Course Guide

IBM BigInsights Text Analytics (v4)

Course code DW654 ERC 1.0

IBM Training

December 2015

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, ibm.com and BigInsights are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, and the Adobe logo, are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Contents

Course overview

Preface overview

This course is designed to introduce the student to the capabilities of BigSheets. BigSheets is a component of IBM BigInsights through the Analyst and the Data Scientist module. It provides the analyst the ability to be able to visualize and analyze data stored on the HDFS using a spreadsheet type interface without any programming.

Intended audience

The course is designed for business analysts that does not want to deal with any coding to get insight on their data.

Topics covered

Topics covered in this course include:

IBM BigInsights Text Analytics:

- Text Analytics overview
- Task analysis
- Annotation Query Language
- Candidate generation
- Filtering and consolidation
- Working with pre-built extractors

Course prerequisites

Participants should have:

- Students should be familiar with Hadoop and the Linux file system.
- Although not required, it would also be helpful for students to take the DW644 - IBM BigInsights BigSheets course to have a better understanding of how BigSheets can be used with Text Analytics.
- Students can attend many free courses at www.bigdatauniversity.com to acquire the necessary requirements.

Document conventions

Conventions used in this guide follow Microsoft Windows application standards, where applicable. As well, the following conventions are observed:

- **Bold**: Bold style is used in demonstration and exercise step-by-step solutions to indicate a user interface element that is actively selected or text that must be typed by the participant.
- *Italic*: Used to reference book titles.
- CAPITALIZATION: All file names, table names, column names, and folder names appear in this guide exactly as they appear in the application.
  To keep capitalization consistent with this guide, type text exactly as shown.

Additional training resources

- Visit IBM Analytics Product Training and Certification on the IBM website for details on:
  - Instructor-led training in a classroom or online
  - Self-paced training that fits your needs and schedule
  - Comprehensive curricula and training paths that help you identify the courses that are right for you
  - IBM Analytics Certification program
  - Other resources that will enhance your success with IBM Analytics Software
- For the URL relevant to your training requirements outlined above, bookmark:
  - Information Management portfolio: http://www-01.ibm.com/software/data/education/
  - Predictive and BI/Performance Management/Risk portfolio: http://www-01.ibm.com/software/analytics/training-and-certification/

IBM product help

Help type	When to use	Location
Task-oriented	You are working in the product and you need specific task-oriented help.	*IBM Product* - Help link
Books for Printing (.pdf)	You want to use search engines to find information. You can then print out selected pages, a section, or the whole book. Use Step-by-Step online books (.pdf) if you want to know how to complete a task but prefer to read about it in a book. The Step-by-Step online books contain the same information as the online help, but the method of presentation is different.	Start/Programs/*IBM Product*/Documentation
IBM on the Web	You want to access any of the following:	
	• IBM - Training and Certification	• http://www-01.ibm.com/software/analytics/training-and-certification/
	• Online support	• http://www-947.ibm.com/support/entry/portal/Overview/Software
	• IBM Web site	• http://www.ibm.com

IBM Training

IBM BigInsights Text Analytics

- Text Analytics overview
- Task analysis
- Annotation Query Language
- Candidate generation
- Filtering and consolidation
- Working with pre-built extractors

IBM Training

Text Analytics Overview

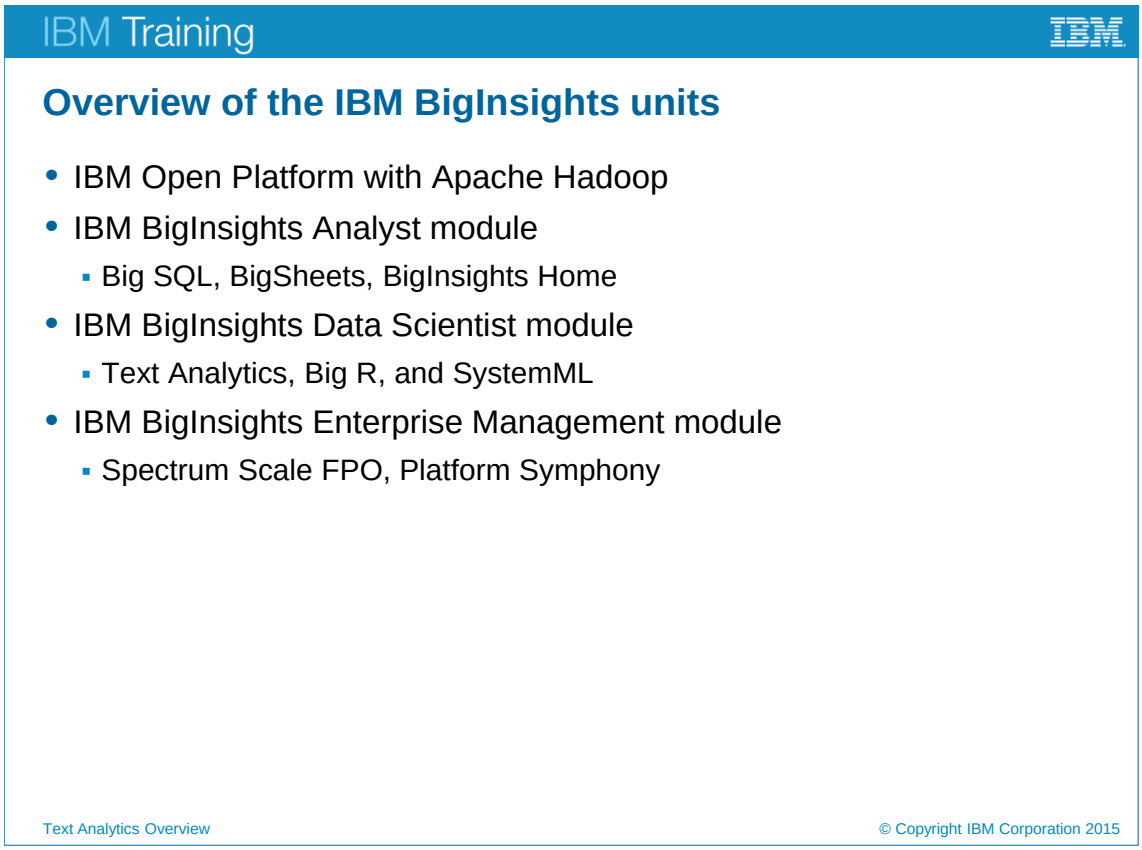
IBM BigInsights v4.1

IBM Training

Unit objectives

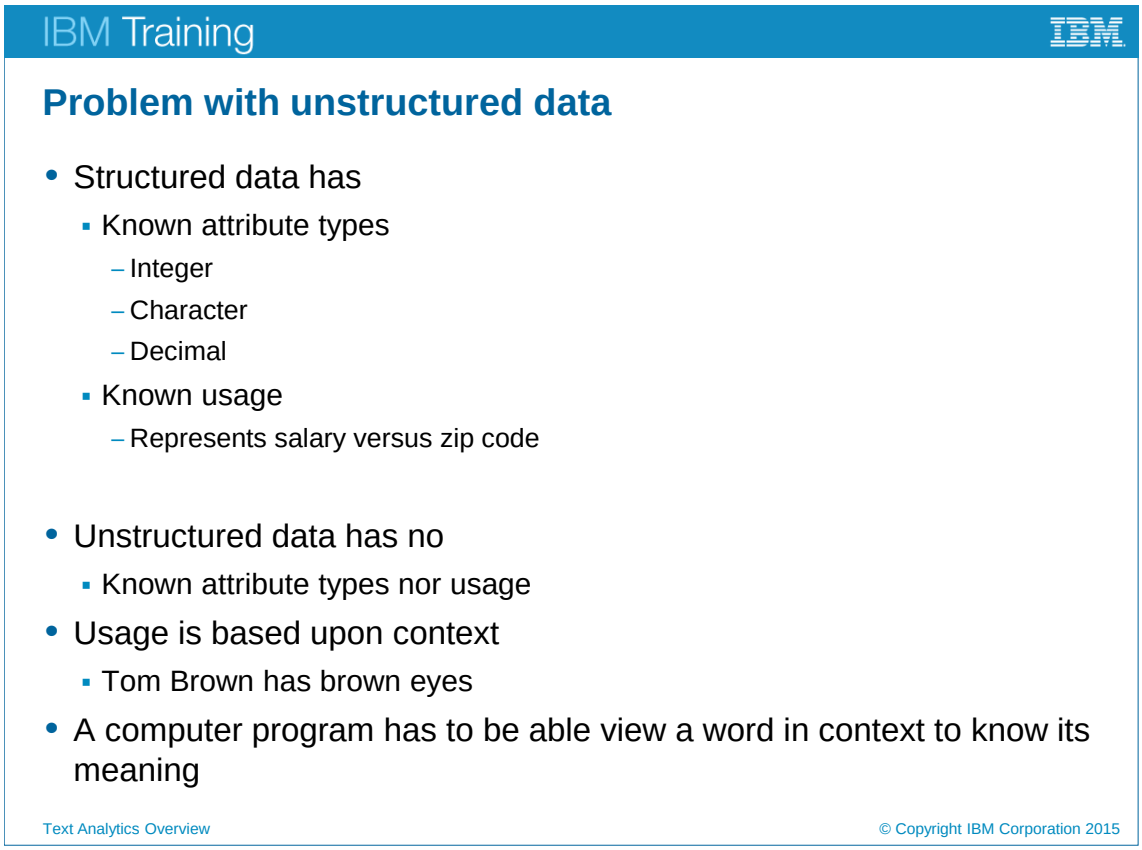
- Overview of the BigInsights module
- Compare structured vs unstructured data
- Understand how to design your project
- Describe and list the steps used for text analytics

Unit objectives



Overview of the IBM BigInsights units

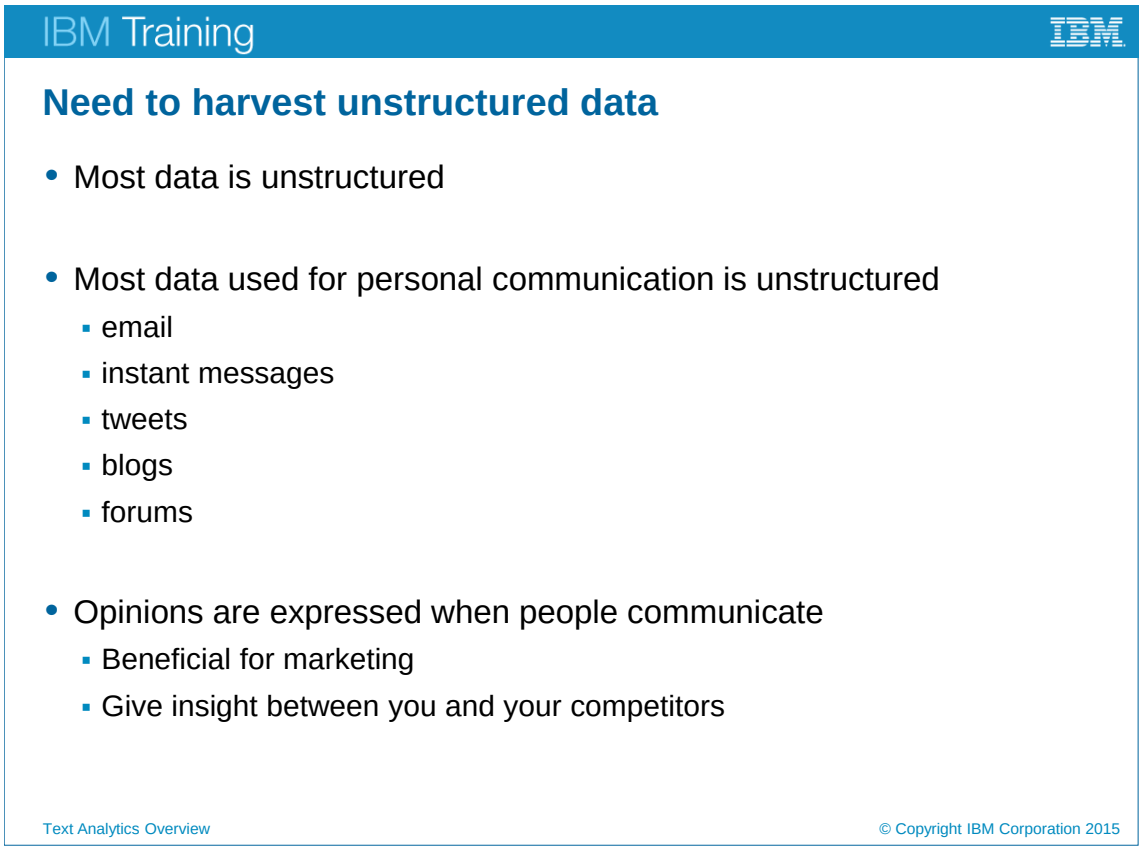
At this point, you probably are well aware and versed in the BigInsights offering units that were introduced early this year. If not, this slide will get you up to speed. First of all, every BigInsights installation will require that you install the IBM Open Platform (IOP). With that, you can decide to use one of the three IBM value add units created to suit the various job roles and users of BigInsights. The first one on the list is the BigInsights Analyst unit, where you can find Big SQL, BigSheets and the BigInsights Home page. The Data Scientist unit has our Text Analytics, Big R, and SystemML. The Enterprise Management unit has the Spectrum Scale FPO (formerly GPFS – FPO) and Platform Symphony.



