February 5, 2018 there was..."
      }
    },
    {
      "recordId": "c3",
      "data": {
        "contractText": null
      }
    }
  ]
}
```

The Web API output format follows the same pattern. It must contain a set of records containing a *recordId* property and a “property bag” named *data*. The custom skill for the contract date enrichment example returns a single output, *contractDate*, which is in the shape of a multi-part complex type. The output from your Web API should look like:

```
{
  "values": [
    {
      "recordId": "b5",
      "data": {
        "contractDate": {
          "day": 5,
          "month": 2,
          "year": 2018
        }
      }
    },
    {
      "recordId": "a1",
      "data": {
        "contractDate": {
          "day": 3,
          "month": 11,
          "year": 2017
        }
      }
    },
    {
      "recordId": "c3",
      "data": {},
      "errors": [
        {
          "message": "contractText field required "
        }
      ],
      "warnings": [
        {
          "message": "Date not found"
        }
      ]
    }
  ]
}
```

Azure Functions makes creating custom skills simple, although they are not the only way to create a custom skill. As long as your API endpoint meets the interface requirements for a cognitive skill, the approach you take is immaterial. Example Azure Function code for the date extractor example above would look similar to the following:

```
[FunctionName("DateExtractor")]
public static async Task<IActionResult> Run(
    [HttpTrigger(AuthorizationLevel.Function, "post", Route = null)] HttpRequest req,
    ILogger log)
{
    log.LogInformation("Date Extractor function: C# HTTP trigger function processed a request.");

    var response = new WebApiResponse
    {
        Values = new List<OutputRecord>()
    };

    string requestBody = new StreamReader(req.Body).ReadToEnd();
    var data = JsonConvert.DeserializeObject<WebApiRequest>(requestBody);

    // Do some schema validation
    if (data == null)
    {
        return new BadRequestObjectResult("The request schema does not match expected schema.");
    }
    if (data.Values == null)
    {
        return new BadRequestObjectResult("The request schema does not contain a values array.");
    }

    // Calculate the response for each value.
    foreach (var record in data.Values)
    {
        if (record == null || record.RecordId == null) continue;

        OutputRecord responseRecord = new OutputRecord
        {
            RecordId = record.RecordId
        };

        try
        {
            responseRecord.Data = ExtractFirstDate(record.Data.ContractText).Result;
        }
        catch (Exception e)
        {
            // Something bad happened, log the issue.
            var error = new OutputRecord.OutputRecordMessage
            {
                Message = e.Message
            };

            responseRecord.Errors = new List<OutputRecord.OutputRecordMessage>
            {
                error
            };
        }
        finally
```

To see complete example code, including input and output objects and other methods called, view the Create a custom skill for Azure Cognitive Search article. You can also use the Azure Search Power Skills GitHub repo as a place to get started building custom cognitive skills.

In addition to updating your pipeline's skillset when adding a custom skill, the index and indexer also need to be updated. The output fields for the custom skill must be added to the index, and any required field mappings must be included in the indexer. Use the REST API or SDK for .NET to make these updates.

Incremental enrichment

Incremental enrichment is a new feature in Azure Cognitive Search that adds caching and statefulness to an enrichment pipeline. These additional capabilities help to preserve your investment in existing output while changing only those documents impacted by particular modification.

Incremental enrichment adds a cache to the enrichment pipeline, implemented as a container in an Azure Storage account. The indexer caches the results from document cracking plus the outputs of each skill for every document. When a skillset is updated, only the changed, or downstream, skills are rerun. Updated results are written to the cache, and the document is updated in the search index or the knowledge store.

Additional enrichment resources

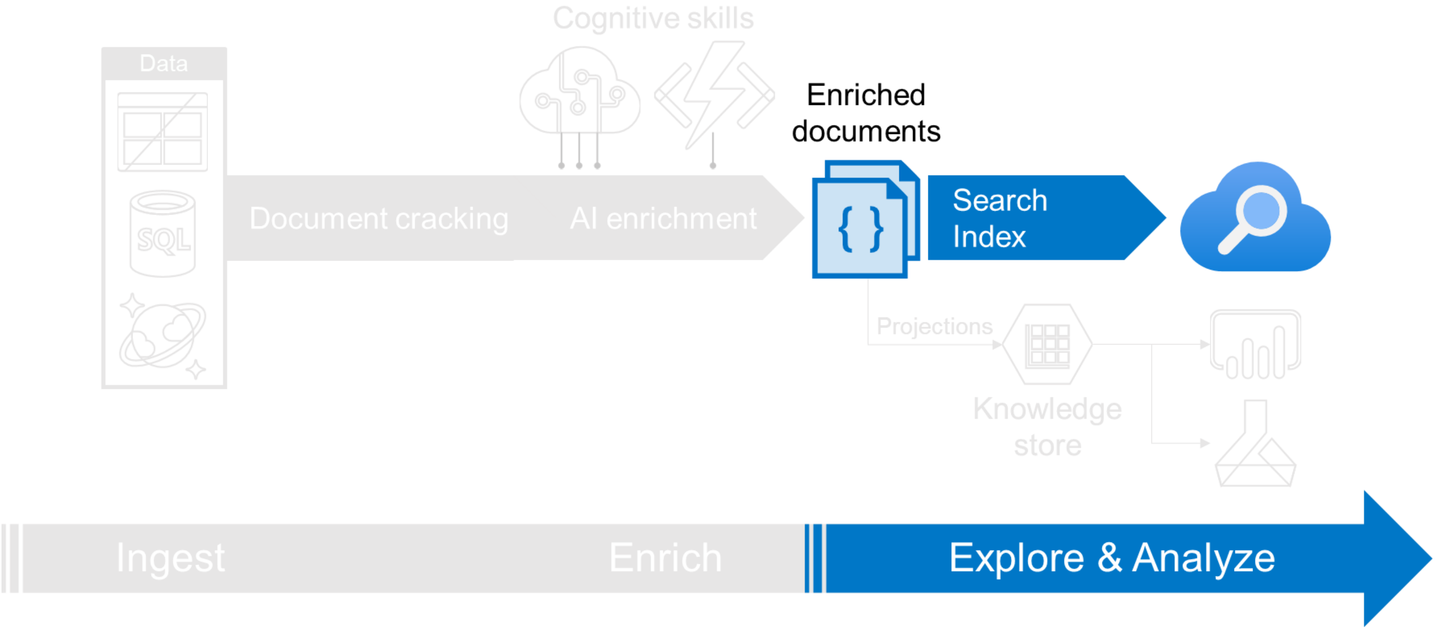
- Skillset concepts and composition in Azure Cognitive Search
- Create a skillset using the REST API
- Example: Create a custom skill
- Introduction to incremental enrichment
- Tips for AI enrichment in Azure Cognitive Search

Explore & Analyze

The output from a knowledge mining enrichment pipeline in Azure Cognitive Search can be a search index or projections in a knowledge store. Both outputs are products of the same pipeline, derived from the same inputs, but resulting in an output that is structured, stored, and used in very different ways.

