Agenda

Demo Sample Application for Cognitive Search

Review Cognitive Search Features powering the sample application

Index Creation

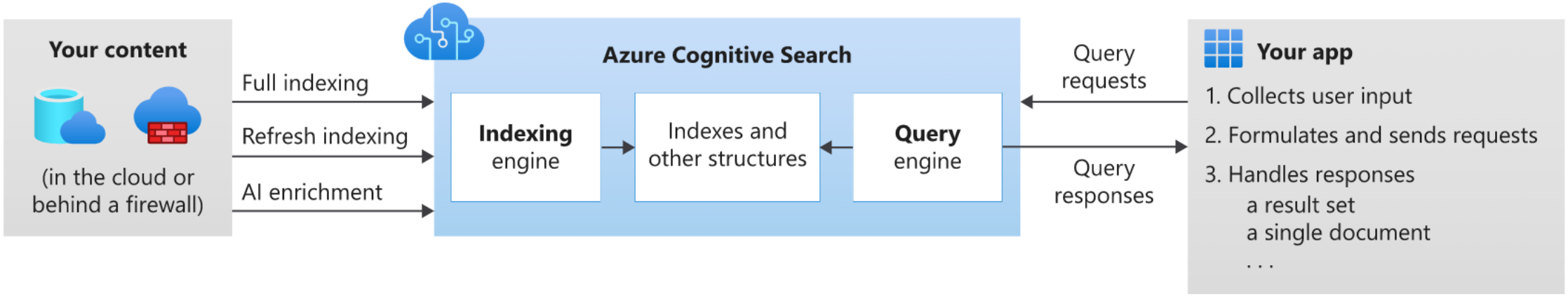
AI Enrichment

- Built-in Skills
- Custom Skills

Indexer

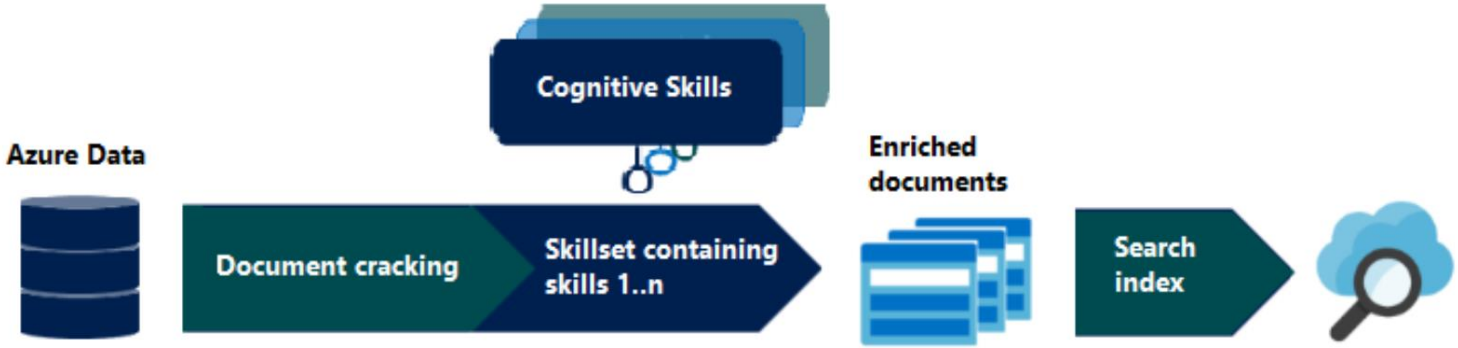
Search Experience

Azure Cognitve Search



- **Indexing** brings content into to your search service and makes it searchable
- **AI Enrichment** adds content transformation during indexing. Enrichments create new information from the original content. AI processing to make unsearchable content types text-searchable
- The **search experience** is defined using the Cognitive Search APIs, and can include autocomplete, synonym matching, fuzzy matching, filter, and sort

AI Enrichment Pipeline



Document cracking – extract text and images. Image content can be routed to skills that perform image processing, while text content is queued for text processing

Built-in Text Analytics Cognitive Skills:

- Entity Recognition skill to extract “people”, “organizations”, “locations”, “product”, etc.. (see Language studio: Named entities tryout)
- Language detection skill to detect the language of the document
- Text translation cognitive skill to translate the document content

Built-in Computer Vision Cognitive Skills:

- OCR recognizes typeface and handwritten text in scanned documents
- Image cognitive skills to generate tags and captions from images in the documents

Custom Skill: Apply transformations unique to your content by calling an external function. Custom Skills are wrapped in an interface definition that allows for integration into the pipeline.