Problem with unstructured data

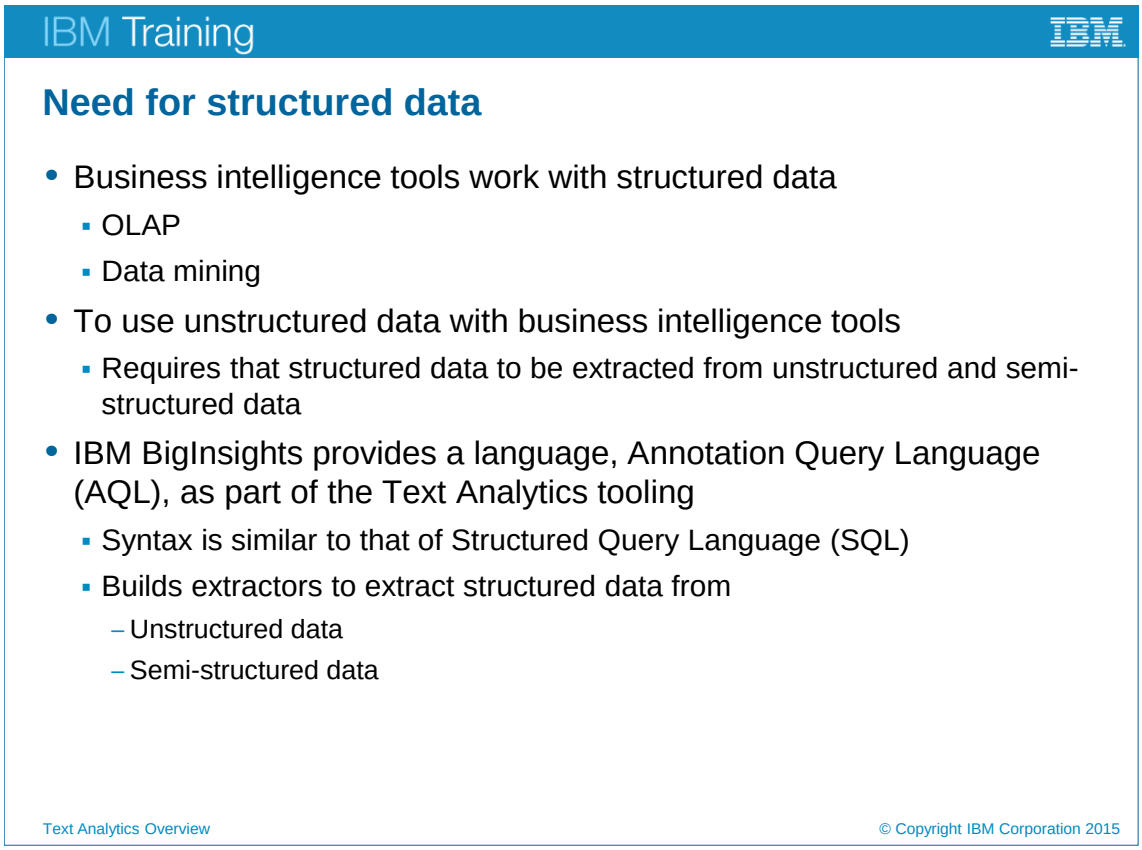
Computers have been working with structured data from the outset. With structured data you know its attribute type, integer, character, decimal, and you know its usage, does it represent a salary value or a zip code. Armed with this knowledge, your program can process the data in a meaningful way.

But unstructured data, by its very nature, does not have known attribute types and data usage. The only way to discern that information is base upon the context usage of the data. For human beings, this usually does not present a problem. But writing a program to do that is extremely difficult. Take the phrase, "Tom Brown has brown eyes." It is easy for us to understand that *Brown* is a proper noun and *brown* is an adjective. But context is very difficult for a program to understand. To a program the two words are essentially identical strings of characters.



Need to harvest unstructured data

Although computers have been working with structured data for close to sixty years, the amount of stored structured data is minute when compared to unstructured data. Just think about the number of emails, instant messages, tweets, and word processing documents that are created on a daily basis. Then think about the number of web pages that exist, the number of blogs and forums. And all of that data is unstructured or semi-structured. (I am not even going to attempt to go down the path of video and audio files.) If all of that unstructured data was meaningless, then even though there is a large amount of it, we would not bother with it. But it is not meaningless. The types of data that I listed are used for communication. People express opinions when they communicate. And opinions are very important when it comes to marketing. So what if you were able to extract opinions about your products from all of this communication data. That would give you a good insight on what customers thought about your products and also how they compared your products with your competitor's.



Need for structured data

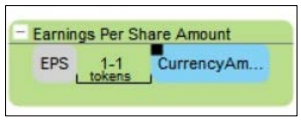
The need is to extract structured data from unstructured and semi-structured data. Why? Because business intelligence tools, that allow for data analysis, use structured data. This goes for OLAP (online analytical programming), data mining and even for simple spreadsheet analysis. IBM BigInsights provides a language called Annotation Query Language (AQL) that is designed to build extractors to extract structured data from both unstructured and semi-structured data. The syntax of the language was modeled after Structured Query Language (SQL) in order to lessen the learning curve.

IBM Training

Design your project

- Understand the general structure and flow of text to be extracted
- Work from the bottom up
- Identify the columns needed in the results
- Work in an iterative manner
- Refine the results

- Example:
  - In the text strings, "The EPS was $1.64" and "The EPS is $ 1.64", the term "EPS" is a keyword that identifies the dollar value in the sentence as earnings per share. You can define an extractor as a sequence with the literal "EPS" followed by a currency amount within one token or word



Text Analytics Overview

Design your project

Understand the general structure and flow of text to be extracted.

- Look for recurring patterns in text to be extracted, avoided, or used to provide context for the target terms.
- Identify keywords specific to your use case.
- Identify numeric and alphanumeric strings to be extracted using a character pattern.
- Estimate the amount of text to be included in the results

Work from the bottom up.

- Start small. Extract as much text as possible using the provided extractors, regular expressions and dictionary extractors.
- Test. Run each extractor component individually to validate results.
- Refine results. Remove or add output columns and specify column names to be used in the results.
- Build incrementally. Combine these basic extractors into sequences and unions, one step at a time.

Identify the columns needed in the results.

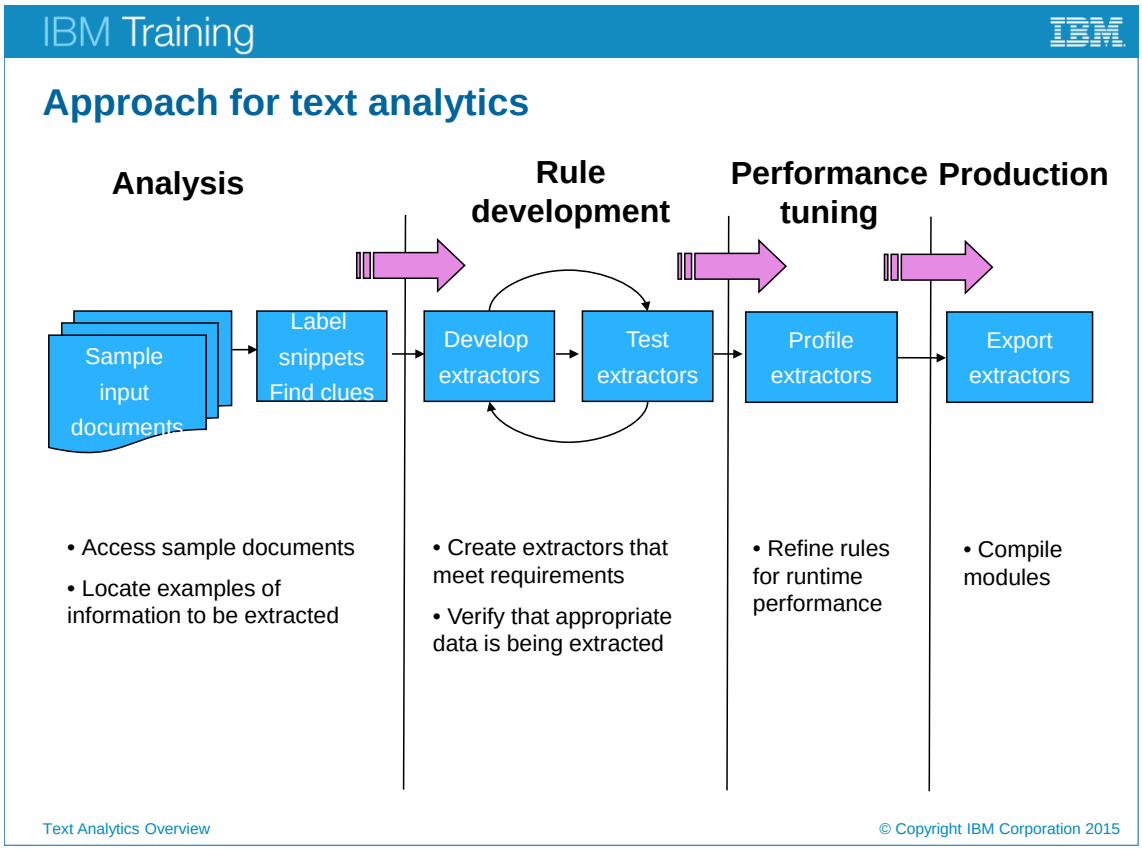
- Identify the name and the contents for each column. For example, for currency values, do you want the currency symbol and the number in one column or separated into two columns?
- Note the matched text from the source document to validate results and support audit activities when the results are later joined with other data sources.

Work in an iterative manner.

- To expedite troubleshooting, add an extractor, run it, check the results, refine and repeat, rather than drag a number of extractors onto the canvas, combine them, and run the resulting extractor.

Refine the results.

- Rename output columns, define filters and consolidate rows to achieve the desired results



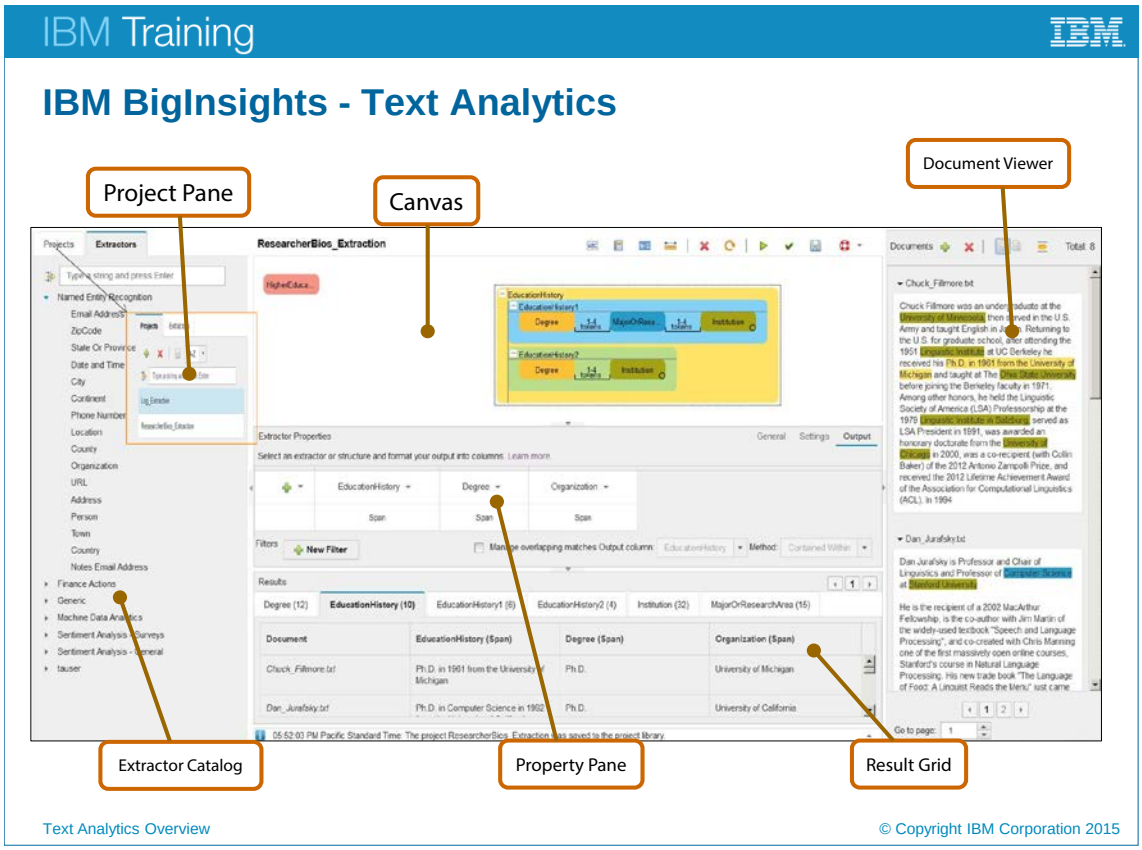
Approach for text analytics

Here we get into the approach for text analytics. There are four phases shown here. Analysis. Rule development. Performance tuning. Production.

