Chapter 11: Finding Answers: Watson Discovery

Return to Table of Contents

This chapter is an update to the original chapter, where we used Retrieve and Rank to create a collection, process queries against that collection as both a straight retrieve and then later including the ranking capability as part of that service.

Retrieve and Rank has been replaced with Watson Discovery. This means that we have better exploration tools available to us and, as well, can do things like create the necessary configuration file interactively in the browser rather than only on our laptops. We can still do what we did in the past, we just have more options, now.

Concept Changes

When using Revrieve and Rank, we are much more aware of the underlying technology and work with converted documents and do significant manual work to train the Ranking system. A lot of code is written to handle document upload, document conversion, and training. All of that is now handled by the Web interface for Watson Discovery. We can still write code to perform all those steps, and this code is included in the tutorial. However, it is no longer necessary to do that work.

Training can take longer, because we need 50 queries in the system to train Discovery and the documents returned for each query need to have some of them rated as relevant and some as not relevant. This is, in practical terms, the work of a subject matter expert. For us to do that, we need to actually read each of the 12 documents we're going to use in this small example, so that we can apply the 'relevant/not-relevant' responses.

Process Changes

All of the work can be done through the Watson Discovery Web Interface. We will still write a query interface, which draws very heavily from the Watson Discovery News work done in Chapter 8.

Code Changes

Only the query interface needs to be written for both the server side and the browser side

Server Side

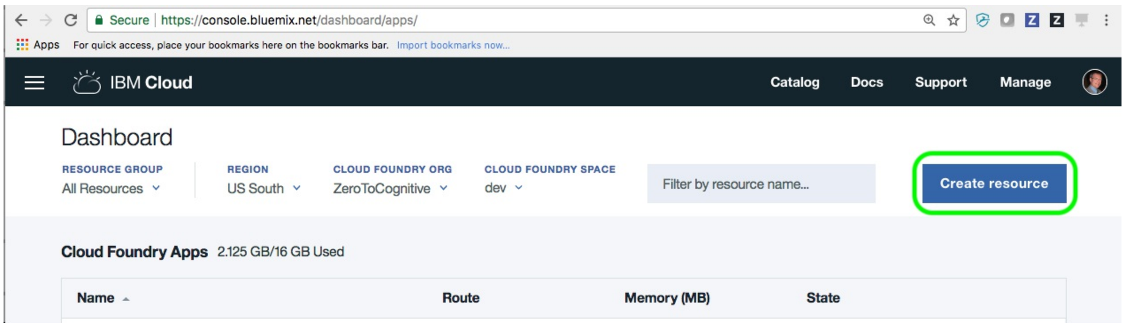
We need a "find" service

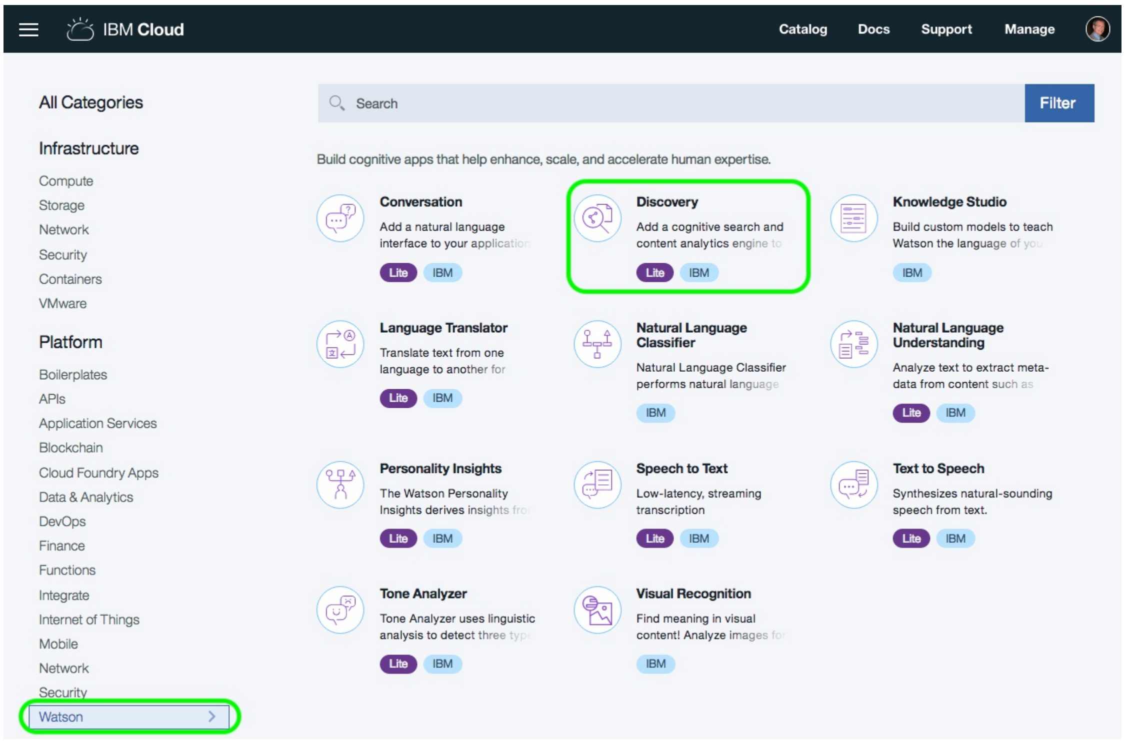
Client (Browser)