Demo – Create Index

AI Enrichment: Indexer - Stages of Indexing

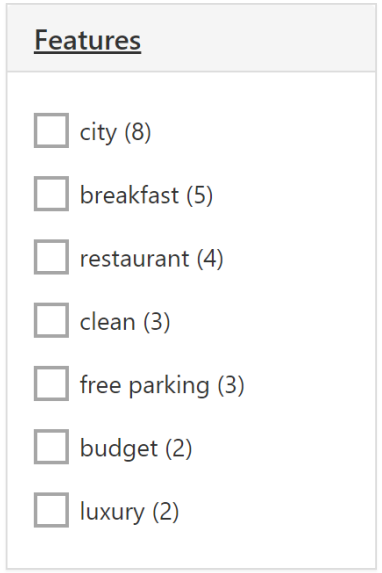
An ***indexer*** in Azure Cognitive Search is a crawler that extracts searchable text and metadata from an external Azure data source and populates a search index using field-to-field mappings between source data and your index.

Field Mappings - An indexer extracts text from a source field and sends it to a destination field in an index. When field names and types coincide, the path is clear. However, you might want different names or types in the output, in which case you need to tell the indexer how to map the field.

Skillset execution is an optional step that invokes built-in or custom AI processing, it is where enrichment occurs

Output Field Mappings - The output of a skillset is really a tree of information called the *enriched document*. Output field mappings allow you to select which parts of this tree to map into fields in your index.

AI Enrichment: Define Built-in Skill



1. Define CustomEntityLookup skill in the skillset definition

```
{
      "@odata.type": "#Microsoft.Skills.Text.CustomEntityLookupSkill",
      "name": "#7",
      "description": "extract custom entities",
      "context": "/document/merged_content",
      "defaultLanguageCode": "en",
      "entitiesDefinitionUri": "",
      "globalDefaultCaseSensitive": true,
      "globalDefaultAccentSensitive": true,
      "globalDefaultFuzzyEditDistance": 0,
      "inputs": [
        {
          "name": "text",
          "source": "/document/merged_content",
          "sourceContext": null,
          "inputs": []
        }
      ],
      "outputs": [
        {
          "name": "entities",
          "targetName": "features"
        }
      ],
      "inlineEntitiesDefinition": [
      {
        "name" : "breakfast"
      },
      {
        "name" : "city"
      },
```

2. Add new field to the index definition to store the output of the built-in skill execution

```
{
  "name": "features",
  "type": "Collection(Edm.String)",
  "searchable": true,
  "filterable": true,
  "retrievable": true,
  "sortable": false,
  "facetable": true,
  "key": false,
  "indexAnalyzer": null,
  "searchAnalyzer": null,
  "analyzer": "standard.lucene",
  "normalizer": null,
  "synonymMaps": []
}
```

Azure Cognitive Search Service REST | Microsoft Docs

AI Enrichment: Define Built-in Skill

Sample output of the CustomEntityLookupSkill


```
{
  "values" :
  [
   {
    "recordId": "1",
    "data" : {
     “features": [
      {
       "name" : "breakfast",
       "description" : "",
       "id" : "differentIdentifyingScheme987",
       "matches" : [
        {
         "text" : "breakfast",
         "offset" : 13,
         "length" : 9,
         "matchDistance" : 0
        },
       ]
      },
      {
       "name" : "budget",
       "description" : "",
       "matches" : [
        {
         "text" : "budget",
         "offset" : 37,
         "length" : 6,
         "matchDistance" : 0
        }
       ]
      }
     ]}}]}
```


3. Add new output field mapping to indexer definition to the list of features

```
"outputFieldMappings": [
    {
      "sourceFieldName": "/document/merged_content/features/*/name",
      "targetFieldName": "features",
      "mappingFunction": null
    },
```

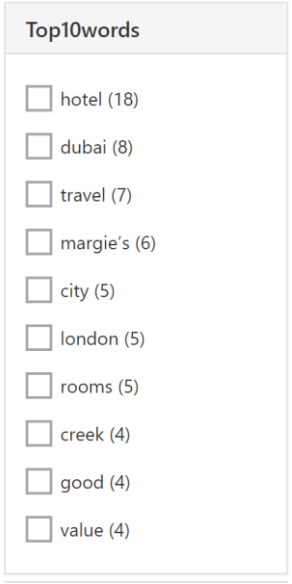
4. Search - facet by “features”

https://search-service-westus.search.windows.net/indexes/cogsearch-demo2/docs?api-version=2020-06-30&search=*&$count=true&facet=features


```
{
    "@odata.context": "https://search-serv
        ",
    "@odata.count": 20,
    "@search.facets": {
        "features": [
            {
                "count": 8,
                "value": "city"
            },
            {
                "count": 5,
                "value": "breakfast"
            },
            {
                "count": 4,
                "value": "restaurant"
            },
```