Querying the search index

In Azure Cognitive Search, a query is a full specification of a round-trip operation. Parameters on the request provide match criteria for finding documents in an index, which fields to include or exclude, execution instructions passed to the engine, and directives for shaping the response.



Index and query design are tightly coupled in Azure Cognitive Search. An essential fact to know up-front is that the index schema you define, with attributes on each field, determines the types of queries you can build against those fields. The index attributes assigned to a field specify which operations are allowed. Index attribute options include the ability to specify whether a field is searchable, sortable, retrievable, filterable, or facetable.

Azure Cognitive Search sits on top of Apache Lucene and gives you a choice between two query parsers for handling typical and specialized queries.

- Simple query syntax
- Full Lucene query syntax

Requests using the simple parser are formulated using the simple query syntax, selected as the default for its speed and effectiveness in free form text queries. The full Lucene query syntax, enabled when you add *queryType=full* to the request, exposes the widely adopted and expressive query language developed as part of Apache Lucene. Full syntax extends the simple syntax, so any query you write for the simple syntax runs under the full Lucene parser.

In Azure Cognitive Search, query execution is always against a single index. You cannot join indexes or create custom or temporary data structures as a query target. Queries against the index are authenticated using an API key provided in the request. Query results are returned as JSON documents.

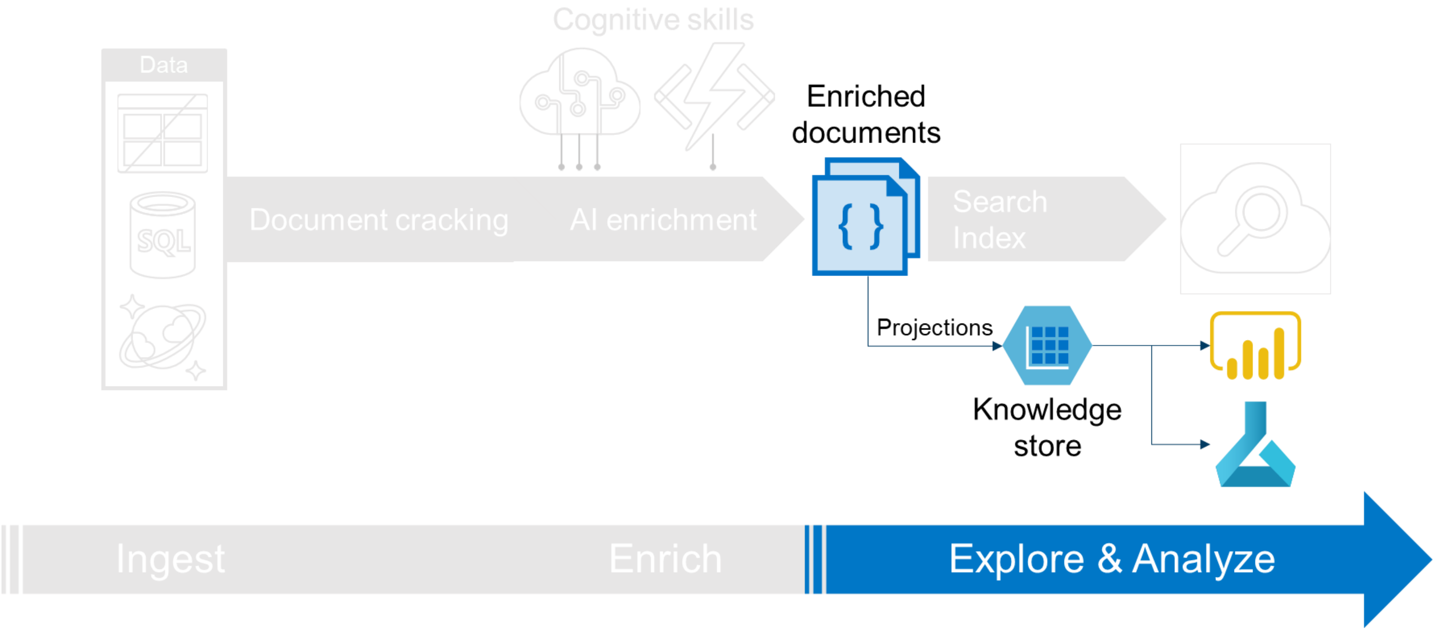
You can shape results by setting parameters on the query, selecting specific fields for the response, and using filters. Parameters on the query can be used to structure the result set in the following ways:

- Limiting or batching the number of documents in the results (50 by default)
- Selecting fields to include in the results
- Setting a sort order
- Adding hit highlights to draw attention to matching terms in the body of the search results

Filters provide criteria for selecting documents used in an Azure Cognitive Search query. Unfiltered search includes all documents in the index. A filter scopes a search query to a subset of documents. For example, a filter could restrict full-text searches to just those products having a specific brand or color, at price points above a certain threshold. Filters can be used anytime you want to constrain search using value-based criteria.

Knowledge store

Knowledge store is a new feature of Azure Cognitive Search that persists output from an AI enrichment pipeline for independent analysis or downstream processing. An enriched document is a pipeline's output, created from content that has been extracted, structured, and analyzed using AI services. In a standard AI pipeline, enriched documents are transitory, used only during indexing, and then discarded. The knowledge store preserves the enriched documents, allowing you to take the structure, context, and actual content gleaned from your AI enrichment pipeline and make it available for downstream processes like machine learning and data science workloads.