We need a "find" interface

Executing the Lab

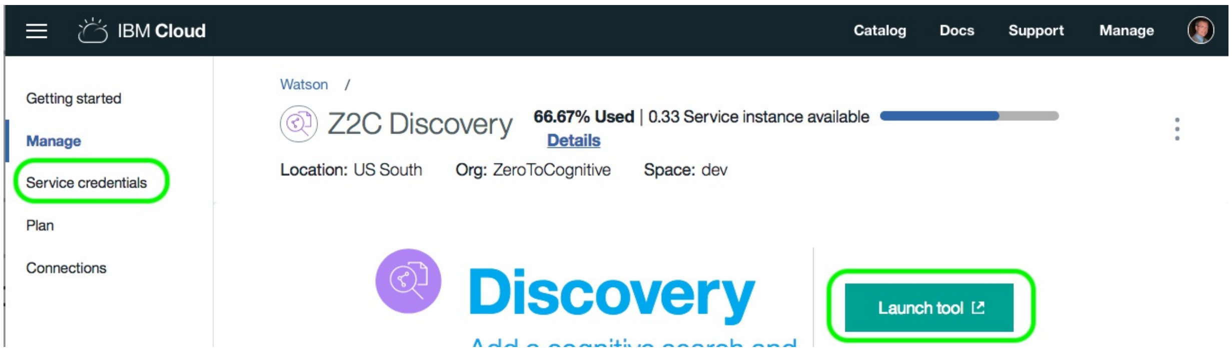
- We are going to use the IBM Institute for Business Value documents located at this URL: https://www-935.ibm.com/services/us/gbs/thoughtleadership/
- There are 12 documents at this location. We want all 12 of them downloaded to our system and stored in the Chapter11/HTML/Documents/Source folder.
- Log into IBM Cloud (formerly IBM Bluemix) at
  - https://console.ng.bluemix.net (Americas)
  - https://console.eu-gb.bluemix.net (Great Britain)
  - https://console.eu-de.bluemix.net (Europe)
  - https://console.au-syd.bluemix.net (Asia-Pacific)

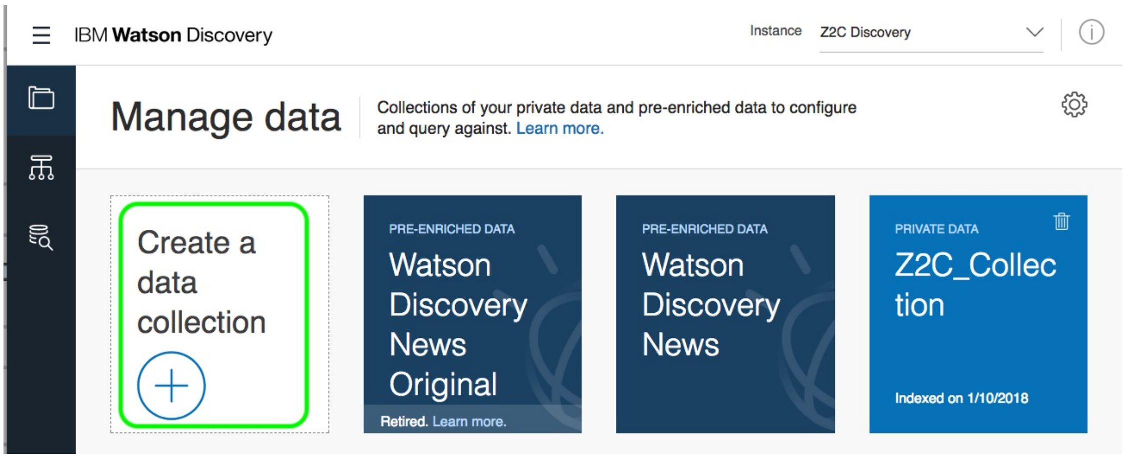
- 

- Create, if you haven't already done so, a Watson Discovery Service (this is the same service you would have used in Chapter 8)
  - Select "Watson" from the left hand menu and then select Discovery from the set of displayed services.
  - 
IBM Cloud
Catalog
Docs
Support
Manage
All Categories
Infrastructure
Compute
Storage
Network
Security
Containers
VMware
Platform
Boilerplates
APIs
Application Services
Blockchain
Cloud Foundry Apps
Data & Analytics
DevOps
Finance
Functions
Integrate
Internet of Things
Mobile
Network
Security
Watson
Search
Filter
Build cognitive apps that help enhance, scale, and accelerate human expertise.
Conversation
Add a natural language interface to your application
Discovery
Add a cognitive search and content analytics engine to
Knowledge Studio
Build custom models to teach Watson the language of your
Language Translator
Translate text from one language to another for
Natural Language Classifier
Natural Language Classifier performs natural language
Natural Language Understanding
Analyze text to extract meta-data from content such as
Personality Insights
The Watson Personality Insights derives insights from
Speech to Text
Low-latency, streaming transcription
Text to Speech
Synthesizes natural-sounding speech from text.
Tone Analyzer
Tone Analyzer uses linguistic analysis to detect three types
Visual Recognition
Find meaning in visual content! Analyze images for
Lite
IBM

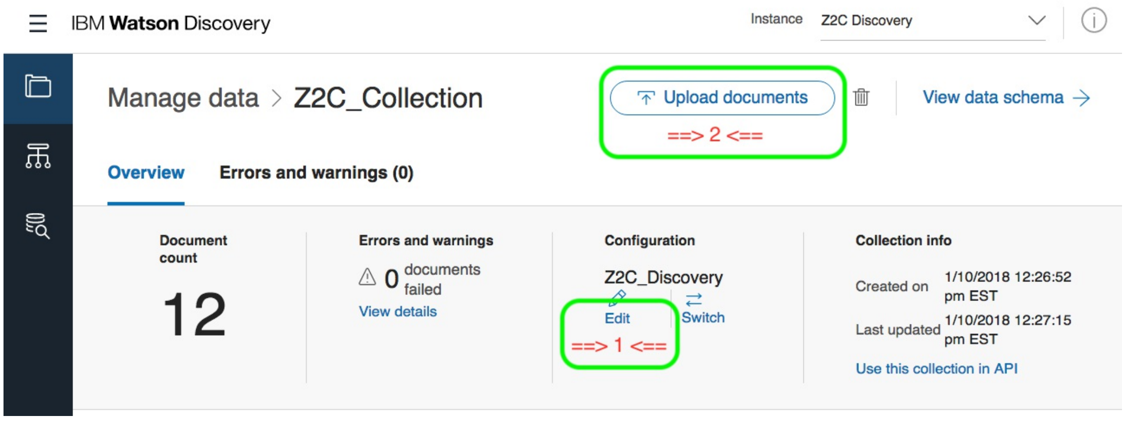
- Click on Service Credentials and then copy your service credentials into your env.json file. In the following file, you'll see environmentID, configurationID, and collectionID. Those are retrieved from the Manage Data page for your collection by clicking on the Use Collection In API link. (this is explained later)
  - (env.json example)