In the first phase, you take a look at a few sample documents that are somewhat representative of your entire dataset. From those documents, you label snippets and find clues to locate examples of the information to be extracted. In this phase, you would generally enlist the help of subject matter experts that are well familiar with the documents to find better clues to aid in the text extraction.

Then, in the second phase, you develop and test your extractors. You stay in this phase until you are satisfied that the extractors are able to provide the information that meets your requirements.

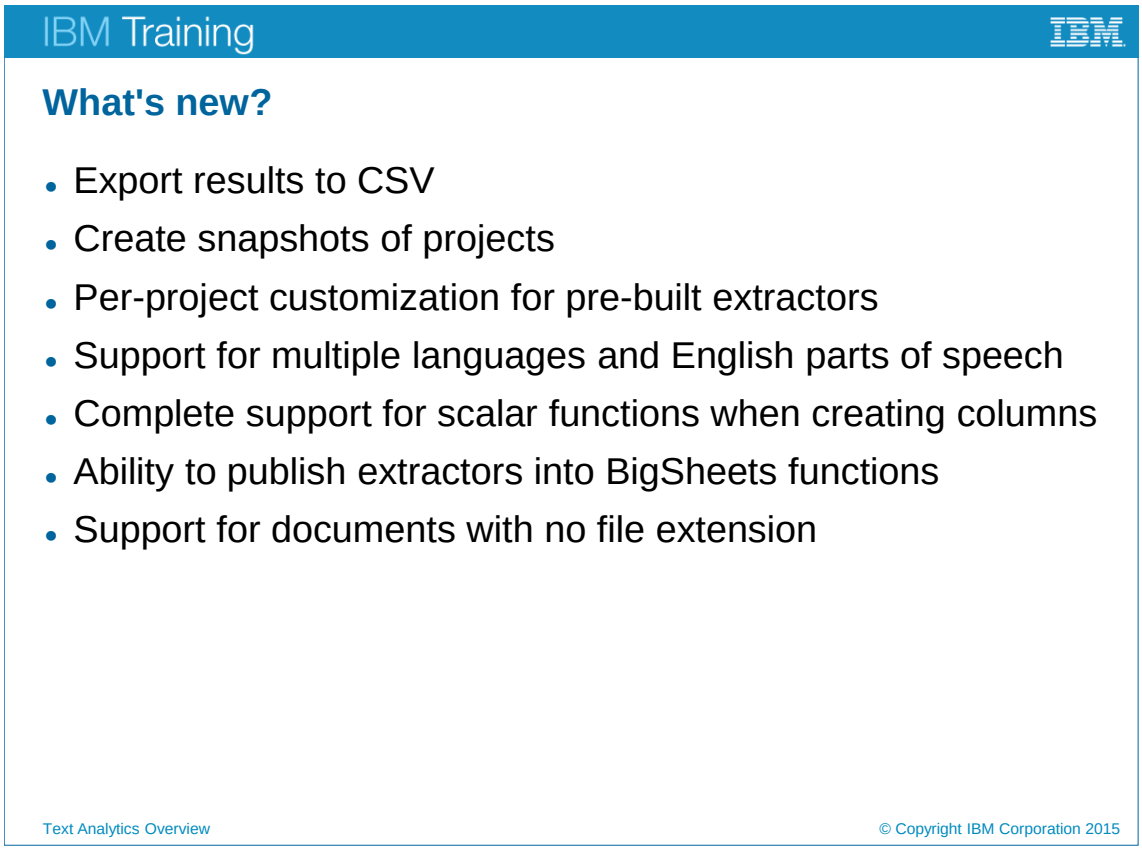
The next phase is performance tuning, where you refine the rules for runtime performance. Finally, you export the extractors and use them with tools like BigSheets to perform your analysis.



IBM BigInsights - Text Analytics

If you haven't seen the new Text Analytics Web UI until now, here it is. Once of the nice thing about this tool is that you do not have to know anything about the language underneath. It is using the Annotation Query Language (AQL), but the UI generates all that for you, making it easy for you to do what you need, without having to mess with the coding beneath.

Essentially, you will be using this interface for all your Text Analytics work. On the left side is the Project Pane and the Extractor Catalog. The Project Pane lists all your current projects. Selecting a project from there will load all and any extractors that is part of the project onto the Canvas, which is right in the center of the UI. The Canvas is where you will be doing a lot of your work. You organize, manage, and manipulate extractors on the Canvas. When you click on any of the extractors within the Canvas, the extractor's properties appear beneath. The Property Pane is used for editing the properties of the extractors. If you are using a pre-built extractor from the Extractor Catalog, you can customize them for your particular project via the Property Pane. The Result Grid sits below the Property Pane and shows the result of the Extractor when you run it against a set of documents. Documents are shown on the right side on the Document Viewer. All of these panes can be resized, expanded/collapsed as needed. In the demonstration, you'll have to resize and/or expand and collapse the panes in order to see everything.



What's new?

Here are some of the new features with Text Analytics in the V4.1 release of IBM BigInsights. You have the option to export the results of your extractors to CSV. This allows you to use other tools to continue with your analysis. Creating snapshot of projects is something is was introduced in this release to allow you to rollback your extractors as you need. Each of the pre-built extractors can be customized per project allowing a lot of flexibility. There is now support for multiple languages and English parts of speech (more on this on the next slide). There is now complete support for scalar functions when creating columns. Now you also have the ability to export your extractors as a BigSheets function. Finally, there is support for documents with no file extension as well. Visit the knowledge center for the V4.1 release to find more information regarding these new features.

IBM Training

Multilingual support

- Use of a multilingual tokenizer
  - Languages that do not use white space tokenization, such as Chinese languages
  - Allows use of English Parts of Speech
  - Does not allow all of the pre-built extractors with other languages
- Set up in the Ambari Home Page
  - Under the Text Analytics service → Configs → Advanced ta-web-tooling-config. Type 'multilingual' or 'standard' to switch between the different tokenization
- Standard tokenization is much faster than multilingual

Text Analytics Overview

Multilingual support

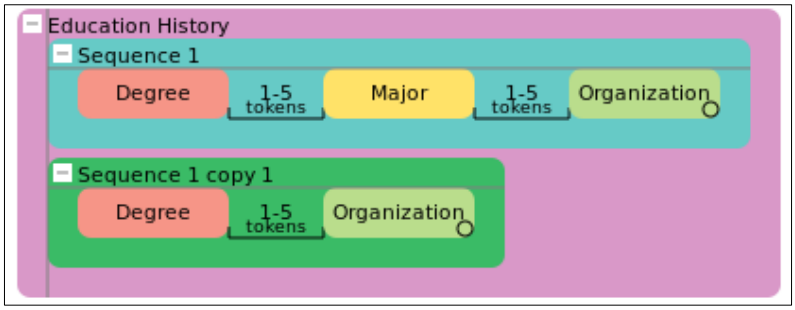
In this release of the Text Analytics tooling, there is multilingual support. Essentially, support for languages that do not use white space tokenization, such as Chinese languages. It allows the use of English parts of speech. However, this does not allow the use of pre-built extractors with other languages. You have to use the multilingual tokenizer for all other languages.

By default, the standard tokenization is much faster than multilingual, so you need to set this option in the Ambari Home page if you need the multilingual tokenizer.

IBM Training

Demonstration 1

Extract education histories from biographies



Text Analytics Overview

Demonstration 1: Extract education histories from biographies

Demonstration 1: Extracting education histories from biographies

Purpose:
The purpose of this demonstration is to give you an end-to-end feel of how to use BigInsights Text Analytics to analyze text data. In subsequent units and demonstrations, you will get to work with the individual components to better understand how to use Text Analytics.

User ids / Passwords

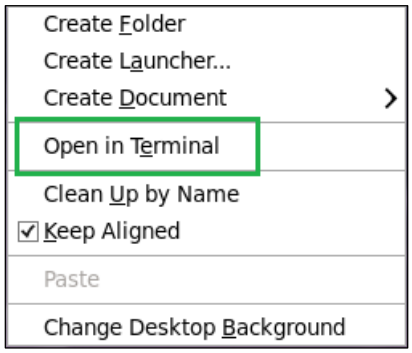
OS:	**biadmin/biadmin**
Root:	**root/dalvm3**
Ambari:	**admin/admin**
BigInsights Home:	**guest/guest-password**

Ambari Services Required:

- HDFS
- MapReduce2
- YARN
- Knox (also start the Demo LDAP service)
- BigInsights - Text Analytics
- BigInsights - Home

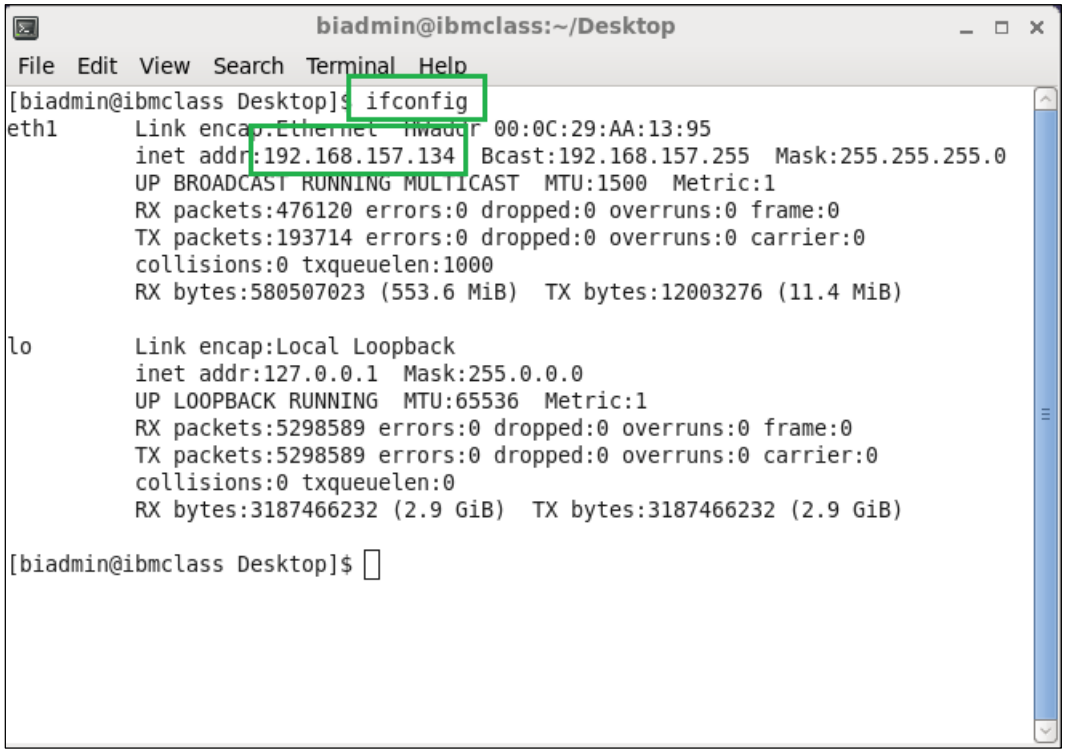
Task 1. Starting up the required services via Ambari.

1. Login to the OS using **biadmin/biadmin**.
2. Once the OS has loaded, verify that the assigned IP address matches that in the /etc/hosts file. To do so, open up a new Terminal. Right-click somewhere on the desktop and select **Open in Terminal**.



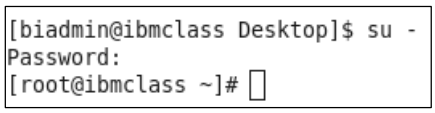
3. In the terminal window that appears, type in:
   `ifconfig`

4. Note the ip address that has been assigned. In the next few steps, you will update the /etc/hosts file if the ip address listed isn't the same as what is shown as a result of ifconfig.