Enriched documents are expressed as projections in a knowledge store. Projections are views of enriched documents that can be saved to physical storage for knowledge mining purposes. A projection lets you "project" your data into a shape that aligns with your needs, preserving relationships so that tools like Power BI can read the data with no additional effort. Projections can be articulated as tables, objects, or files.

To create a projection, you must shape the data using either a Shaper skill to create a custom object or use the inline shaping syntax within a projection definition. A data shape contains all the data you intend to project, formed as a hierarchy of nodes. Reshaping allows you to define a projection that aligns with your intended use of the information while preserving relationships. Projections can be tabular, with data stored in rows and columns in Azure Table storage, or JSON objects stored in Azure Blob storage. You can define multiple projections of your data as it is being enriched. Multiple projections are useful when you want the same data shaped differently for individual use cases.

For example, table projections should look like the following:

```
{
  "name": "your-skillset",
  "skills": [
    "your skills"
  ],
  "cognitiveServices": {cognitive services key info},
  "knowledgeStore": {
    "storageConnectionString": "an Azure storage connection string",
    "projections": [
      {
        "tables": [
          {
            "tableName": "MainTable",
            "generatedKeyName": "SomeId",
            "source": "/document/EnrichedShape"
          },
          {
            "tableName": "KeyPhrases",
            "generatedKeyName": "KeyPhraseId",
            "source": "/document/EnrichedShape/*/KeyPhrases/*"
          },
          {
            "tableName": "Entities",
            "generatedKeyName": "EntityId",
            "source": "/document/EnrichedShape/*/Entities/*"
          }
        ]
      },
      {
        "objects": []
      },
      {
        "files": []
      }
    ]
  }
}
```

Physically, a knowledge store is Azure Storage, either Azure Table storage, Azure Blob storage, or both. Any tool or process that can connect to Azure Storage can consume the contents of a knowledge store. You can connect to use using tools like Power BI or Storage Explorer to explore and analyze the enriched documents.

Additional exploration and analysis resources

- Search query overview
- Simple query syntax for Azure Cognitive Search
- Lucene query syntax in Azure Cognitive Search
- Using filters in Azure Cognitive Search
- How to build a facet filters
- How to work with search results in Azure Cognitive Search
- Introduction to knowledge stores
- Overview of projections in a knowledge store

- Create a knowledge store using REST and Postman
- How to shape and export enrichments into knowledge store projections
- How to connect to knowledge store with tools and apps

Use Microsoft Azure to jump start your knowledge mining solution

The Microsoft Azure AI knowledge mining portfolio is an industry-leading collection of AI services. Azure Cognitive Search is the only cloud search service with built-in knowledge mining capabilities. It uses a state-of-the-art natural language stack created by Microsoft researchers and native integration with Microsoft Cognitive Services to turn raw unstructured information into searchable content.

Most enterprises have a wide variety of data sources, storing both unstructured and structured data. Azure Cognitive Search reduces the complexity of data ingestion and index creation by integrating with either popular Azure storage solutions or any other data sources and offering index functionality exposed through a simple REST API or .NET SDK.

To unlock the undiscovered knowledge from the content stored in disparate data sources, organizations typically need more than one AI model or API. Azure solutions offer end-to-end tools that reduce the need for developers to stitch a variety of services together. With built-in Microsoft Cognitive Services APIs, Azure Cognitive Search provides access to a broad selection of AI services and the ability to integrate and orchestrate them easily. These services offer not only breadth but also help companies deploy knowledge mining solutions quickly without the need to buy data and train models internally.

Azure Cognitive Search also supports the integration of custom AI models so that developers can build and integrate AI models specifically tailored to their business or industry, such as legal clauses, industrial parts, or pharmaceutical terms. Users can plug in an existing model or build a new one using Azure Machine Learning or Azure Functions in any framework or language (TensorFlow, Python, etc.). Users can also adjust search results using custom-tuned ranking models that tie search results to business goals.

There are several ways to surface search results, including web, enterprise applications, or a bot interface. To leverage the AI enriched documents outside of an Azure Cognitive Search index, a new feature called Knowledge Store makes it possible to project the index documents into tabular or object stores. Data projected into the knowledge store can be analyzed with tools like Power BI or used to train machine learning models in Azure Machine Learning or Azure Databricks.

Azure Cognitive search provides a fully configured search service featuring intuitive user experiences. It offers capabilities such as scoring, faceting, suggestions, synonyms, and geo-search. The fully managed service in Azure allows organizations to avoid the operational overhead needed to debug index corruption, monitor service availability, or manually scale during traffic fluctuations. Knowledge mining solutions built with Azure Cognitive Search are enterprise-grade managed services that:

- Streamline development and reduce maintenance overhead
- Benefit from the massive compute power of the Azure cloud, making it possible to store and process vast amounts of data with a 99.9% uptime SLA
- Provide industry-leading security and privacy, and a broad set of international and industry-specific compliance standards
- Protect content from malicious acts with encryption built-in throughout the entire indexing pipeline
- Let companies control per-user access to content through security filters
- Offer multi-layer security across physical data centers, infrastructure, and operations

Cognitive Skills in your toolbox

When using Azure Cognitive Search to build a knowledge mining solution, you can choose from a wide range of pre-trained Microsoft Cognitive Services below.

Language APIs:

- Text Analytics API helps developers detect sentiment, key phrases, named entities, and language from text:
    - Key phrase extraction evaluates unstructured text and returns a list of strings denoting key talking points in the input text.
    - Sentiment analysis returns a numeric score between 0 and 1, where scores closer to 1 indicate positive sentiment and scores closer to 0 indicate negative sentiment.
    - Named entity recognition detects named entities in text, including people, locations, organizations, and more.
    - Language detection returns the detected language and a numeric score between 0 and 1. Scores close to 1 indicate 100% certainty in the identified language.
- Translator Text API uses neural translation models to translate text into different languages in real-time.

Vision APIs:

- Face API enables developers to detect and compare faces, organize images into groups based on similarities, and identify previously tagged people in images. Features include:
    - Face verification checks the likelihood that two faces belong to the same person.
    - Face detection finds faces in an image and predicts facial attributes like age, gender, pose, and more.
    - Emotion recognition returns a set of emotions for each face in an image.