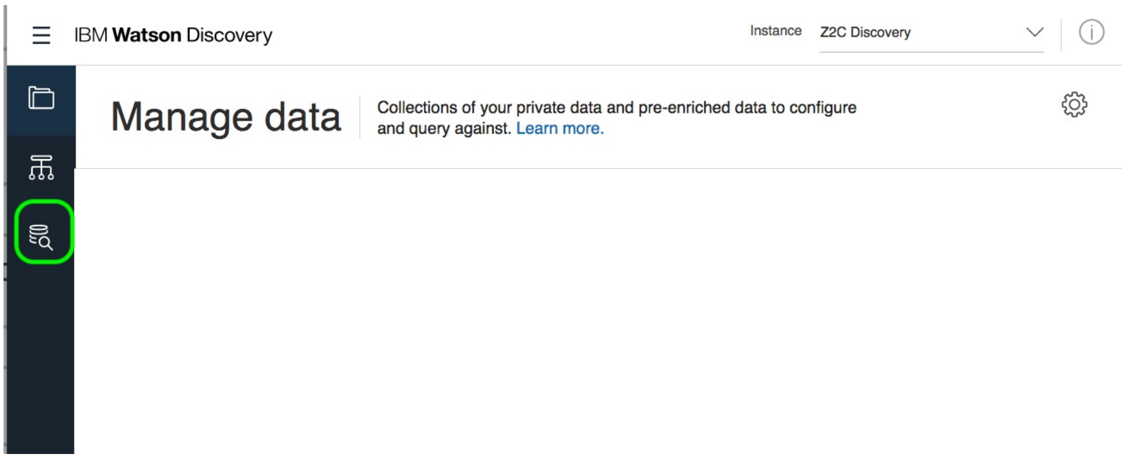
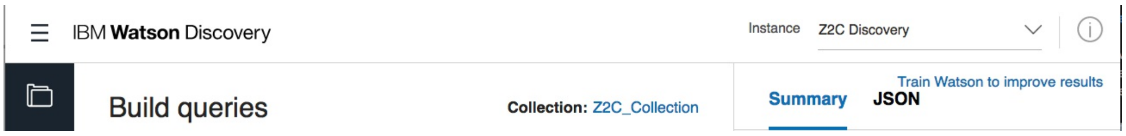
```
        "discovery": {
          "url":
"https://gateway.watsonplatform.net/discovery/api",
          "password": "password-from-credentils",
            "username": "user-name-from-credentials",
            "version_date": "2017-11-07",
            "documentList": "documents.json",
            "source_path": "HTML/Documents/Source",
            "environmentID": "your-environment-id",
            "configurationID": "your-configuration-id",
            "collectionID": "your-collection-id"
```

- 

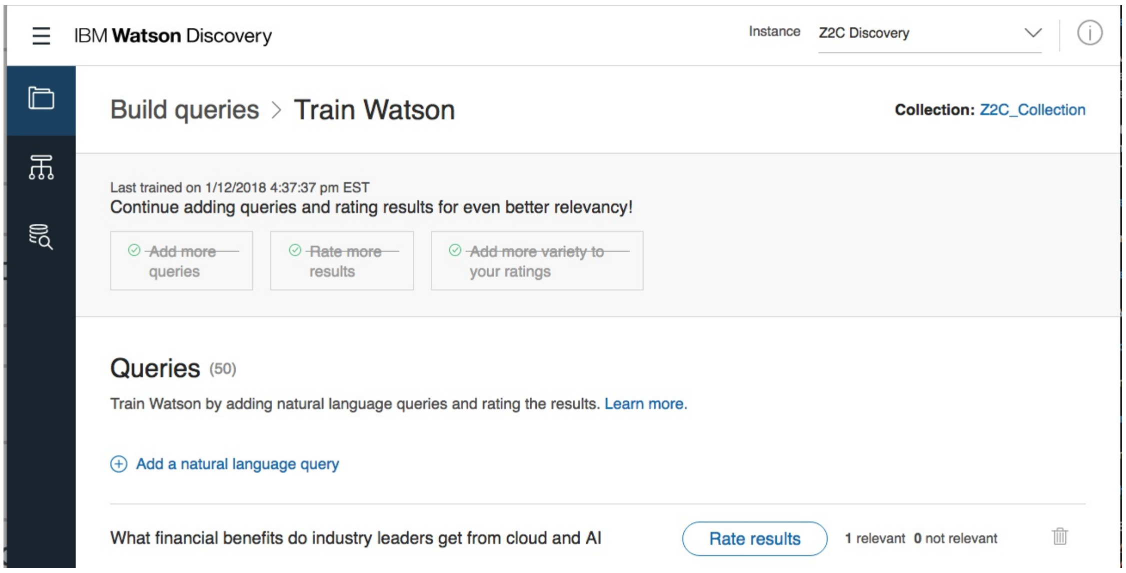
- Then click on Launch Tool and Create a new Data Collection by clicking on the Plus sign.
    - 

    - This will create a private data collection. You see in my example that a private data collection has already been created titled Z2C_Collection
- On the following page,
    - First, Edit the configuration and add in all 'enhancements'. This gives you sentiment, keyword, concepts, etc.
    - 2nd, upload all of your documents from Chapter11/HTML/Document/Source

- 

- We now need to Train Watson Discovery. What's that mean? It means that we need to execute a minimum of 50 different queries against our collection and rank the results from each query. We rank results by taking one of 3 actions:
  - Click on "Relevant" if the returned document is relevant to the query
  - Click on "Not Relevant" if the returned document is particularly not relevant to the query.
  - Not all documents need to be marked, but all queries should be ranked.
  - There is a sample list of 50 queries at the end of this readme file.
- Click on the query icon on the left hand navigator
  - 
IBM Watson Discovery
Instance Z2C Discovery
Manage data
Collections of your private data and pre-enriched data to configure and query against. Learn more.
- And then click on Train Watson in the top right hand corner
  - 

- Which will bring you to the Training Query page. On this page, you'll type or paste in queries and also rank those queries

- 

IBM **Watson** Discovery

Instance Z2C Discovery

Build queries > Train Watson

Collection: Z2C_Collection

Last trained on 1/12/2018 4:37:37 pm EST

Continue adding queries and rating results for even better relevancy!