5. Switch to the root user using the password **dalvm3**. Type in:

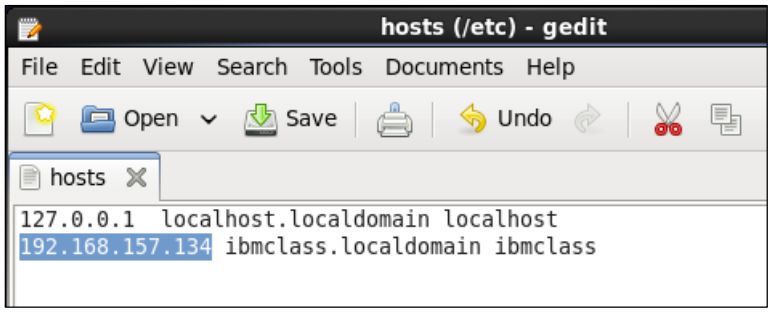
su -



6. Use your favorite text editor to open up the **/etc/hosts** file. I will be using gedit. Type in:

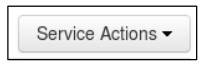
```
gedit /etc/hosts
```

7. Update the ip address to that of which was listed when you ran the *ifconfig* command.

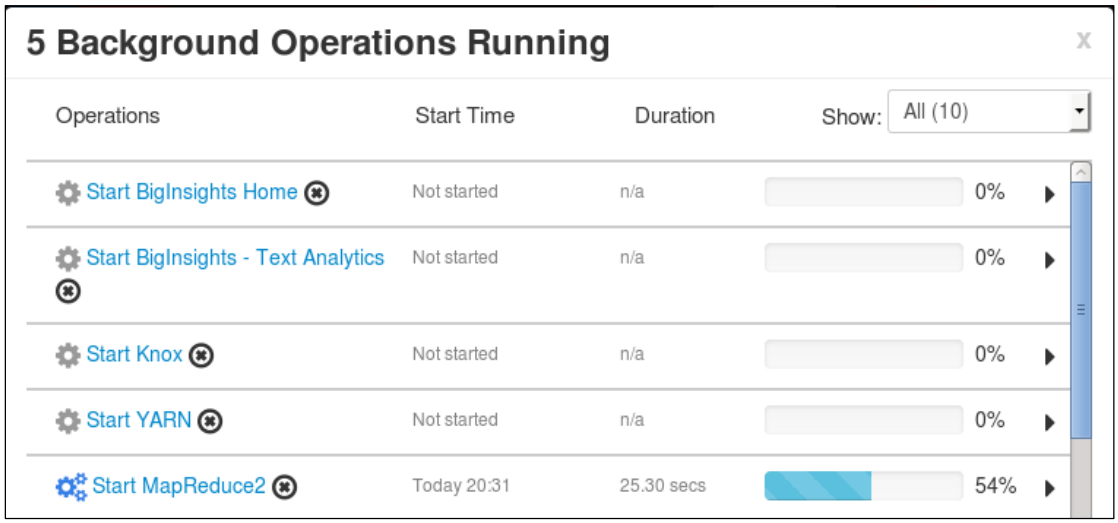


8. Save and close that file.
9. Close the terminal window.

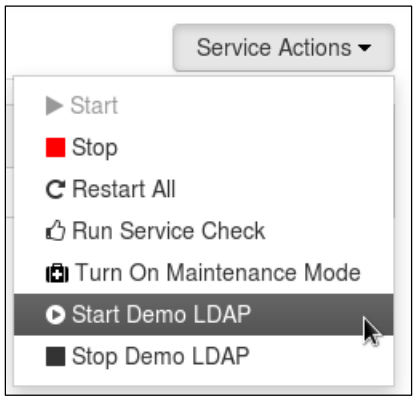
10. Open the **Firefox** web browser.
11. The login to **Ambari** is **admin/admin**. Go ahead and log in.
12. All the services on the left side should be showing a red triangle with an exclamation mark inside  .
    This means that the services have not been started. If it is a yellow icon, it means that the IP address was not resolved since you updated the */etc/hosts* file, so you need to wait until it turns red before you can start up the services.

    Ambari Services Required:
    HDFS
    MapReduce2
    YARN
    Knox (also start the Demo LDAP service)
    BigInsights - Text Analytics
    BigInsights - Home
    Start up the services listed above by doing the following:
    - Select the service from the left side panel.
    - Click the **Service Actions** button from the right side. 

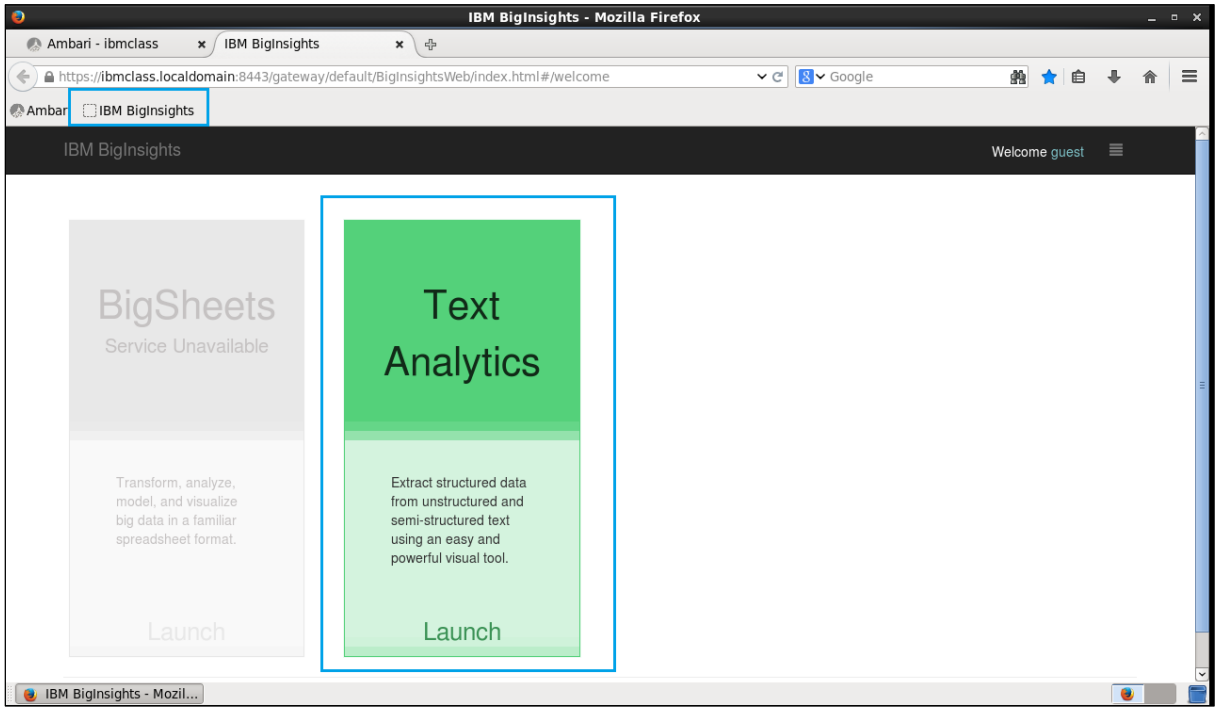
    - Depending on the service, you will have a number of options. Select the **Start** action to start up that service.
    - Confirm the start action in the popup.
    - Do the same thing for all the other services listed.
    - You can queue up each of these actions and view them in the background operations queue.



- When all of these services have started, click on the Knox service to start up start the **Demo LDAP** service under its **Service Action** menu. You need this service to authenticate into the BigInsights Home page.



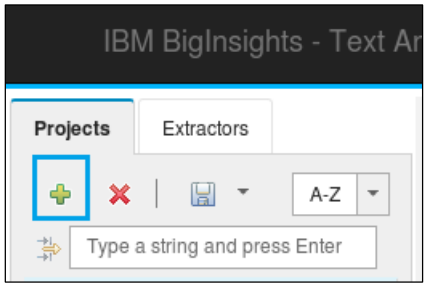
13. Once that has started, open up a new browser tab.
14. Navigate to the **BigInsights Home** page:
    https://ibmclass.localdomain:8443/gateway/default/BigInsightsWeb/index.html
    There is a bookmark already set up with this. The bookmark name is **IBM BigInsights**
15. Log in using **guest/guest-password**.



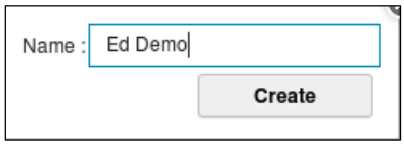
16. Click the **Text Analytics** link to bring up the Web UI.

Task 2. Creating a new Text Analytics project.

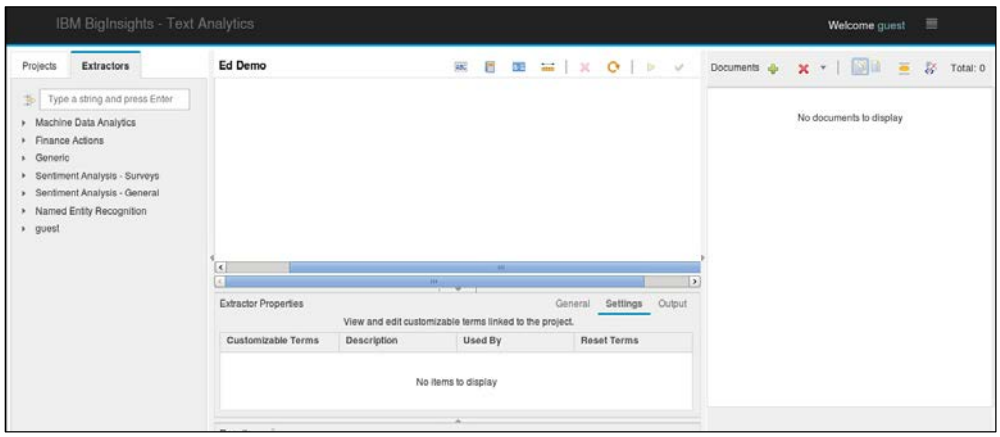
1. On the Projects tab on the left, create a new project. Click the **green plus** icon.



2. Name the project **Ed Demo.**

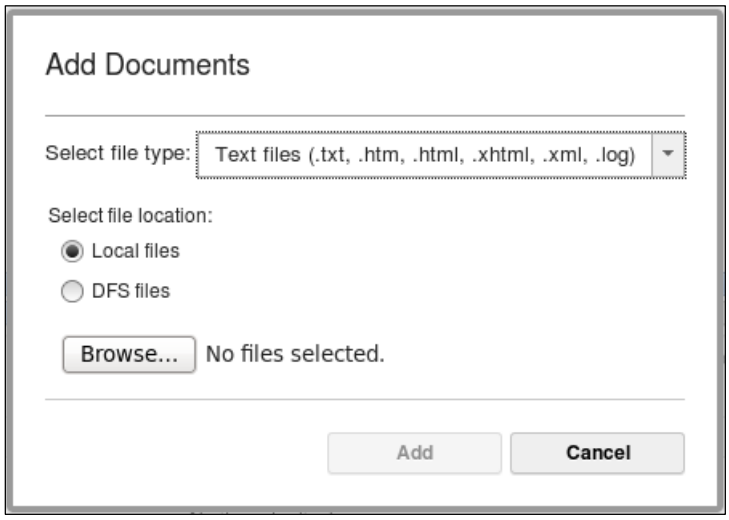


3. Click the **Create** button to create the project.

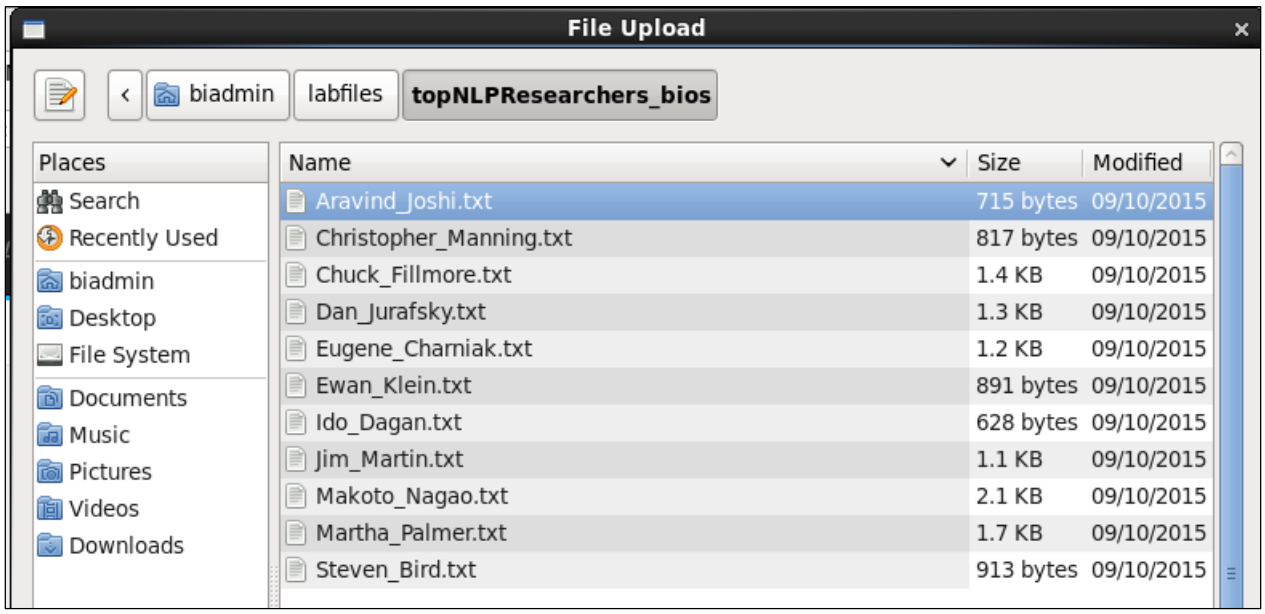


Task 3. Importing documents into the Web UI.