- Computer vision API extracts rich information from images to categorize and process visual data. Features include:
    - Image analysis and tagging identifies visual content like objects, image type, or color scheme
    - Handwriting and printed text recognition (OCR) detects embedded printed and handwritten text, extracts recognized words into machine-readable character streams, and enables searching.
    - Brand, celebrity, and landmark recognition recognizes more than 1,500 global brands and logos, 1 million celebrities from business, politics, sports, and entertainment, as well as 9,000 natural and woman-made landmarks from around the world.
- Ink recognizer API recognizes digital handwriting, common shapes, and the layout of inked documents for various scenarios like notetaking, form-filling, content search, and document annotation.
- Video indexer API extracts metadata such as spoken words, written text, faces, speakers, celebrities, emotions, topics, brands, and scenes from video and audio files.
- Form recognizer API applies advanced machine learning to accurately extract text, key/value pairs, and tables from forms while understanding the relationships between fields and entries. Users can leverage prebuilt models or train a model tailored to their company’s documents using only five samples.

Key knowledge mining scenarios with Azure Cognitive Search

By exploring everyday use cases, business leaders and developers can find inspiration towards taking the first steps into leveraging the benefits of knowledge mining in Azure Cognitive Search.

Digital content management

Given the amount of unstructured data created daily, many companies are struggling to make use of or find information within their files. Leveraging knowledge mining through a search index makes it easy for end customers and employees alike to locate what they are looking for faster. Using knowledge mining with Azure Cognitive Search, organizations in any industry can create advanced search experiences for content such as articles, images, or products.

For example, a publication could build cognitive search into its website to help readers find what they're looking for faster:

- **Ingest**: article and image archives, photos and videos, internal documents, marketing assets, brochures
- **Enrich**: automatic image captioning and object detection with computer vision, celebrity recognition, language translation, and entity recognition
- **Explore and analyze**: integrate search index into a website

Or, a professional sports league could leverage knowledge mining to collect and organize media about specific players, teams, or statistics and make it searchable for fans:

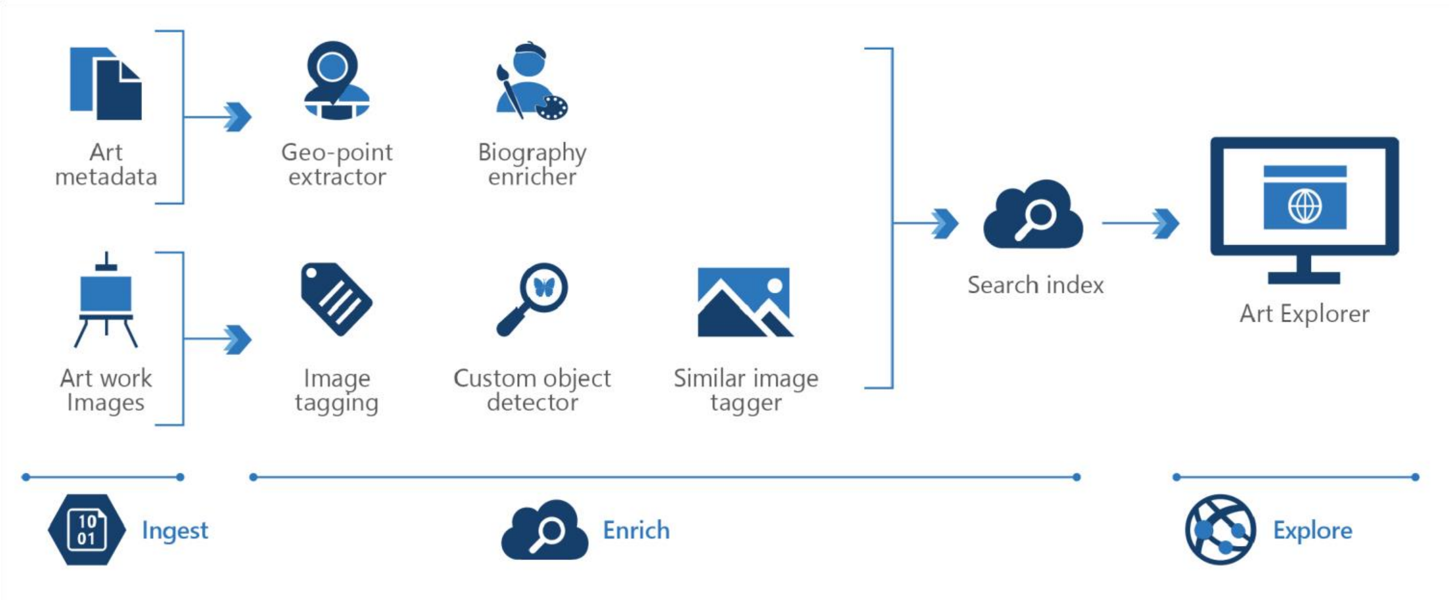
- **Ingest**: game photos, game recaps, box scores, player interview videos, historical game data, league transactions, and contact information
- **Enrich**: entity recognition, face detection, and customized statistical analysis
- **Explore and analyze**: integrate search index into a website

The Metropolitan Museum of Art (the Met) illustrates how knowledge mining enhances media exploration. The Met is one of the world's largest art museums, with nearly 2 million works of art representing 5,000 years of human history from across the globe.

The Met houses its comprehensive collection of artwork in New York City but makes it accessible globally through a website where the public can explore images and information about the art. People can search by visual elements such as the art medium or technique, specific objects, people, or colors in the art. They can also search by historical details such as the artist's biographical information, where and when the piece was created, the artist's influences, and so on.

Until recently, the Met's staff had to tag every piece of art with keywords manually. Using computer vision models, they now automatically recognize objects depicted in the piece or identify visually similar artworks. By pulling in metadata about the art from the web and the Met's internal data sources, the knowledge mining solution also automatically extracts information about the artist and the geographic location related to a piece through custom AI models called geo-point extraction and biography enricher. Knowledge mining is helping to uncover new details and relationships among the works of art.