~~Add more~~ queries

~~Rate more~~ results

~~Add more variety to~~ your ratings

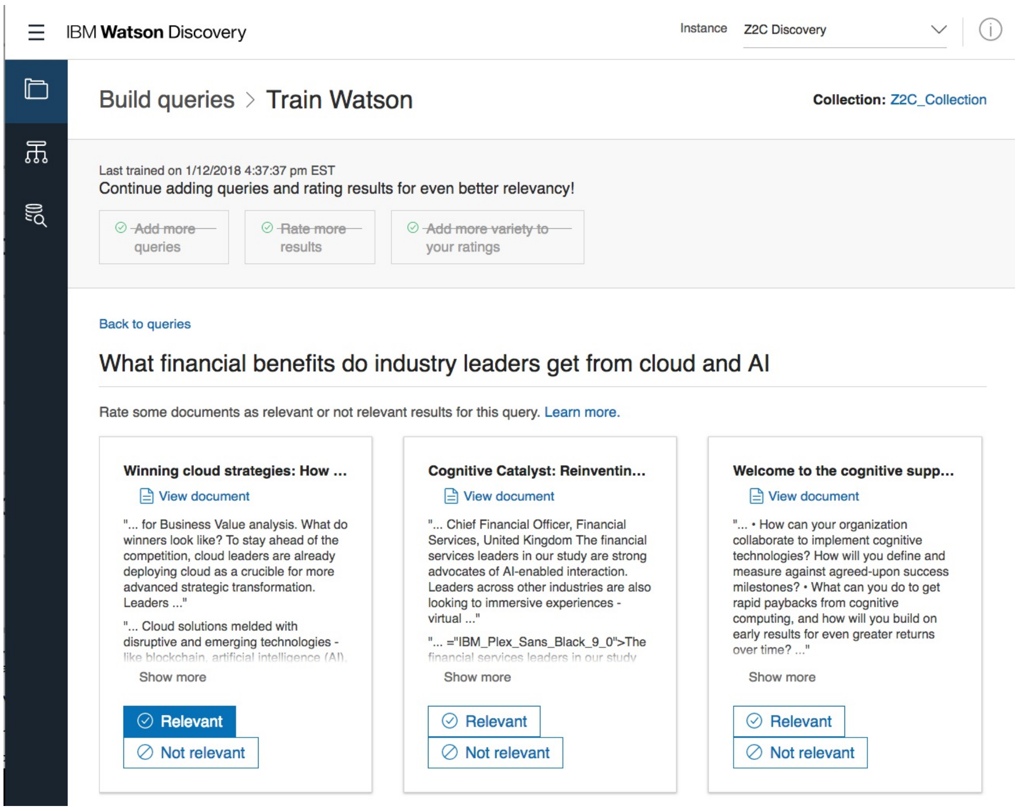
Queries (50)

Train Watson by adding natural language queries and rating the results. Learn more.

Add a natural language query

What financial benefits do industry leaders get from cloud and AI | Rate results | **1** relevant **0** not relevant

- In the middle of the page, you can see three boxes, all of which have a line through them. As you add more queries and provide more ranking and more ranking diversity, these boxes will go from a bold text to what you see here – greyed out with a line through them. when all three look like this, you have completed the minimum required amount of ranking. Watson will now start training. Wait an hour or so after training starts before trying any queries.
- At the bottom of the page, you can see the query 'What financial benefits do industry leaders get from cloud and AI' and a button marked "Rank" to the right.
- To Rank a query, click on the Rank button, which will take you to a page similar to the following:

- ○ 

IBM **Watson** Discovery

Instance Z2C Discovery

Build queries > Train Watson

Collection: Z2C_Collection

Last trained on 1/12/2018 4:37:37 pm EST

Continue adding queries and rating results for even better relevancy!

~~Add more~~ queries | ~~Rate more~~ results | ~~Add more variety to~~ your ratings

Back to queries

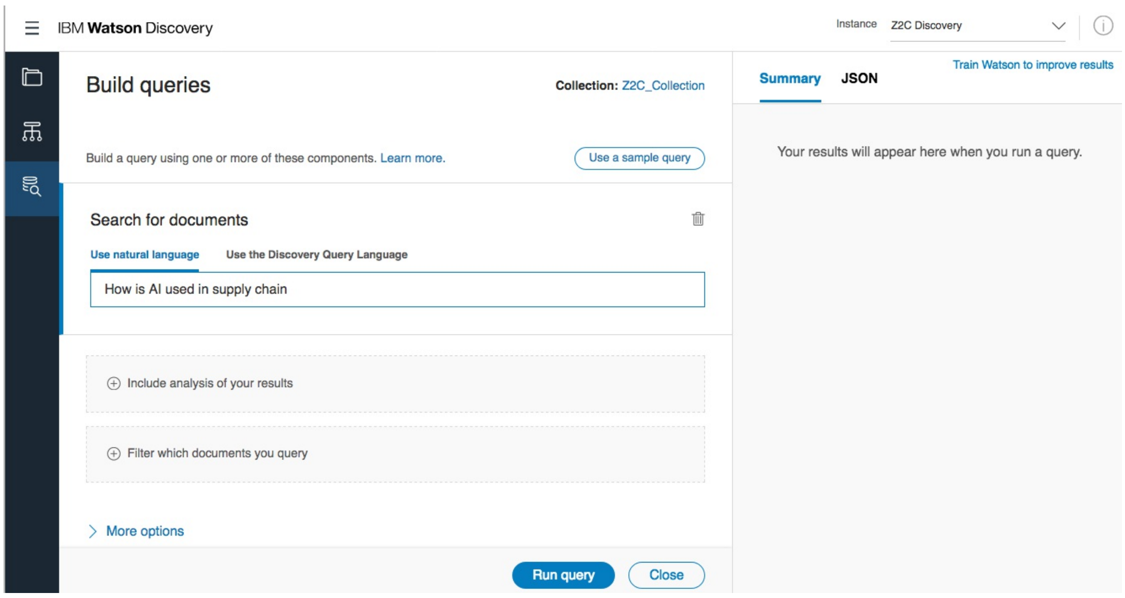
What financial benefits do industry leaders get from cloud and AI

Rate some documents as relevant or not relevant results for this query. Learn more.