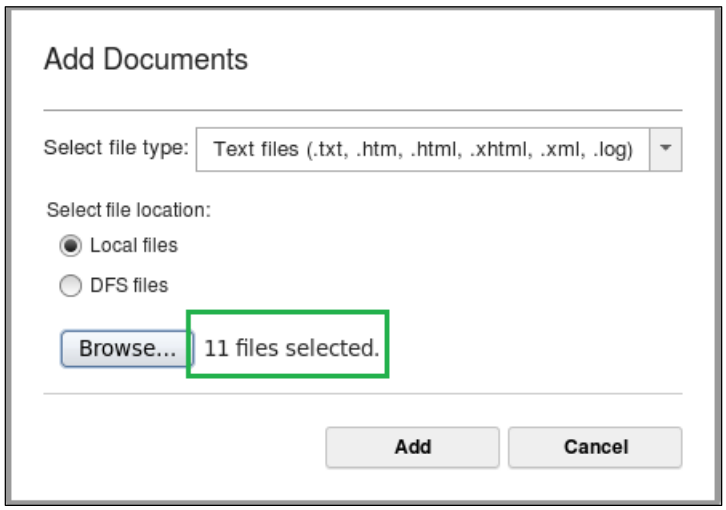
1. On the **Documents** pane on the right. Click the **green plus** to add documents to the project.



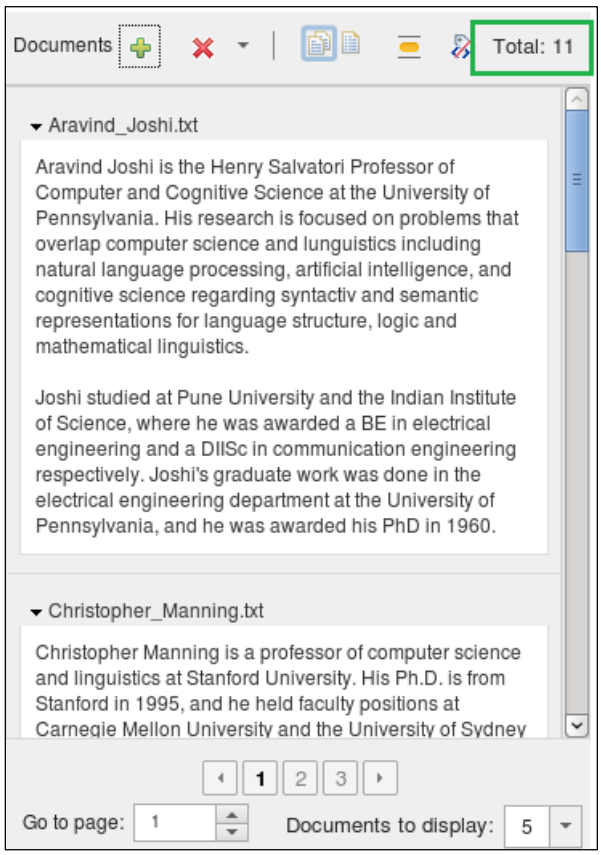
2. The file type is **Text files**. In our case, the files are located on the local filesystem. Alternatively, you can load files that are residing on your Distributed File System (DFS). Click **Browse…** to select your files.
3. Navigate to */home/biadmin/labfiles/topNLPResearchers_bios/*



4. Select all 11 files and click **Open.**

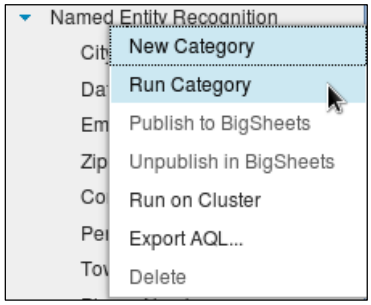


5. Finally, click **Add** to load the files into your project.
6. The files will show up on the **Documents** pane and are ready to be used.

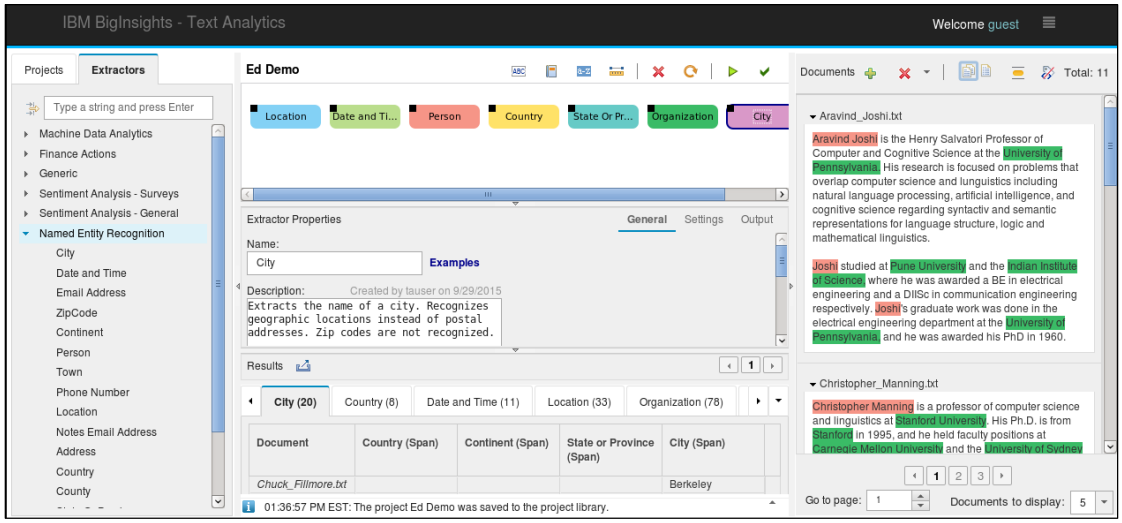


Task 4. Running your first extractor.

1. Under the **Extractors** tab, expand the **Named Entity Recognition** category to see all the pre-built extractors underneath.
2. Run all of the extractors underneath the **Named Entity Recognition** category. Right-click the category and select **Run Category**.

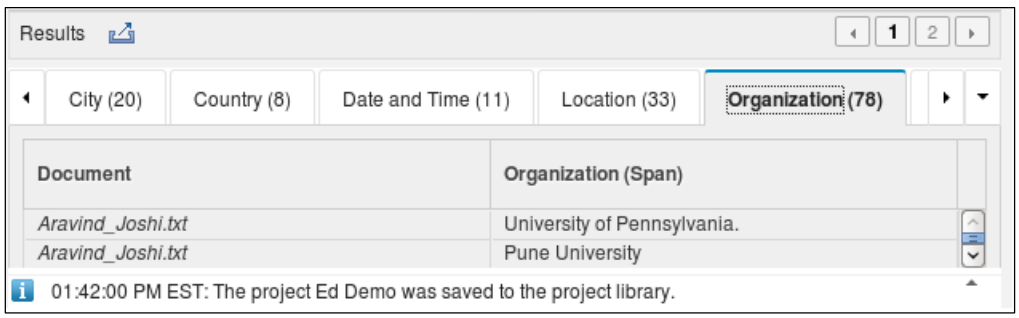


3. When the run finishes, you will see something similar to this:

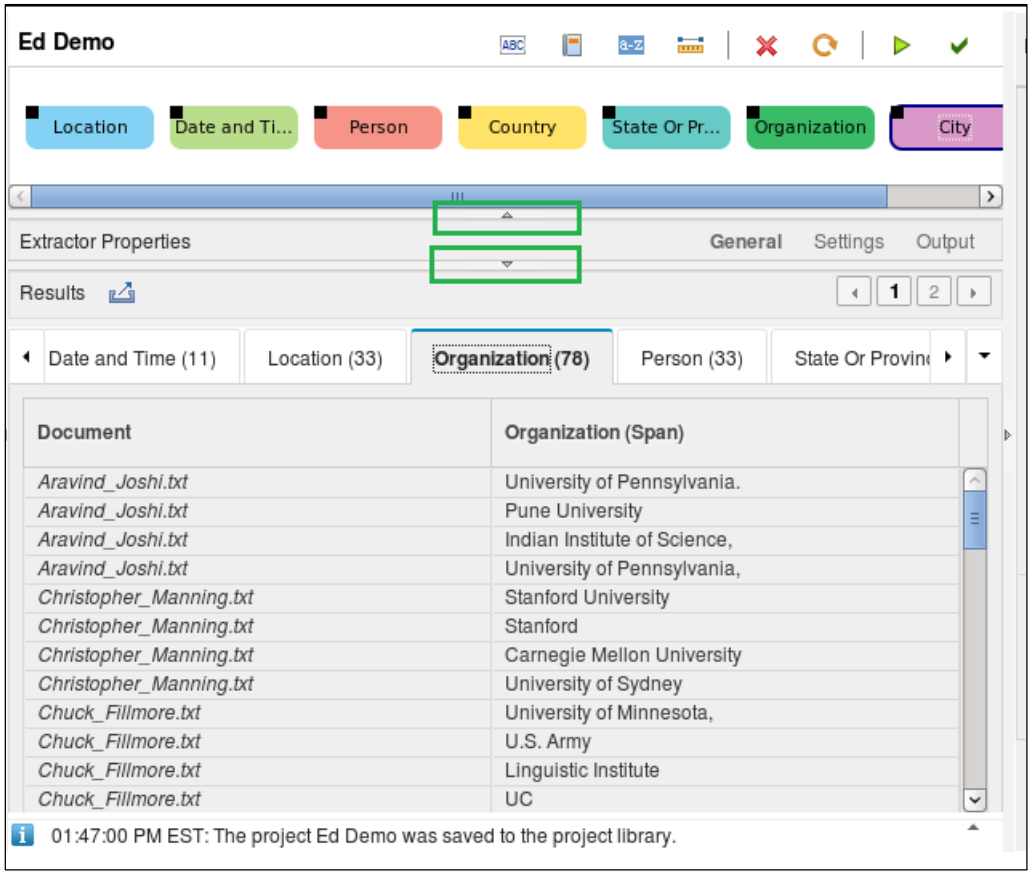


Task 5. Examining the results of the run.

1. Each of those items on the canvas are the various extractors. The one of interest to us now is the **Organization**. In my output, it is the green rectangle. Yours may be different. On the **Results** pane at the bottom of the canvas, click the **Organization** tab to bring up its results.



2. Go ahead and collapse the **Extractor Properties** pane and resize the **Results** pane to make more room. Click on the horizontal bar with the upside-down triangle to toggle the expand/collapse function. The same bar can be used to resize if you bring your mouse cursor over it and then click and drag to resize.



3. In the canvas, go ahead and delete all the other extractors. Keep the **Organization** one. Select the ones you wish to delete and press the red x. Alternatively, select and press the **Delete** key on your keyboard.

   Note that you can drag and drop extractors along the canvas.

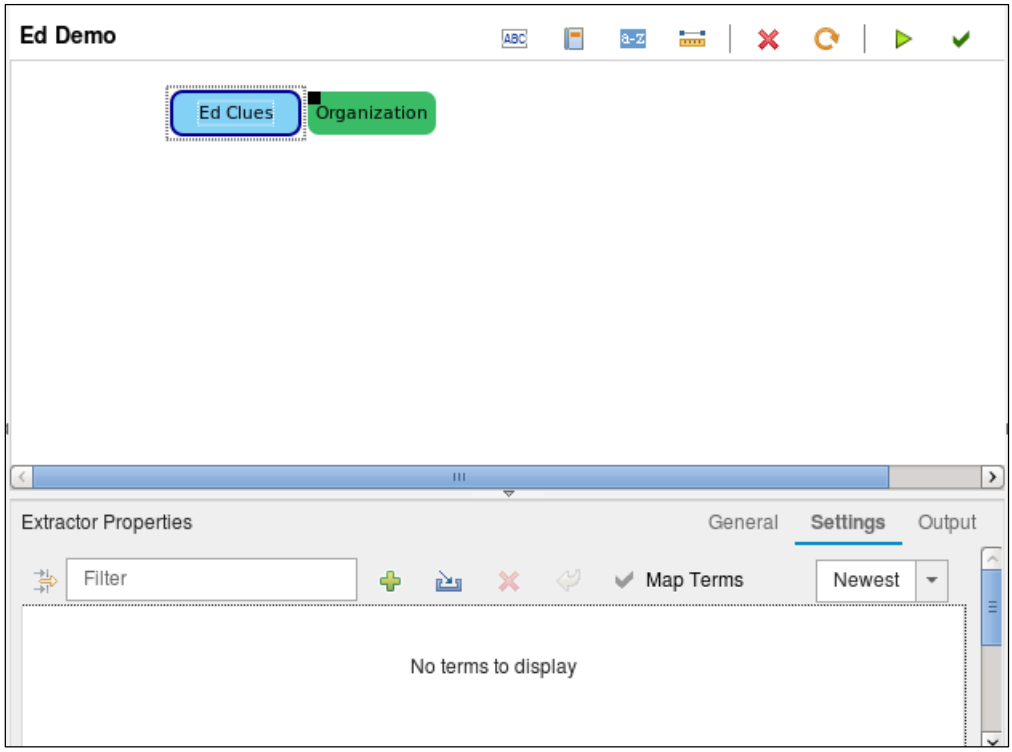
Task 6. Creating a dictionary of clues to search.