Digital content management architecture



Key technologies used to develop digital content management tools

- Azure Cognitive Search
- Microsoft Computer Vision API
- Microsoft Face API
- Web API custom skill interface

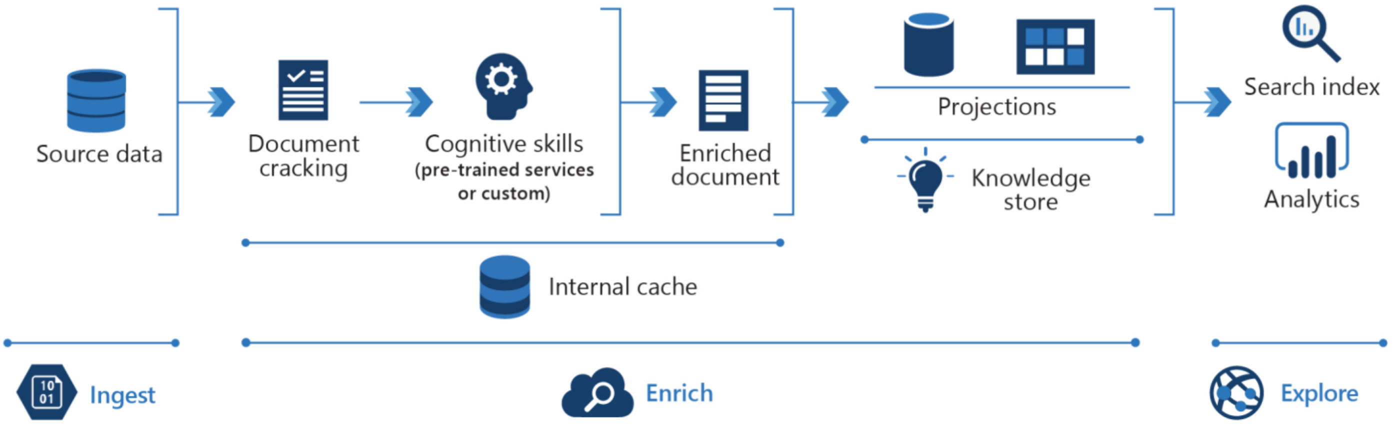
Customer support and feedback analytics

For many companies, customer support is costly and inefficient. Knowledge mining can help customer support teams quickly find the right answer for a customer inquiry or assess customer sentiment at scale.

Every company is looking to enhance the customer experience. Knowledge mining solutions can aggregate and analyze data to discover trends about what customers are saying and use that information to improve products and services:

- **Ingest**: customer support tickets, chat logs, call transcriptions, customer emails, customer payment history, product reviews, social media feeds, online comments, feedback forms, and surveys
- **Enrich**: keyphrase extraction, sentiment analysis, language translation, bot services, custom models to focus on specific products or company policies
- **Explore and analyze**: compile enriched documents in the knowledge store and project them into tabular or object stores, then surface trends in an analytics dashboard, such as frequent issues, popular products, and much more

Feedback analytics sample architecture

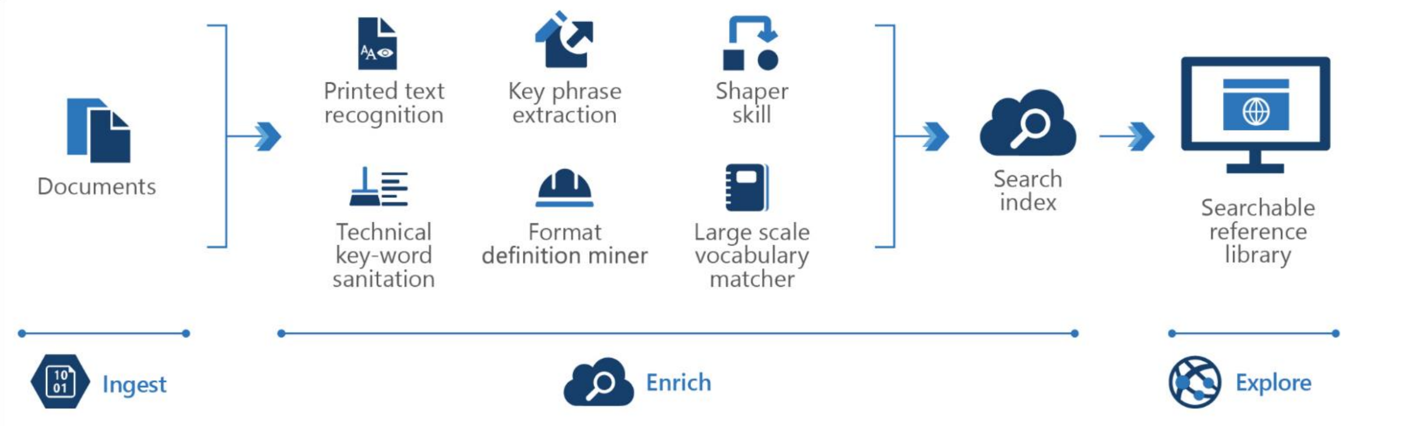


Cognitive search solutions can also help customer service teams find answers to customer questions faster by searching large volumes of information more quickly:

- **Ingest**: customer chat logs, customer support call recordings, company support documentation, product and warranty information, legal documents, customer profiles, customer service manuals
- **Enrich**: keyphrase extraction, sentiment analysis, entity recognition, language detection, language translation, bot services, custom models to focus on specific products or company policies
- **Explore**: integrate search index into customer service support application

Offshore dredging supplier and wet market equipment manufacturer Royal IHC uses knowledge mining to power a searchable reference library for its customer support teams. Before they developed a knowledge mining solution, support personnel spent about 25% of their time searching for the right documentation to address customer inquiries. Royal IHC ingests technical documents and uses AI services such as text recognition and keyphrase extraction to enrich them. It also built customized AI models such as technical keyword sanitation, format definition miner, shaper skill, and large-scale vocabulary matcher to meet the needs of its particular use case. The reference library web application empowers service engineers to find information quickly and serve customers more efficiently.

Document search architecture



Key technologies used to enhance organizational customer support and feedback analytics

- Azure Cognitive Search
- Microsoft Text Analytics API
- Microsoft Translator Text API

- Web API custom skill interface

Data extraction and process management

At an operational level, manual data entry is often time-consuming and error-prone. Knowledge mining streamlines business processes by extracting information from business documents and funneling it automatically into business applications.