Winning cloud strategies: How ...
View document
"... for Business Value analysis. What do winners look like? To stay ahead of the competition, cloud leaders are already deploying cloud as a crucible for more advanced strategic transformation. Leaders ..."
"... Cloud solutions melded with disruptive and emerging technologies - like blockchain, artificial intelligence (AI)
Show more
Relevant
Not relevant

Cognitive Catalyst: Reinventin...
View document
"... Chief Financial Officer, Financial Services, United Kingdom The financial services leaders in our study are strong advocates of AI-enabled interaction. Leaders across other industries are also looking to immersive experiences - virtual ..."
"... ="IBM_Plex_Sans_Black_9_0">The financial services leaders in our study
Show more
Relevant
Not relevant

Welcome to the cognitive supp...
View document
"... • How can your organization collaborate to implement cognitive technologies? How will you define and measure against agreed-upon success milestones? • What can you do to get rapid paybacks from cognitive computing, and how will you build on early results for even greater returns over time? ..."
Show more
Relevant
Not relevant

  - ○ Here, you'll tag documents as "Relevant" or "Not Relevant". Note while not all documents are tagged, all queries should be ranked.
- You can test your collection by going to the Query Page and submitting natural language queries.
  - ○ click on the query icon on the left side and select your collection
  - ○ 

IBM **Watson** Discovery

Instance Z2C Discovery

Build queries

Collection: Z2C_Collection

Build a query using one or more of these components. Learn more.

Use a sample query

Search for documents

Use natural language | Use the Discovery Query Language

How is AI used in supply chain

Include analysis of your results

Filter which documents you query

More options

Run query | Close

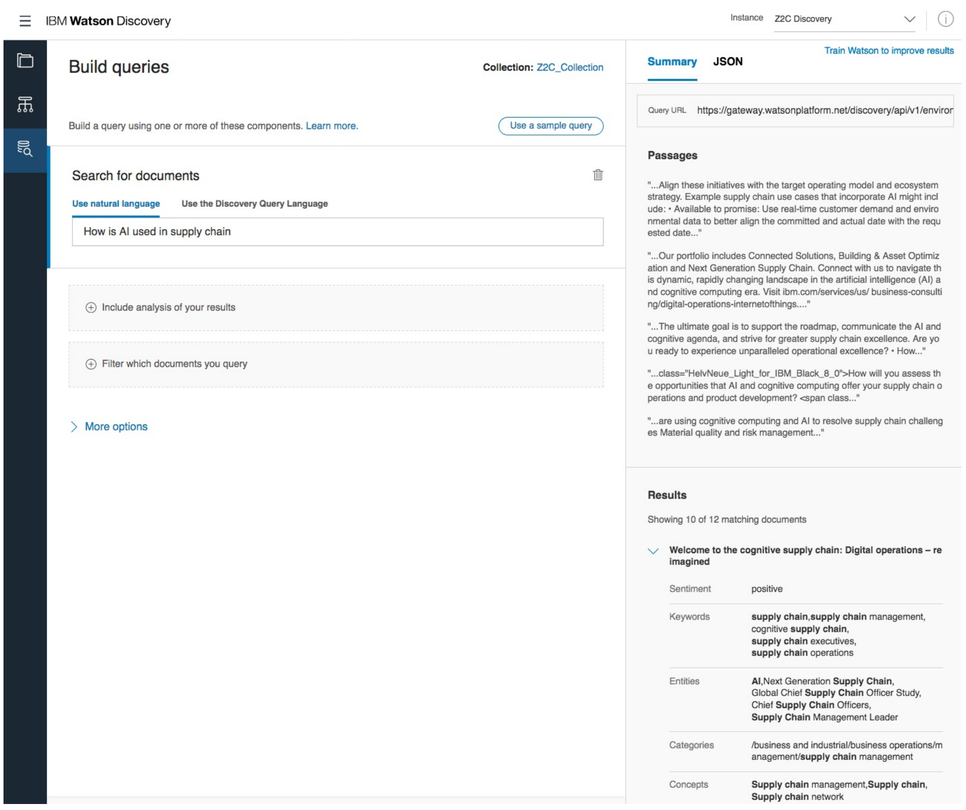
Train Watson to improve results

Summary | **JSON**

Your results will appear here when you run a query.

    - ■ click on "Search for documents" and type in a natural language query

- Click on the "Run Query" button at the bottom of the page

◦ 

IBM Watson Discovery

Instance Z2C Discovery

Build queries

Collection: Z2C_Collection

Build a query using one or more of these components. Learn more.

Use a sample query

Search for documents

Use natural language Use the Discovery Query Language

How is AI used in supply chain

Include analysis of your results

Filter which documents you query

More options

Summary JSON

Train Watson to improve results

Query URL https://gateway.watsonplatform.net/discovery/api/v1/enviror

Passages

"...Align these initiatives with the target operating model and ecosystem strategy. Example supply chain use cases that incorporate AI might incl ude: • Available to promise: Use real-time customer demand and enviro nmental data to better align the committed and actual date with the requ ested date..."

"...Our portfolio includes Connected Solutions, Building & Asset Optimiz ation and Next Generation Supply Chain. Connect with us to navigate th is dynamic, rapidly changing landscape in the artificial intelligence (AI) a nd cognitive computing era. Visit ibm.com/services/us/ business-consulti ng/digital-operations-internetofthings...."