1. At the toolbar above the canvas, click the **New Dictionary** button.

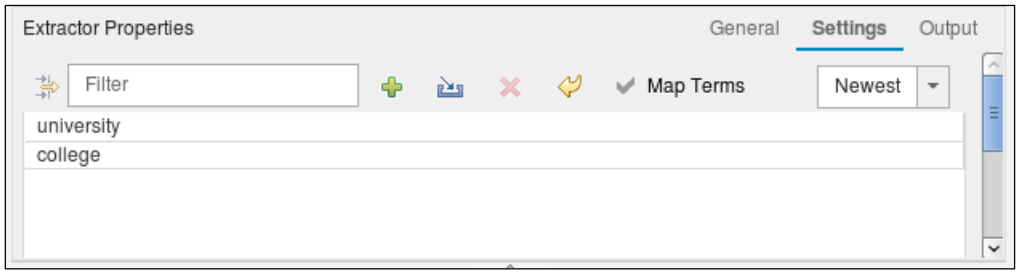
2. On the canvas, the dictionary shows up. Name the dictionary, **Ed Clues.**

3. Add the following clues into the **Extractor Properties** pane:
   - college
   - university

   Go ahead and collapse the Results pane and expand the Extractor properties pane (if you haven't done so already).

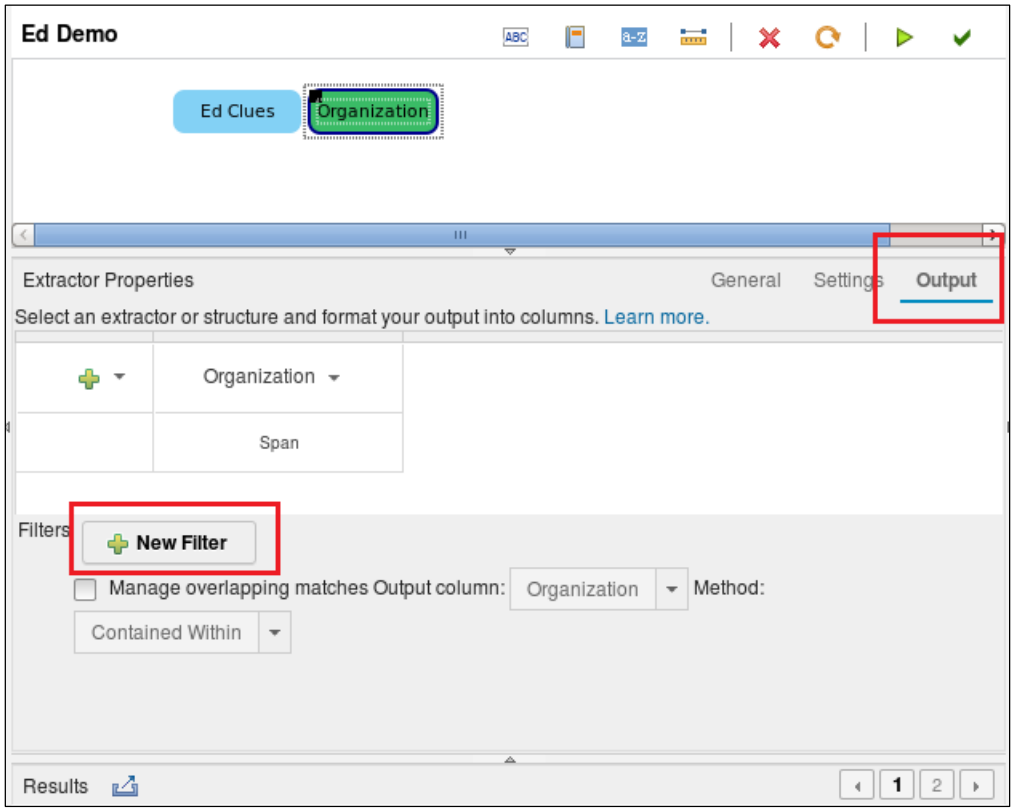


4. To add a new term. Click the **green plus** and add the clues.



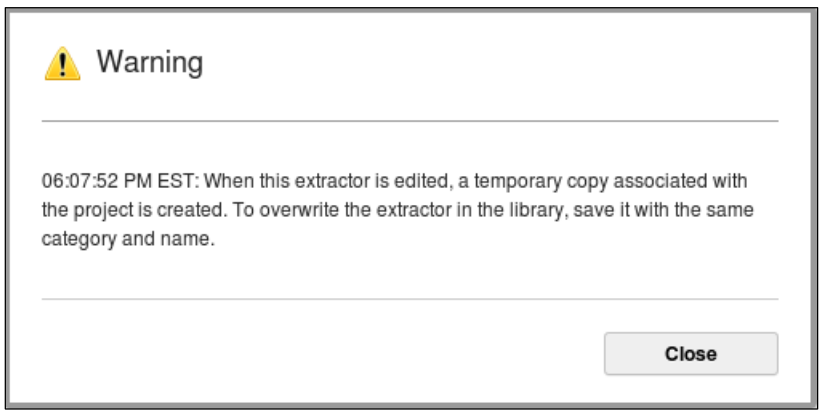
Task 7. Filtering the results and running the updated extractor.

1. On the canvas, click on the **Organization** extractor.
2. Click the **Output** tab.



Note a couple of things. To see all of the properties, I kept the **Results** pane collapsed and resized the **Extractor Properties** pane.

3. As you may have guessed, you are going to filter out rows that do not contain the clues in the dictionary. Click on the **New Filter** button.
4. Click **Close** to get out of that *Warning* dialog.

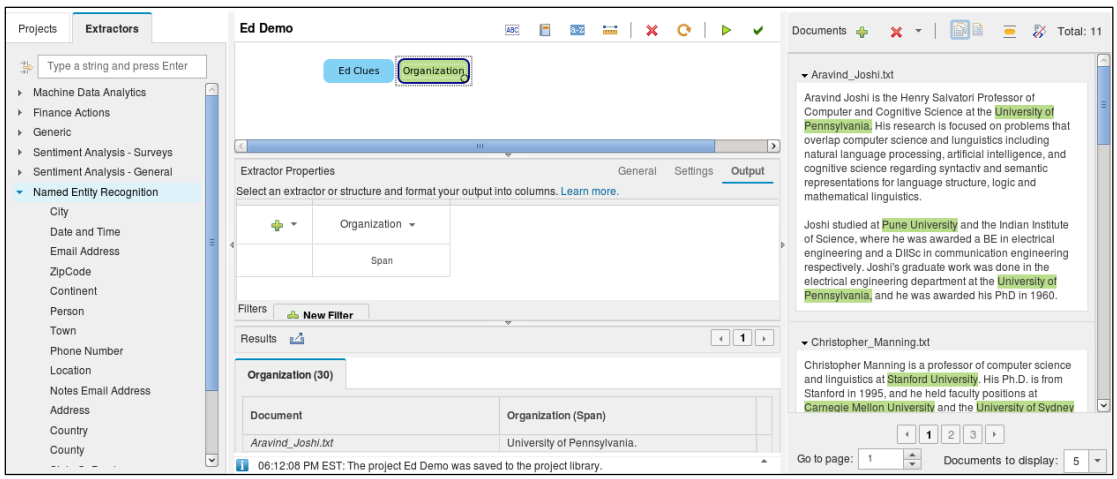


5. Edit the filter to: **Include** rows where **Organization text contains dictionary terms in Ed Clues (case sensitivity:Ignore Case**).

   For the purpose of showing the screenshot: I collapsed the panes to the left and right of it. You can do the same if you need more room to edit the filter.

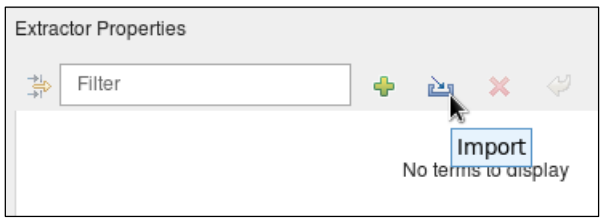
6. Before you run the extractor. Restore (expand) the Results pane to see that there are currently 78 rows where it originally matched. Run the extractor by clicking on the **green play arrow.**

7. Now note that the results went from 78 to 30 rows because you are only matching on the terms in the **Ed Clues** dictionary. The rest were filtered out.



Task 8. Creating a dictionary by importing a file.

1. Create a new dictionary by clicking the **New Dictionary** button.
2. Name the dictionary: **Degree**
3. With the **Degree** dictionary selected, click the **Import** button from the **Extractor Properties** pane:

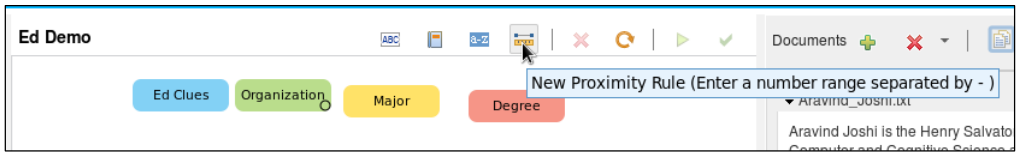


4. Select the **Degree.txt** file from under */home/biadmin/labfiles* and import it.

5. Create another dictionary called **Major** and import the **Majors.txt** file.
6. You should now have two new dictionaries created and loaded with terms: **Degree** and **Major**.

Task 9. Creating proximity rules.

1. Create a Proximity rule. Click the **New Proximity Rule** button.



2. Give the range of the proximity rule of **1 - 5** tokens.

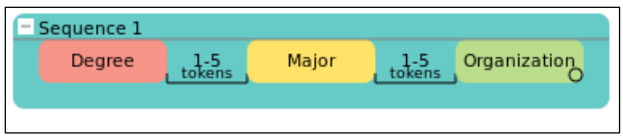
3. Create the same proximity rule again so that you have two proximity rules total.

Task 10. Creating a sequence of extractors.

1. On the canvas, arrange the extractors into a sequence using drag and drop. Arrange the extractors in this order:

   Degree, 1-5 tokens, Major, 1-5 tokens, Organization

   When you drag an extractor or a rule next to another, a blue bar will appear on indicating that it will attach to that side when you let go of the mouse button.
2. When you have done it correctly, you would have created a new sequence:

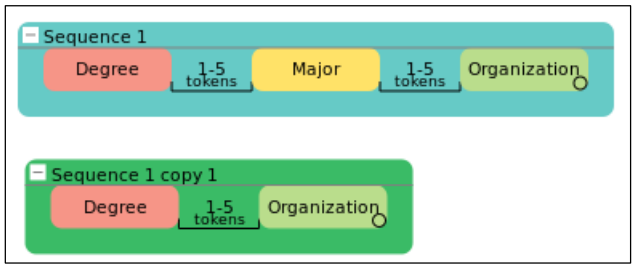


3. With that sequence selected, run the sequence by clicking the **green play arrow** on the toolbar.

4. Once again, I resized the **Results** pane so that I can view them. Do so if you need to for your environment.

   Notice in the Results pane, there are four tabs. Each of the tabs represent the results from each individual extractors, plus the fourth tab for the sequence of the three extractors with the proximity rules. Because I know the data well, and this is a made up demonstration scenario, I know that in the sequence tab, you are missing the sequence from the University of Michigan "Ph.D in 1961 from the University of Michigan"
5. Right-click on **Sequence 1.** Select **Copy.**
6. Right-click on the canvas and select **Paste as New Copy.**

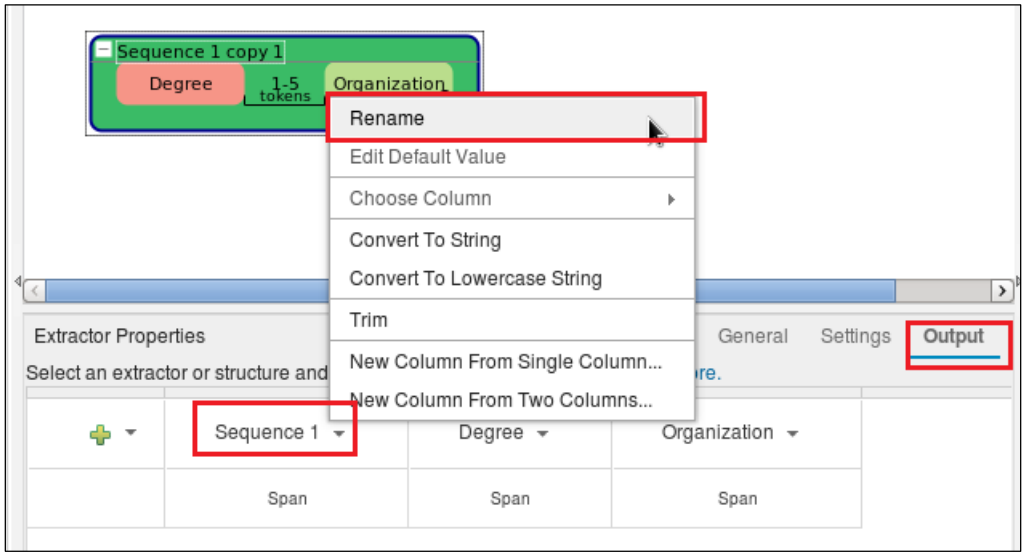
   Note: **Paste as New Copy** is essentially cloning the original extractor. If you did the normal paste. any changes made to the source will affect the copy as well.
7. In **Sequence 1 copy 1,** remove **Major** and one of the proximity rules by dragging it out of the sequence. You can delete those two items. This is what it should look like now.