AI Enrichment: Custom Skill – Top10Words

Apply custom transformations unique to your content by calling an external function



1. Content sent to Top10WordsHttpFunc Custom Skill


```
{
    "values": [
        {
            "recordId": "e1",
            "data":
            {
                "text": "this hotel is very good for the money. The rooms are exceptionally large with good
                    toiletries in the bathroom. Hotel room was spotless, staff were exceptionally helpful and polite.
                    The French restaurant looked good although unfortunately I did not have time to try it out, but
                    highly spoken about. I would not hesitate to stay here again. Five minutes in a taxi to City
                    Centre Shopping Mall or the old area of Deira with gold/spice souks. The people who own this
                    chain of hotels really seem to care about their guests."
            }
        },
        {
            "recordId": "e2",
            "data":
            {
                "text": "Stayed here in two rooms with family of 5 for two nights. From the outside the hotel looks
                    run down but rooms were well decorated and clean. Free parking was a bonus. Reception staff were
                    really helpful, telling us about the buses and where to go. Just off the main road but with ac on
```


2. Result returned by Top10WordsHttpFunc


```
{
    "values": [
        {
            "recordId": "e1",
            "data": {
                "text": [
                    "good",
                    "hotel",
                    "exceptionally",
                    "money",
                    "rooms",
                    "large",
                    "toiletries",
                    "bathroom",
                    "room",
                    "spotless"
                ]
```

Integrate the Custom Skill

1. **Define the Custom WebApi skill in the skillset definition**

```
{
  "@odata.type": "#Microsoft.Skills.Custom.WebApiSkill",
  "name": "top10words",
  "description": "Extract Top 10 Words Skill",
  "context": "/document",
  "uri": "https://km-ch-func-app.azurewebsites.net/api/Top10WordsHttpFunc?code=NVp9mH721UVMCFGCbkK1JhnyRbaMjVmN[redacted]==",
  "httpMethod": "POST",
  "timeout": "PT1M30S",
  "batchSize": 1,
  "degreeOfParallelism": null,
  "inputs": [
    {
      "name": "text",
      "source": "/document/merged_content",
      "sourceContext": null,
      "inputs": []
    }
  ],
  "outputs": [
    {
      "name": "top10words",
      "targetName": "top10words"
    }
  ],
  "httpHeaders": {}
}
```

Web API Input Format

```
{
    "values": [
        {
            "recordId": "e1",
            "data":
            {
                "text": "Four score and
                    conceived in Liberty
            }
        },
        {
            "recordId": "e2",
            "data":
            {
                "text": "Now we are eng
                    conceived and so ded
            }
```

Web API Output Format

```
{
    "values": [
        {
            "recordId": "e1",
            "data": {
                "top10words": [
                    "four",
                    "score",
                    "seven",
                    "years",
                    "ago",
                    "fathers",
                    "brought"
```

2. Add new field to the index definition to store the output of the custom skill execution

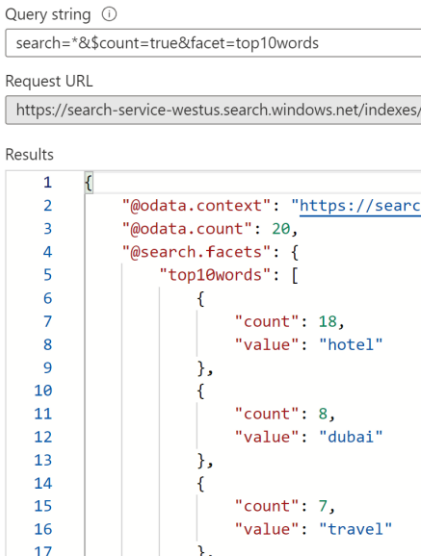
```
{
  "name": "top10words",
  "type": "Collection(Edm.String)",
  "searchable": true,
  "filterable": true,
  "retrievable": true,
  "sortable": false,
  "facetable": true,
  "key": false,
  "indexAnalyzer": null,
  "searchAnalyzer": null,
  "analyzer": "en.microsoft",
  "normalizer": null,
  "synonymMaps": []
}
```

3. Add new output field mapping to indexer definition to the list of features

```
"outputFieldMappings": [
    {
      "sourceFieldName": "/document/top10words",
      "targetFieldName": "top10words",
      "mappingFunction": null
    },
```