"...The ultimate goal is to support the roadmap, communicate the AI and cognitive agenda, and strive for greater supply chain excellence. Are yo u ready to experience unparalleled operational excellence? • How..."

"...class="HelvNeue_Light_for_IBM_Black_8_0">How will you assess th e opportunities that AI and cognitive computing offer your supply chain o perations and product development? <span class..."

"...are using cognitive computing and AI to resolve supply chain challeng es Material quality and risk management..."

Results

Showing 10 of 12 matching documents

Welcome to the cognitive supply chain: Digital operations – re imagined

Sentiment	positive
Keywords	**supply chain**,**supply chain** management, cognitive **supply chain**, **supply chain** executives, **supply chain** operations
Entities	**AI**,Next Generation **Supply Chain**, Global Chief **Supply Chain** Officer Study, Chief **Supply Chain** Officers, **Supply Chain** Management Leader
Categories	/business and industrial/business operations/management/**supply chain** management
Concepts	**Supply chain** management,**Supply chain**, **Supply chain** network

- explore your results

Creating a Web interface:

- we'll reuse some of the UX work we did in Chapter 8
- we have to create a web page to hold the query and it's results
- we have to create a javascript file to manage the user experience
- We have to create a single service to execute the query

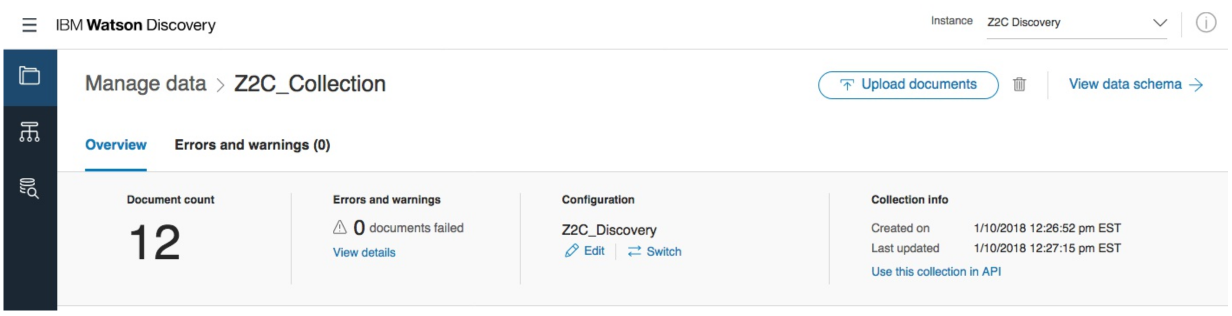
##Creating the web service:

- The discovery_complete.js file has the following routine in it (there are others which you can explore which support creating an administrative interface on the web for Watson Discovery)

```
    /**
    * find queries a specific collection in a
specific environment for the currently identifed
for the userid and password stored in the
```

```
env.json file
    *
    * @param {object} req - nodejs object with
the request information
    * req.body.queryString has the query details
    * @param {object} res - nodejs response
object
    * @param {object} next - nodejs next object
- used if this routine does not provide a
response
    */
    exports.find = function(req, res, next)
    {
    let _method = 'find';
    discovery.query({ environment_id:
config.discovery.environmentID, collection_id:
config.discovery.collectionID,
natural_language_query:
encodeURIComponent(req.body.queryString),
passages: true },
        function (err, response) {
        if (err)
        {console.log('error:', err);
        res.send({'result': 'error', 'message':
err.message});}
        else
        {res.send({'result': 'success', 'data':
response});}
    });
    }
```

- The environment and collection id are gathered programatically from the Watson Discovery Service. If you don't want to get the data programatically, you can copy it from the Watson Discovery Service web interface by going to the displayed page and clicking on "Use this collection in API" and then storing the displayed information someplace like your env.json file.

- 

Manage data > Z2C_Collection

Overview | Errors and warnings (0)

Document count: 12

Errors and warnings: 0 documents failed — View details

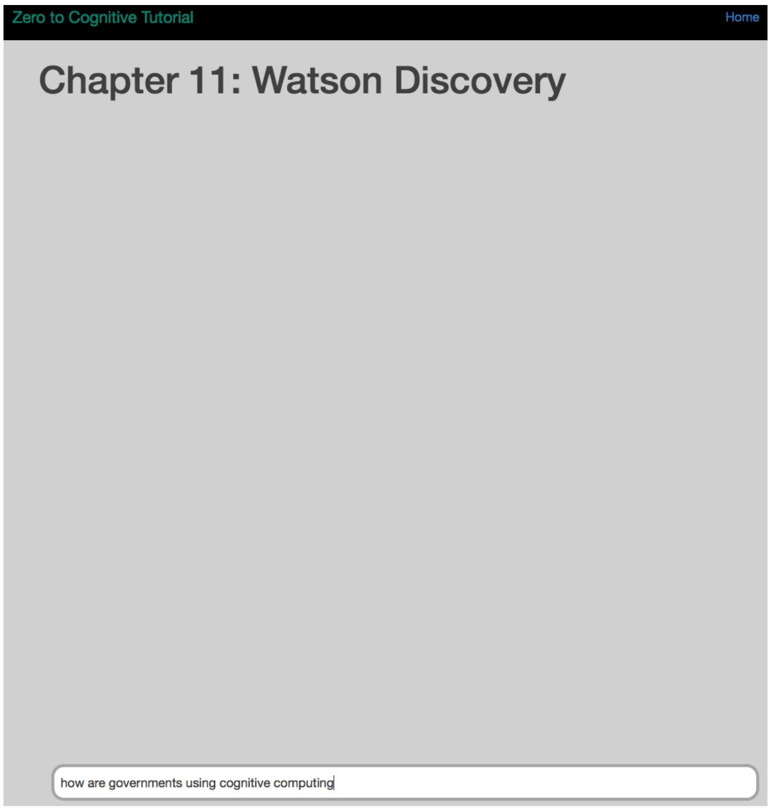
Configuration: Z2C_Discovery — Edit | Switch

Collection info
Created on 1/10/2018 12:26:52 pm EST
Last updated 1/10/2018 12:27:15 pm EST
Use this collection in API