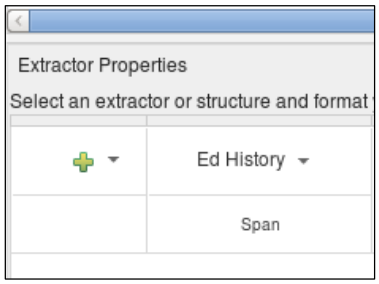


8. Go ahead and run **Sequence 1 copy 1.**
9. Note that the missing entry is now present.

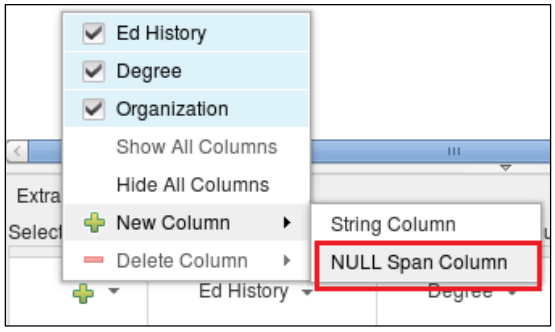
Task 11. Creating a union of extractors.

1. Under the **Extractor Properties**, on the **Output** tab, change the name of the **Sequence 1** column to **Ed History.** Select the **Sequence 1** column options and click **Rename**.



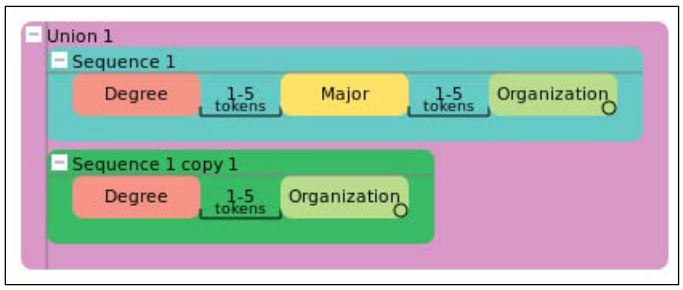


2. Add a new NULL Span Column. Click the **green plus** button → select **New Column → NULL Span Column**.

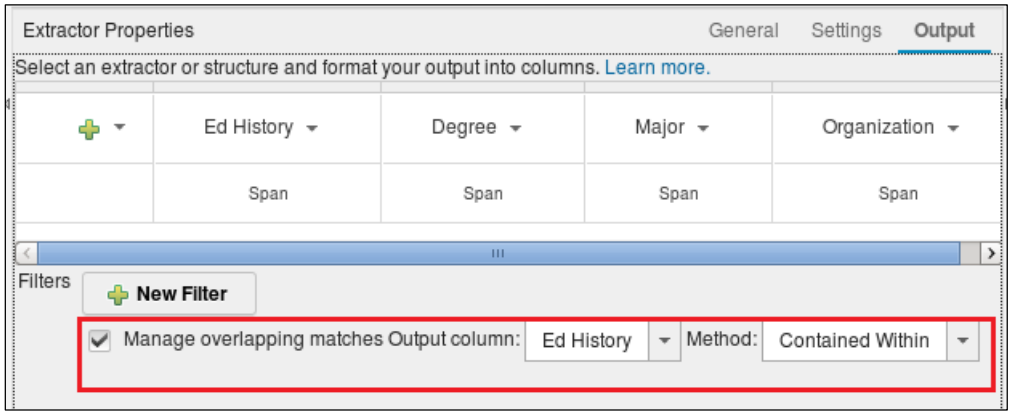


Name the new column Major.

3. At this point, you should have four columns for **Sequence 1 copy 1**.
4. Back on the canvas, click on the **Sequence 1** extractor and rename the **Sequence 1** column to Ed History (just as you had done for the Sequence 1 copy 1 extractor).
5. Now that both extractors have the same schema, drag and drop **Sequence 1** to align vertically with **Sequence 1 copy 1** to create Union 1. The blue bar should be at either the top or the bottom to indicate a union action.



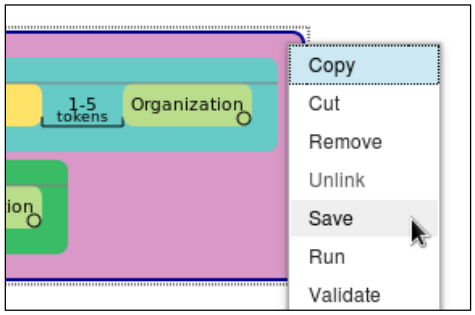
6. Rename **Union 1** to **Education History**. Double-click the text directly on the canvas to rename it.
7. Run **Education History**.
8. Examine the results. There are 11 rows. Notice that there are duplicate values.
9. Back on the **Extractor Properties**, on the **Output tab**, check the box for **Manage overlapping matches.** You may need to resize the properties pane to see it.



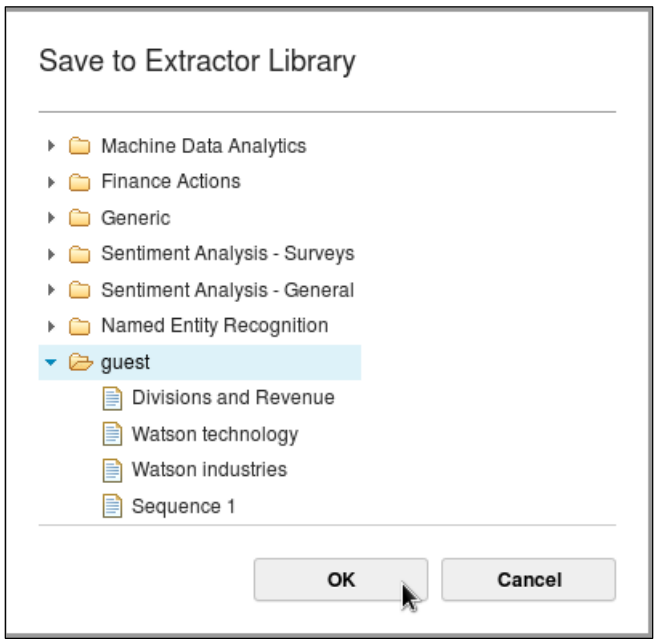
10. Run the **Education History** extractor again. Note that the number of returned rows went from 11 to 7.

Task 12. Saving extractors and exporting results - Optional.

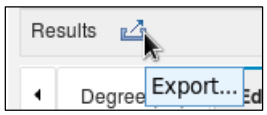
1. Right-click on the **Education History** extractor and select **Save**.



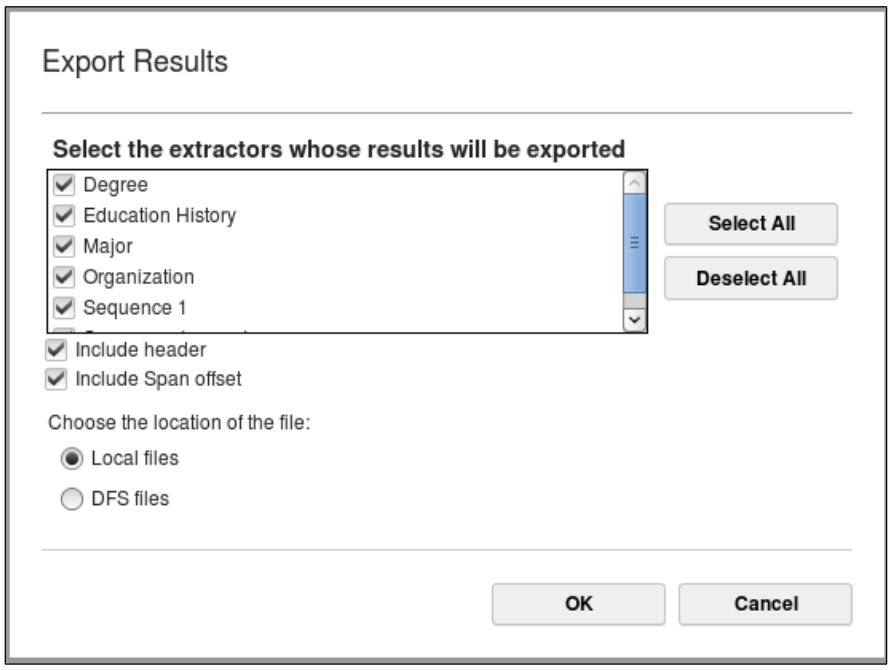
2. Select the **guest** directory and click **OK**.



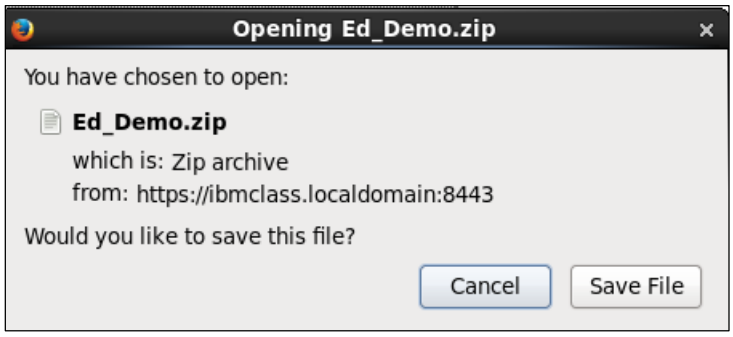
3. In the Results view, click the export button to export the results to csv format.



4. Select the results, specify your options, choose your location, and click **OK**.



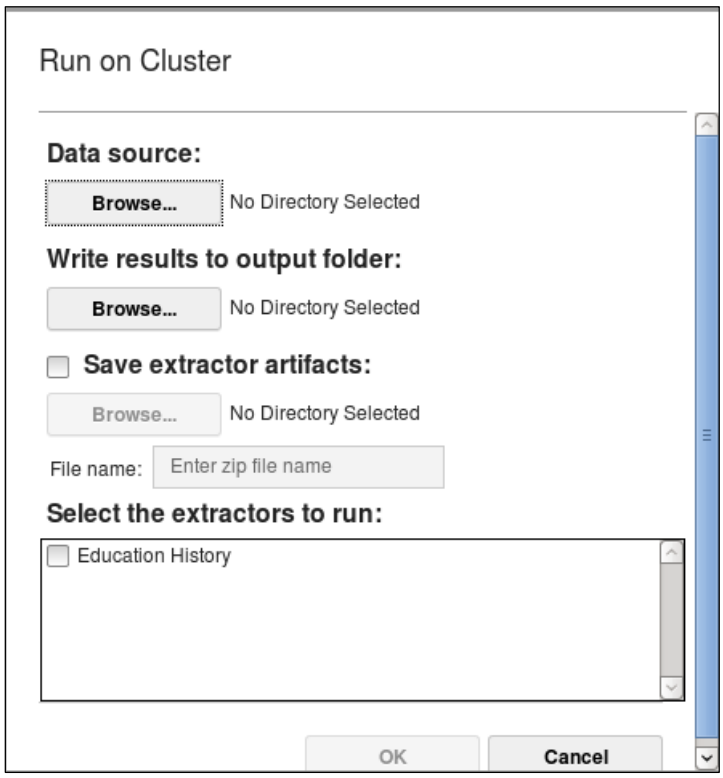
5. You will be prompted to download the file by your web browser:



Task 13. Running on a MapReduce cluster - Optional.

1. From the **Extractor** catalog, expand **guest**, right-click **Education History** and choose **Run on Cluster**

2. Specify the data source, output folder, and the extractor to run.



Task 14. Publishing to BigSheets - Optional.

1. Similar process as pushing the job onto a MapReduce cluster. Make sure that your BigSheets service has started. Since this task is optional, the BigSheets service wasn't required to be started earlier. You can explore this on your own.
2. Note on publishing NULL spans to BigSheets. There is a bug with this release where you will not be able to publish NULL spans to BigSheets. Our Education History extractor contains a NULL span column, if you recall. We added this NULL span column in order to ensure that both of the extractors in the union had the same number of columns. You can still publish the Sequence 1 extractor as a BigSheets function since that one didn't have any NULL spans within.
3. Once published to BigSheets, the extractor will be a function from which you can use to create child worksheets to extract data. From there, you can use BigSheets visualization to paint a picture of your data.

Results:
You created a Text Analytics project which analyzed text data to find the education histories. You created extractors to locate the degree, major, and the organization within the biography files. Then you consolidated and finalized those extractors to create the final Education History extractor. In subsequent demos, you will see how to use each of the individual text analytics steps in more detail.