In practice, an agency with a global network could use knowledge mining to extract relevant data from invoices and populate that information in business documentation for timely processing:

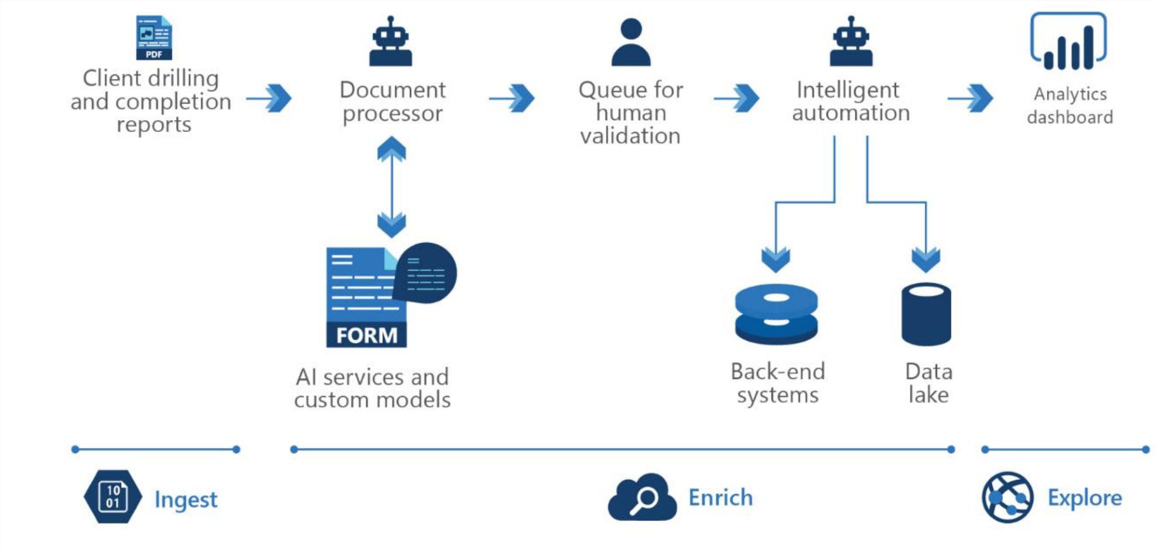
- **Ingest**: SOWs, requests for proposal, invoice archives, sales team correspondence, W2 forms, receipts, healthcare claim forms
- **Enrich**: optical character recognition, forms recognition, layout understanding, table extraction, key-value pair extraction
- **Explore**: automatically populate data from invoices into ELP systems or databases

Likewise, A financial services company that manages billions of dollars in assets may require thousands of documents to be analyzed every day by a fleet of employees. Developers might use knowledge mining to extract and normalize data from these documents and provide custom analytics that helps clients make better investment decisions:

- **Ingest**: bank statements, legal agreements, balance sheets, income statements, cash flow statements, company disclosures, SEC documents, annual reports, etc.
- **Enrich**: optical character recognition, layout understanding, table extraction, key-value pair extraction
- **Explore**: compile enriched documents in the knowledge store and project them into tabular or object stores, then surface trends in an analytics dashboard, such as frequent issues, popular products, and much more

In the oil and gas industry, Chevron illustrates data extraction in action. Chevron receives thousands of oil rig drilling PDF reports every day in its Canada headquarters alone. These reports come in a wide range of formats, making the manual process of extracting the useful data time-consuming and error-prone. Using a robotic process automation (RPA) platform with built-in knowledge mining capabilities, Chevron automatically extracts text, fields, and tables from their highly specialized forms to automate data entry. If data in the forms cannot be processed with sufficient confidence, the software notifies a human that the content needs to be validated. With knowledge mining as part of the RPA solution, subject matter experts have time to focus on more valuable tasks, and Chevron executives can analyze the business with higher speed, accuracy, and depth.

Robotic process automation architecture



Key technologies used to develop data extraction tools for process management

- Azure Cognitive Search
- Form Recognizer
- Web API custom skill interface

Technical content review and research

When organizations task employees with review and research of technical data, it can be tedious to read page after page of dense text. Knowledge mining helps employees quickly review these dense materials. In industries where bidding competition is fierce, or when the diagnosis of a problem must be quick or in near real-time, companies can use knowledge mining to avoid costly mistakes.

Case in point, healthcare professionals have a lot of patient data to keep up with, and they want to stay on top of the latest research. Doctors can use knowledge mining to sift through massive amounts of clinical data and medical publications to make informed decisions about a patient's health:

- **Ingest**: medical journals, anonymized patient data, x-rays, patent records, pharmaceutical filings, etc.
- **Enrich**: keyphrase extraction, metadata extraction, optical character recognition, language translation, customized models for HIPAA compliance, etc.