- The web page is very simple. We have a query input file locked to the bottom of the page (we did this in Chapter 10) and will display the results above the input field.

```
          <div class="container">
      <div class="title">
                  <h1>Chapter 11: Watson
Discovery</h1>
      </div>
      <div class="scrollingPane">
          <div class="col-lg-12" id="discovery">
          </div>
      </div>
          <input id="textInput" class="input
showfocus wide" width="100%",
              placeholder="Type a query here"
type="text", onkeypress="detectKey(event,
'doQuery')">
      </div>
```

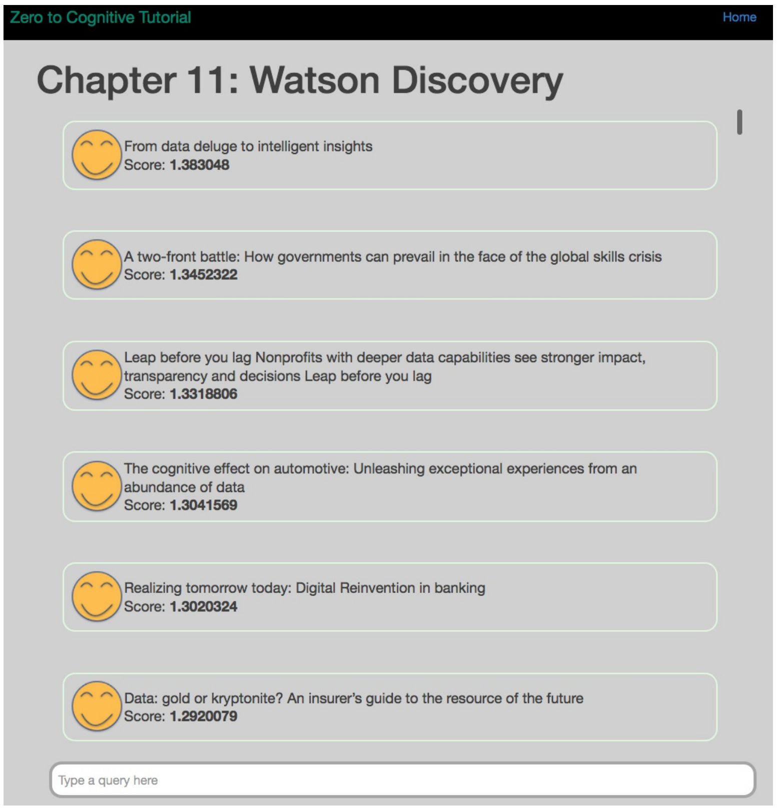
- You may remember the 'detectKey' routine from Chapter10. We've extended it (z2c_utilities.js) by adding a new case statement for this chapter. This enables a person to type a query and execute it simply by pressing the enter key on their keyboard.
    - the initial page looks like this:

- 

Zero to Cognitive Tutorial Home

Chapter 11: Watson Discovery

how are governments using cognitive computing

- Pressing the enter key returns the following scored results with most likely results first
- 

Zero to Cognitive Tutorial Home

Chapter 11: Watson Discovery

From data deluge to intelligent insights
Score: **1.383048**

A two-front battle: How governments can prevail in the face of the global skills crisis
Score: **1.3452322**

Leap before you lag Nonprofits with deeper data capabilities see stronger impact, transparency and decisions Leap before you lag
Score: **1.3318806**

The cognitive effect on automotive: Unleashing exceptional experiences from an abundance of data
Score: **1.3041569**

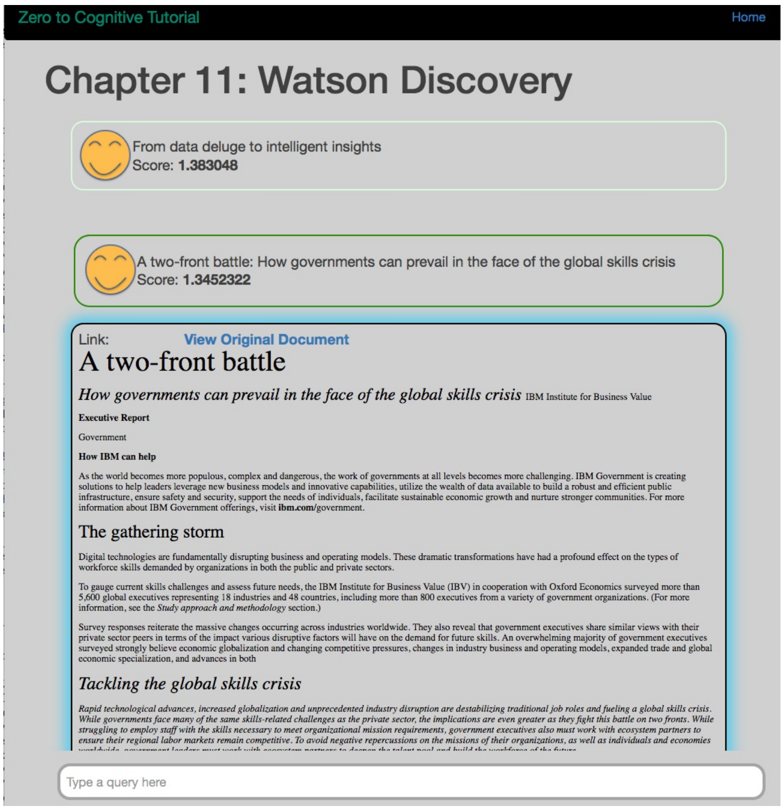
Realizing tomorrow today: Digital Reinvention in banking
Score: **1.3020324**

Data: gold or kryptonite? An insurer's guide to the resource of the future
Score: **1.2920079**

Type a query here

- Clicking on a result line expands the accordian (you'll remember this from Chapter 8) and shows you the text that Watson Discovery stored and, if you kept the documents in your HTML/Documents/Source then clicking on"View Original

Document" will display the original PDF in a new window.