4. Search - facet by “top10words”

https://search-service-westus.search.windows.net/indexes/cogsearch-demo2/docs?api-version=2020-06-30&search=*&$count=true&facet=top10words



Search Experience: Synonyms

- Define synonym map and apply it to the searchable fields in the index

- A query on UK will expand to ”United Kingdom”, “Britain“ and “Great Britain"

```
{
  "name": "geo-synonyms",
  "format": "solr",
  "synonyms":
    "USA, United States, America, United States of America\n
    UK, United Kingdom, Britain, Great Britain\n
    UAE, United Arab Emirates, Emirates\n"
}
```

```
{
            "name": "content",
            "type": "Edm.String",
            "searchable": true,
            "filterable": false,
            "retrievable": true,
            "sortable": false,
            "facetable": false,
            "key": false,
            "indexAnalyzer": null,
            "searchAnalyzer": null,
            "analyzer": "standard.lucene",
            "normalizer": null,
            "synonymMaps": [
                "geo-synonyms"
            ]
        },
```

Search Experience: Synonyms

Search string: search=uk&highlight=content&searchMode=all&$count=true&facet=locations&select=content


```
{
    "@odata.context": "https://search-service-westus.search.windows.net/indexes('cogsearch-demo2')/$metadata#docs(*)",
    "@odata.count": 5,
    "@search.facets": {
        "locations": [ ···
        ]
    },
    "value": [
        { ···
        },
        { ···
        },
        { ···
        },
        { ···
        },
        {
            "@search.score": 0.800081,
            "@search.highlights": {
                "content": [
                    "Margie's Travel Presents… \n\nLondon \nLondon is the capital and \n\nmost populous city of
\n\nEngland and the <em>United</em> \n\n<em>Kingdom</em>.",
                    "Standing on the \n\nRiver Thames in the south \n\neast of the island of <em>Great</em>
\n\n<em>Britain</em>, London has been \n\na major settlement for two \n\nmillennia."
                ]
            },
```

Search Experience: Autocomplete

The Autocomplete API finishes a partially typed query input using existing terms in the search index for use in a secondary query

- Define a suggester in the index
- Reference the suggester during search

```
"suggesters": [
    {
        "name": "sg",
        "searchMode": "analyzingInfixMatching",
        "sourceFields": ["locations"]
    }
```

Search Experience: Autocomplete

Post https://search-service-westus.search.windows.net/indexes/cogsearch-demo2/docs/autocomplete?api-version=2020-06-30

```
{
    "fuzzy": true,
    "search": "lo",
    "suggesterName": "sg",
    "autocompleteMode": "twoTerms"
}
```

Results

```
{
    "@odata.context": "https://search-service-westus.sea
        $metadata#Collection(Microsoft.Azure.Search.V20
    "value": [
        {
            "text": "lombard hotel",
            "queryPlusText": "lombard hotel"
        },
        {
            "text": "los angeles",
            "queryPlusText": "los angeles"
        },
        {
            "text": "lost city",
            "queryPlusText": "lost city"
        },
        {
```

Search Experience: Lucene based Query Language

Example search request
POST indexes/cogsearch-demo/docs/search?api-version=2020-06-30

```
{
  "queryType": "simple",
  "search": "UK +breakfast",
  "highlight": "content",
  "searchMode": "any",
  "count" : true,
  "select" : "content, locations, features, top10words"
}
```

queryType – "**simple**"(default) or "**full**". The Simple query language is intuitive and robust, often suitable to interpret user input as-is without client-side processing. It supports query operators familiar from web search engines, e.g., +, |, -. The Full Lucene query language extends the default Simple query language by adding support for more operators and query types like wildcard, fuzzy, regex, and field-scoped queries.

searchMode - Valid values are "any"(default) or "all". Specifies whether any or all of the search terms must be matched in order to count the document as a match.

highlight - Optional. A set of comma-separated field names used for hit highlights. Only searchable fields can be used for hit highlighting.

count – Optional. This is the count of all documents that match the search and $filter parameters

select - Optional. A list of comma-separated fields to include in the result set. Only fields marked as retrievable can be included in this clause.

Ref - Search Documents (Azure Cognitive Search REST API) | Microsoft Docs

References

- Azure Cognitive Search documentation | Microsoft Docs
- Document operations using Azure Cognitive Search REST APIs | Microsoft Docs
- Azure-Samples/azure-search-knowledge-mining: Azure Search Knowledge Mining Accelerator (github.com)
- Introduction to Azure Cognitive Search - Learn | Microsoft Docs
- Accelerate search index development with Visual Studio Code - Microsoft Tech Community

- **Thank You**