- **Explore**: integrate search index into a searchable directory

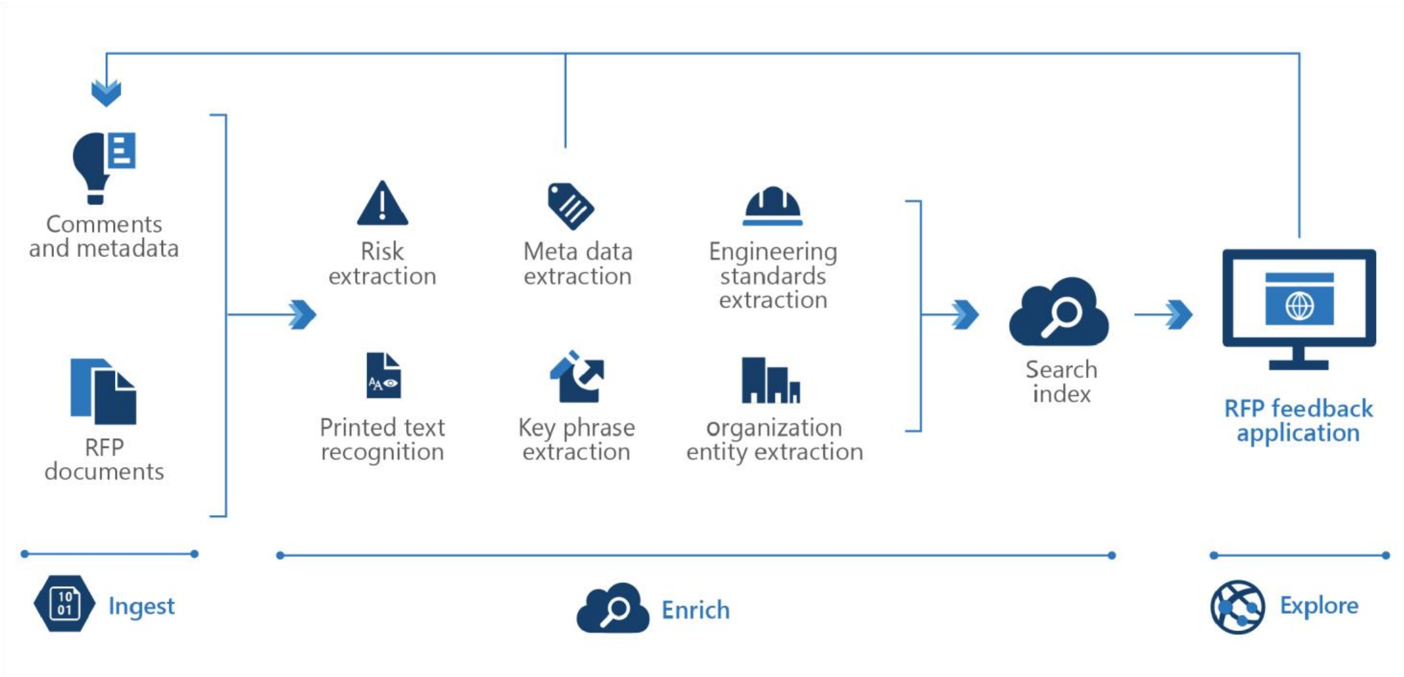
Technical content review is also extremely valuable in the manufacturing and industrial space:

- **Ingest**: technical documents, engineering standards, product manuals, user guides
- **Enrich**: optical character recognition, key phrase extraction, entity recognition, language translation, customized models to extract industry-specific terms such as product names or engineering standards, customized models to flag potential risks or other essential information
- **Explore**: integrate search index into an existing business application

For example, Howden creates products for multiple sectors ranging from mine ventilation and wastewater treatment to heating and cooling. With each new business opportunity, Howden engineers must scour thousands of pages of sources to create an accurate bid. Minor details in the bidding process can make the difference between a healthy profit or lost opportunity on a project.

Howden uses standard AI services to extract risks and names of organizations. They also use custom AI models to extract engineering standards, specialized components, and more from source documents. This information is fed into a collaborative portal where engineers can search the company's extensive records more quickly and share information throughout project lifecycles. As users share new information, the portal continuously ingests it to expand its knowledge base. The portal increases Howden's response time and reduces costly mistakes on bid proposals.

RfP feedback system architecture



Key technologies used to implement tools for technical content review and research

- Azure Cognitive Search
- Text Analytics API
- Translator Text API
- Form Recognizer
- Web API custom skill interface

Auditing and compliance management

In the ever-changing world of regulations, organizations face the challenge of staying on top of audits and compliance. Mistakes in contracts and record-keeping can have serious financial ramifications. At the enterprise level, teams of lawyers might not be enough to catch everything. Knowledge mining can provide helpful assistance for organizations looking to stay above board.

For most organizations, the legal department faces the challenge of reviewing thousands of pages of documentation. Developers could use knowledge mining to help attorneys quickly identify entities of importance from discovery documents and flag important ideas across documents:

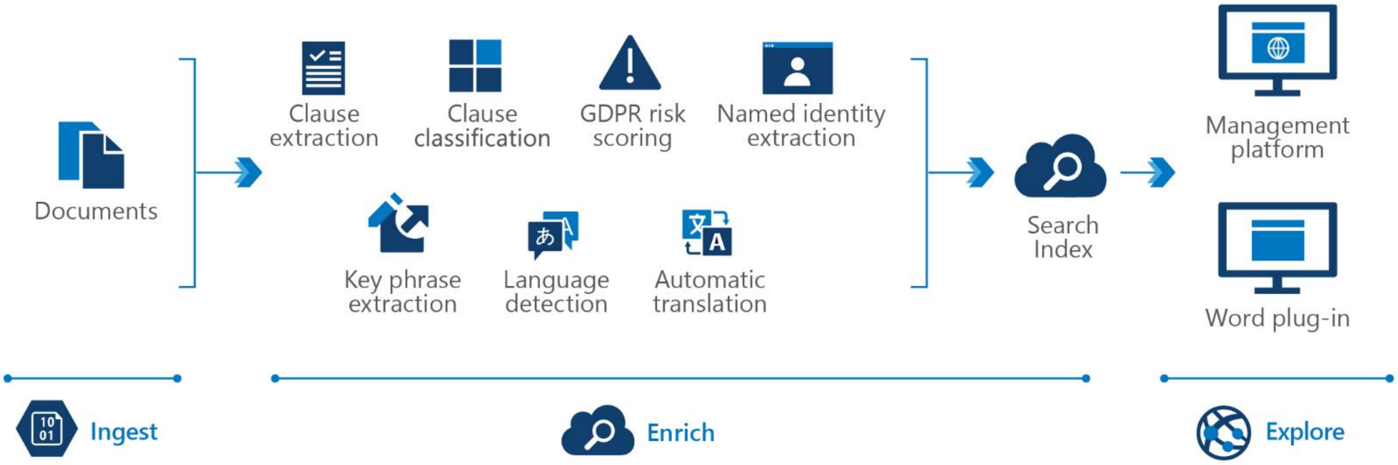
- **Ingest**: affidavits, meeting minutes, operating agreements, non-disclosure agreements, privacy policies, terms of use, memorandums of understanding, licensing agreements, letters of intent, power of attorney, deeds, discovery documentation, etc.