- 
Zero to Cognitive Tutorial
Home

Chapter 11: Watson Discovery

From data deluge to intelligent insights
Score: **1.383048**

A two-front battle: How governments can prevail in the face of the global skills crisis
Score: **1.3452322**

Link: **View Original Document**

A two-front battle

How governments can prevail in the face of the global skills crisis IBM Institute for Business Value

Executive Report

Government

How IBM can help

As the world becomes more populous, complex and dangerous, the work of governments at all levels becomes more challenging. IBM Government is creating solutions to help leaders leverage new business models and innovative capabilities, utilize the wealth of data available to build a robust and efficient public infrastructure, ensure safety and security, support the needs of individuals, facilitate sustainable economic growth and nurture stronger communities. For more information about IBM Government offerings, visit **ibm.com/**government.

The gathering storm

Digital technologies are fundamentally disrupting business and operating models. These dramatic transformations have had a profound effect on the types of workforce skills demanded by organizations in both the public and private sectors.

To gauge current skills challenges and assess future needs, the IBM Institute for Business Value (IBV) in cooperation with Oxford Economics surveyed more than 5,600 global executives representing 18 industries and 48 countries, including more than 800 executives from a variety of government organizations. (For more information, see the *Study approach and methodology* section.)

Survey responses reiterate the massive changes occurring across industries worldwide. They also reveal that government executives share similar views with their private sector peers in terms of the impact various disruptive factors will have on the demand for future skills. An overwhelming majority of government executives surveyed strongly believe economic globalization and changing competitive pressures, changes in industry business and operating models, expanded trade and global economic specialization, and advances in both

Tackling the global skills crisis

Rapid technological advances, increased globalization and unprecedented industry disruption are destabilizing traditional job roles and fueling a global skills crisis. While governments face many of the same skills-related challenges as the private sector, the implications are even greater as they fight this battle on two fronts. While struggling to employ staff with the skills necessary to meet organizational mission requirements, government executives also must work with ecosystem partners to ensure their regional labor markets remain competitive. To avoid negative repercussions on the missions of their organizations, as well as individuals and economies

Type a query here

#Browser Javascript

There are only three routines we create for the browser:

- initiateDiscovery
- doQuery
- display_find
- ***initiateDiscovery***
- This routine simply sets the query source and query result target html objects for later use

```
        let q_source;
        let q_target;

        // initialize the page
    function initiateDiscovery()
```

```
        {
            let _method = 'initiateDiscovery';
            q_source = $("#textInput");
            q_target = $("#discovery");
        }
```

- ***doQuery***

- This routine sets up and executes the asynchronous call the the previously described find service on the server and displays the 'loading' gif we first introduced in chapter 9 (Visual Recognition)

```

        function doQuery()
        {
            let _method = 'doQuery';
            q_target.empty(); q_target.append("
<center><img src='icons/loading.gif' /></center>")
            let _options = {};
            _options.queryString=q_source.val();
            q_source[0].value = "";
            console.log(_method, _options);
            postIt('/discovery/find', _method,
display_find, q_target, _options);

        }
```

- ***display_find***

- this routine extracts part of the returned data, sorts it and then formats the results.

- The opening of the routine creates a temporary array and extracts portions of the returned result set into that array. At the end of the for loop, the array is sorted, based on the document score. The sort is significantly faster with the smaller amount of data to move around memory.

```
        let tmpArr = new Array();
        for (each in _res.data.results)
        {(function(_idx, _arr)
            {
            let tmpObj = {};
            tmpObj.extracted_metadata =
_arr[_idx].extracted_metadata;
            tmpObj.id = _arr[_idx].id;
            tmpObj.sentiment =
_arr[_idx].enriched_text.sentiment.document;
            tmpObj.emotion =
_arr[_idx].enriched_text.emotion.document.emotion;
            tmpObj.result_metadata =
_arr[_idx].result_metadata;
            tmpObj.html = _arr[_idx].html;
            tmpArr.push(tmpObj);
            })(each, _res.data.results);
        }
            tmpArr.sort(function(a,b){return
(b.result_metadata.score > a.result_metadata.score) ?
1 : -1;});
```

- The second for loop formats the output. Document sentiment information is used in an identical manner to Chapter 9. We are using the same accordian structure and CSS as in Chapter 9 and, in fact, nearly identical code.

```
        for (each in tmpArr)
        {
        (function(_idx, _array)
        { console.log('_array['+_idx+'] id is: ',
_array[_idx].id+' name:
'+_array[_idx].extracted_metadata.title+' score:
```

```
'+_array[_idx].result_metadata.score);
            var _hdr = "find_"+_idx+"_header";
            var _bdy = "find_"+_idx+"_content";
            var _sentiment_icon;
            if (_array[_idx].sentiment.label ==
"positive") {_sentiment_icon = '<td><img
src="/icons/positive.png"></td>';}
            if (_array[_idx].sentiment.label ==
"neutral") {_sentiment_icon = '<td><img
src="/icons/neutral.png"></td>';}
            if (_array[_idx].sentiment.label ==
"negative") {_sentiment_icon = '<td><img
src="/icons/negative.png"></td>';}
            var _link = '<tr><td>Link: </td><td><a
href="Documents/Source/'+_array[_idx].extracted_metad
ata.filename+'" target="_blank"><b>View Original
Document</b></a>';
            // since we have this information, let's
display the article summary text in the accordian
window.
            // we're using a table format for display
purposes, so use both columns to display the text
            var _text = '<tr><td
colspan="2">'+_array[_idx].html+'</td></tr>';
            var _hdr_html = '<div class="acc_header
off" id="'+_hdr+'" target="'+_bdy+'"
onClick="accToggle(\'newsfeed\',
\'find_'+_idx+'\');"><table><tr>'+_sentiment_icon+"
<td>"+_array[_idx].extracted_metadata.title+'<br/>Sco
re: <b>'+_array[_idx].result_metadata.score+'</b>
</td></tr></table></div >';
            var _bdy_html = '<div class="acc_body
off" id="'+_bdy+'"><table>'+_link+_text+'</table>
</div>';
            _target.append(_hdr_html+_bdy_html);
        })(each, tmpArr);
        }
```