- **Enrich**: keyphrase extraction, language detection, language translation, and custom models to identify specific legal terms and clauses
- **Explore**: index data in a searchable internal application

Knowledge mining can help in the financial services realm. For example, organizations need to conduct audits of SEC filings to make sure they comply with SEC regulations. Developers can use knowledge mining to create custom models that identify risks in SEC filings based on SEC regulations:

- **Ingest**: company bylaws, operating agreements, bank statements, legal agreements, balance sheets, income statements, cash flow statements, company disclosures, SEC documents, annual reports, transcripts from shareholder meetings
- **Enrich**: keyphrase extraction, language detection, language translation, entity extraction (organizations and people) and custom models to identify certain regulatory obligations
- **Explore**: leverage data for a searchable web application for financial risks

As an example, Icertis offers a contract management platform with built-in knowledge mining capabilities. The platform is used by enterprise customers in over 90 countries to manage more than 5 million contracts. Companies can use the platform to easily search complex documents for specific people, places, and organizations, legal terms, and clauses, or regulatory obligations. A salesperson drafting a new contract could search for existing contracts from customers of similar size, industry, or payment terms, for example. Companies can also use the platform to translate documents into other languages and uncover complex relationships between thousands of contracts across the entire enterprise. These insights help improve contract compliance, reduce risk, accelerate contract negotiations, and optimize agreement terms. The platform has helped Icertis customers import legacy contracts up to 80% faster, improve post-execution compliance by up to 90%, and reduce the cost of compliance by up to 60%.

Contract management platform architecture



Key technologies used to build auditing and compliance management tools

- Azure Cognitive Search
- Text Analytics API
- Translator Text API
- Web API custom skill interface

Getting started

Microsoft has several resources to help you get started with knowledge mining. Below are two solution accelerators, which provide you with starter solutions and the artifacts needed to get up and running with a knowledge mining solution using Azure Cognitive Search. Also, the Knowledge Mining Bootcamp provides a hands-on-lab to guide you through setting up a knowledge mining solution in Azure.

Knowledge Mining Solution Accelerator

The Knowledge Mining Solution Accelerator sample application provides developers with the resources needed to quickly build an initial Knowledge Mining prototype with Azure Cognitive Search. Use this accelerator to jump-start your development efforts with your data or as a learning tool to understand better how you can use Azure Cognitive Search to meet the unique knowledge mining needs of your business.

In the Knowledge Mining Solution Accelerator, you can find the artifacts needed to create a Cognitive Search Solution. The repo includes templates for deploying the appropriate Azure resources, assets for creating your first search index, templates for using custom skills, a basic web app, and PowerBI reports for monitoring search solution performance. Best practices are infused throughout the documentation to help guide you. With Azure Cognitive Search, you can easily index both digital data (such as documents and text files) and analog data (such as images and scanned documents).

The solution accelerator also includes the Knowledge Mining Workshop. The workshop provides a step-by-step lab that allows you to explore how knowledge mining with Azure Cognitive Search can be used to extract information from a demo data set. The workshop contains modules that help you better understand how to ingest content, build custom skills, and then search for information through a web front-end or project the enriched data into compelling visuals you create in Power BI. You will also examine more advanced topics like phonetic search and boosting the relevancy of search results.

Knowledge Mining Bootcamp

For a guided introduction to setting up a knowledge mining solution in Azure, Microsoft provides a free Knowledge Mining Bootcamp. This hands-on lab guides users through the creation of an enterprise search solution by applying knowledge mining to business documents like contracts, memos, presentations, and images. Use Microsoft Azure AI technology to extract insights from unstructured data and expose the results in a Bot interface.

Custom skills code samples

The Azure Search Power Skills GitHub repo contains a collection of useful functions to be deployed as custom skills for Azure Cognitive Search. The skills can be used as templates or starting points for developing custom skills, or they can be used as-is if they happen to meet your requirements.

You can also find an example of creating a custom skill using the Bing Entity Search API in the Azure Cognitive Seach documentation. This example shows you how to create a web API custom skill. This skill accepts locations, public figures, and organizations, and return descriptions for them. The example uses an Azure Function to wrap the Bing Entity Search API so that it implements the custom skill interface.

Additional examples you can use to get started include creating a custom skill using Python and the Cognitive Search Skills Extractor GitHub repository.

Learn more

The staggering amount of data companies generate should not be a barrier to success. There is untapped potential in many organizations—virtual gold mines just waiting to be unearthed with knowledge mining. Whether in media exploration, customer support, process management, technical content review, or auditing and compliance management, knowledge mining gives organizations the tools to gain an edge in the marketplace and improve decision-making.

Additional resources

Take a moment to review the additional resources below to learn more about bringing knowledge mining into your organizational best practices.

- Azure Cognitive Search: Product page for Azure Cognitive Search with information on features and links to documentation and training. Try out knowledge mining with an Azure free trial.
- AI enrichment in Azure Cognitive Search documentation: Documentation resources for AI enrichment in Azure Cognitive Search, including content on getting started, the latest tutorials, and how-to guides for everything related to building an advanced knowledge mining solution.
- Design tips for AI enrichment in Azure Cognitive Search: Review a list of tips and tricks to keep you moving as you get started with AI enrichment capabilities in Azure Cognitive Search.
- Predefined cognitive skills: Learn more about the predefined cognitive skills provided with Azure Cognitive Search that you can include in a skillset to extract content and structure.
- Form Recognizer: Accelerate your business processes by automating information extraction. Form Recognizer applies advanced machine learning to accurately extract text, key/value pairs, and tables from documents.
- Azure Cognitive Services: Learn how to build intelligent and supported algorithms into apps, websites, and bots to see, hear, speak, understand, and interpret your user needs.
- Azure AI: Product page for Azure AI services with information on solutions, services, and documentation.
- Full-text search in Azure Cognitive Search: Learn more about how Lucene full-text search works in Azure Cognitive Search.

This document is provided "as is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

[1] Griffith, Eric. "90 Percent of Big Data We Generate Is an Unstructured Mess" PCMag https://www.pcmag.com/news/364954/90-percent-of-the-big-data-we-generate-isan-unstructured-me. Retrieved on 8/7/2019

[2] Welson-Rossman, Tracey. "'I See Data' – Forge.AI Mines The World's Unstructured Data" Forbes https://www.forbes.com/sites/traceywelsonrossman/2019/01/28/i-seedata-forge-ai-mines-the-worlds-unstructured-data/#591fc3991067. Retrieved on 8/7/2019

[3] Solliance and Microsoft. "Artificial Intelligence: Microsoft Practice Development Playbook" Microsoft https://assetsprod.microsoft.com/mpn/en-us/get-ai-playbook.pdf. Retrieved on 3/5/2020