IBM Training

Unit summary

- Overview of the BigInsights module
- Compare structured vs unstructured data
- Understand how to design your project
- Describe and list the steps used for text analytics

Unit summary

IBM Training

Task Analysis

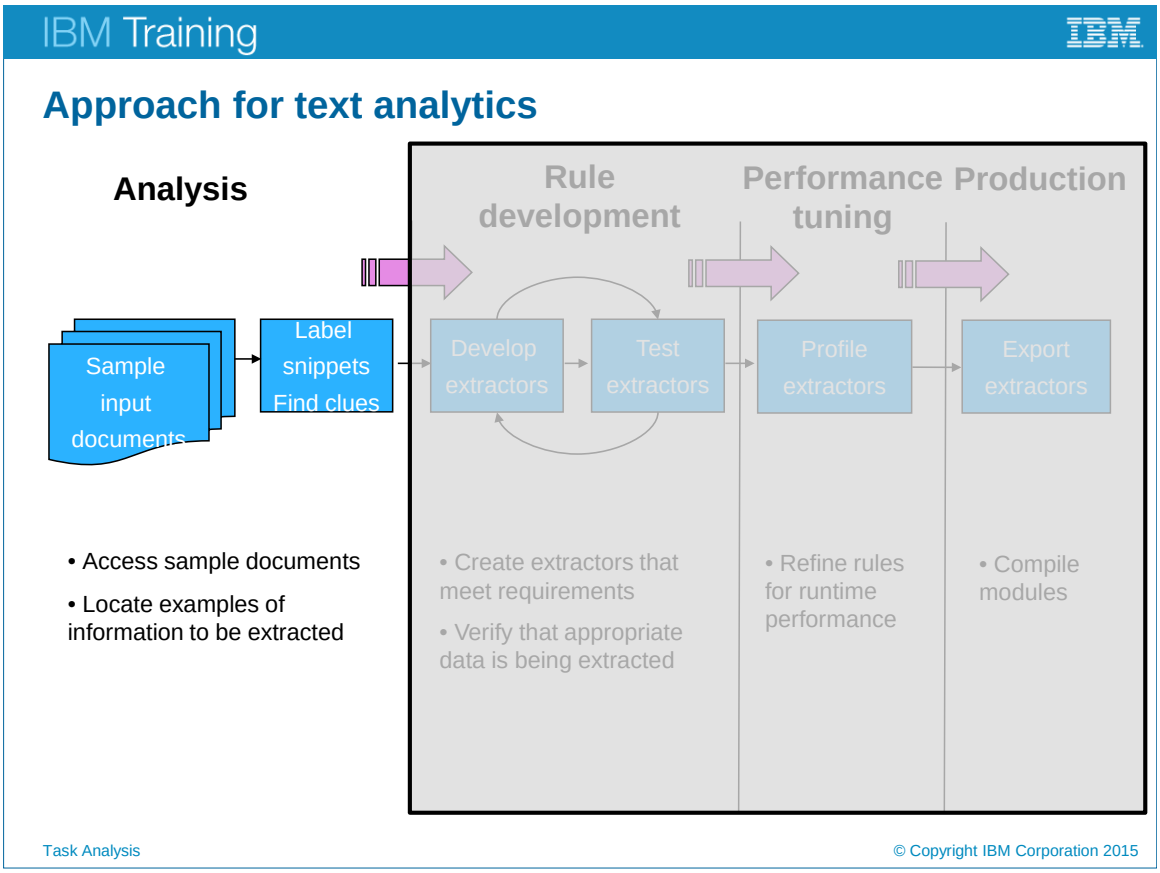
IBM BigInsights v4.1

IBM Training

Unit objectives

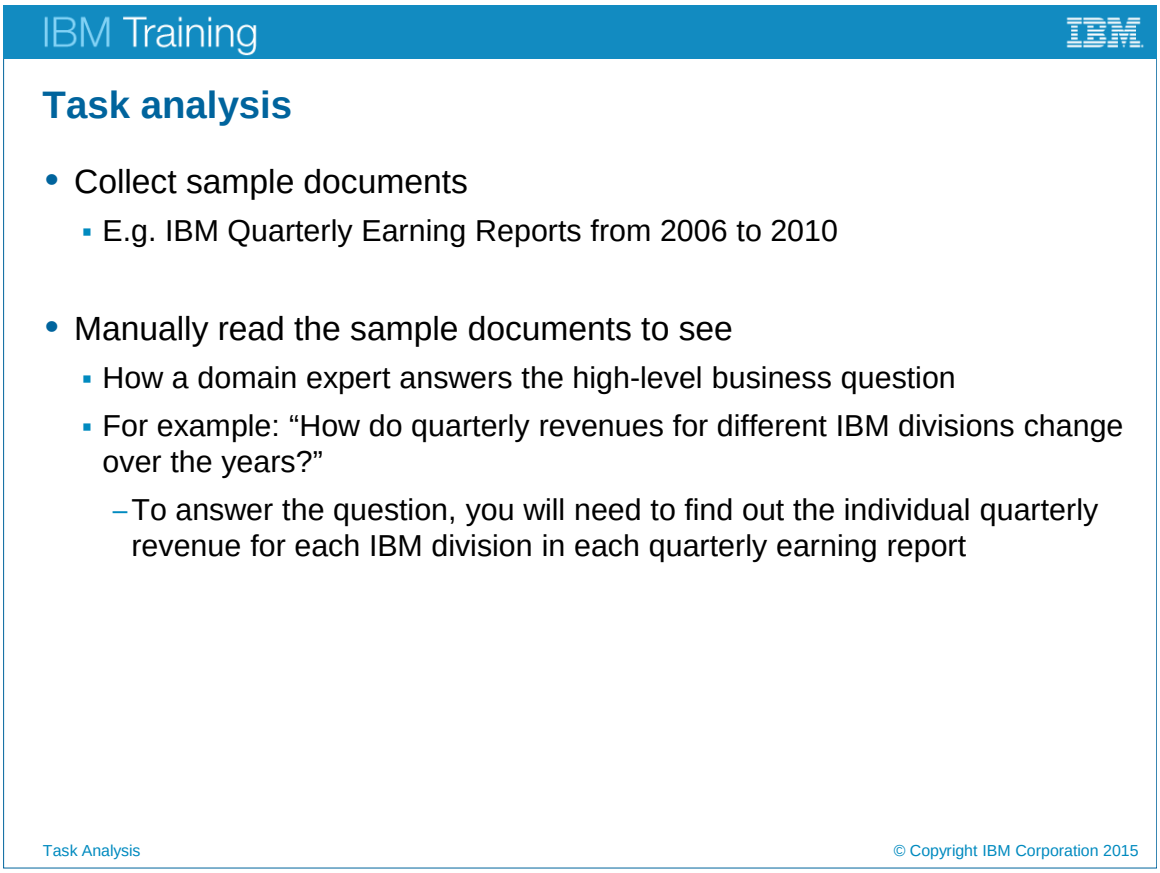
- Describe the approach of the task analysis phase
- List the task analysis steps
- Label clues in your documents that will help you to create your extractor

Unit objectives



Approach for text analytics

In this unit, you focus on the task analysis phase to take a look at a few sample documents. From those documents, you identify snippets and clues of the information to be extracted. In this phase, you would generally enlist the help of subject matter experts that are well familiar with the documents to find better clues to aid in the text extraction.



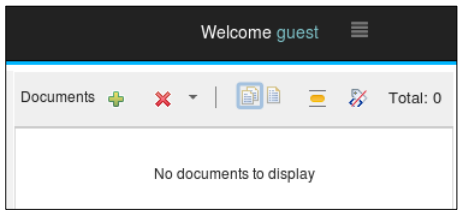
Task analysis

There are two steps to this phase. The first is to collect the sample documents of the entire dataset you wish to analyze. From these documents, you will manually examine them to find clues that will help find what you need.

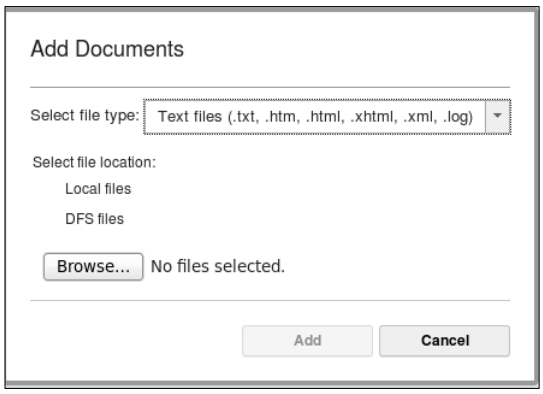
IBM Training

Select a data collection

- Locate the Document viewer pane:



- Click the Add Documents button



Task Analysis

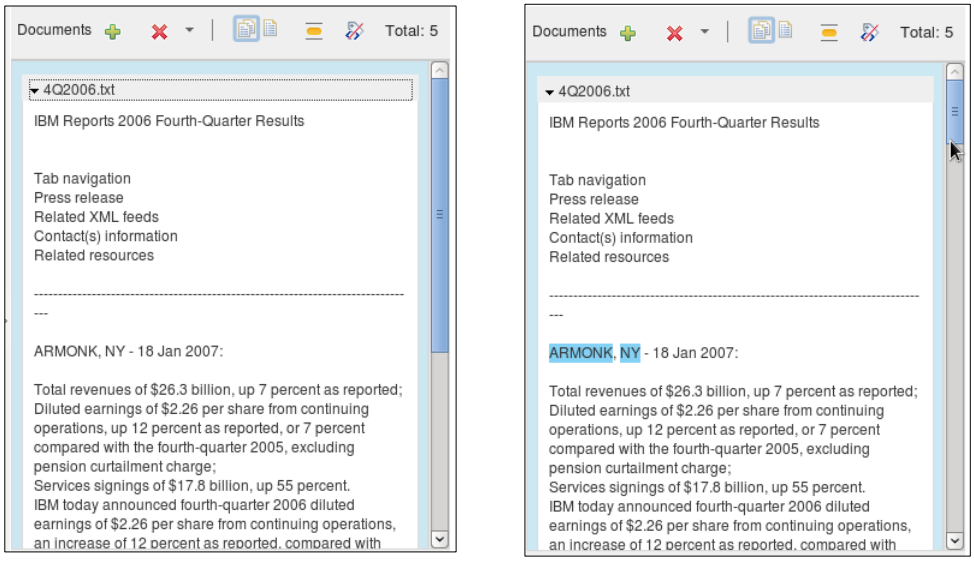
Select a data collection

First step is to select a data collection that you will be analyzing. Locate the document viewer pane (on the right side of the UI). From there, click on the **Add Documents** button, the green plus icon and specify the file type and the file location, either on the local filesystem or the DFS. Then click browse and select one or more files and select **Add**.

IBM Training

Load the data collection

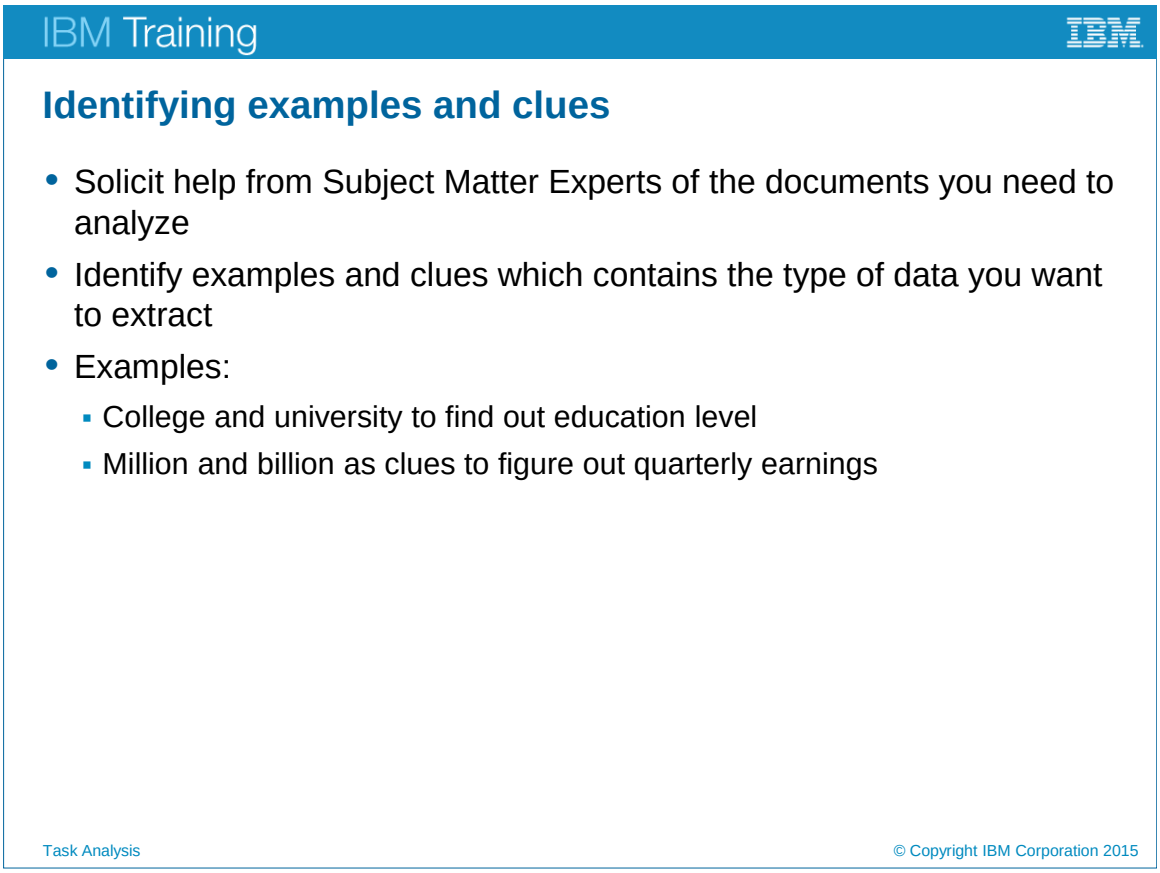
- Document Viewer



Task Analysis

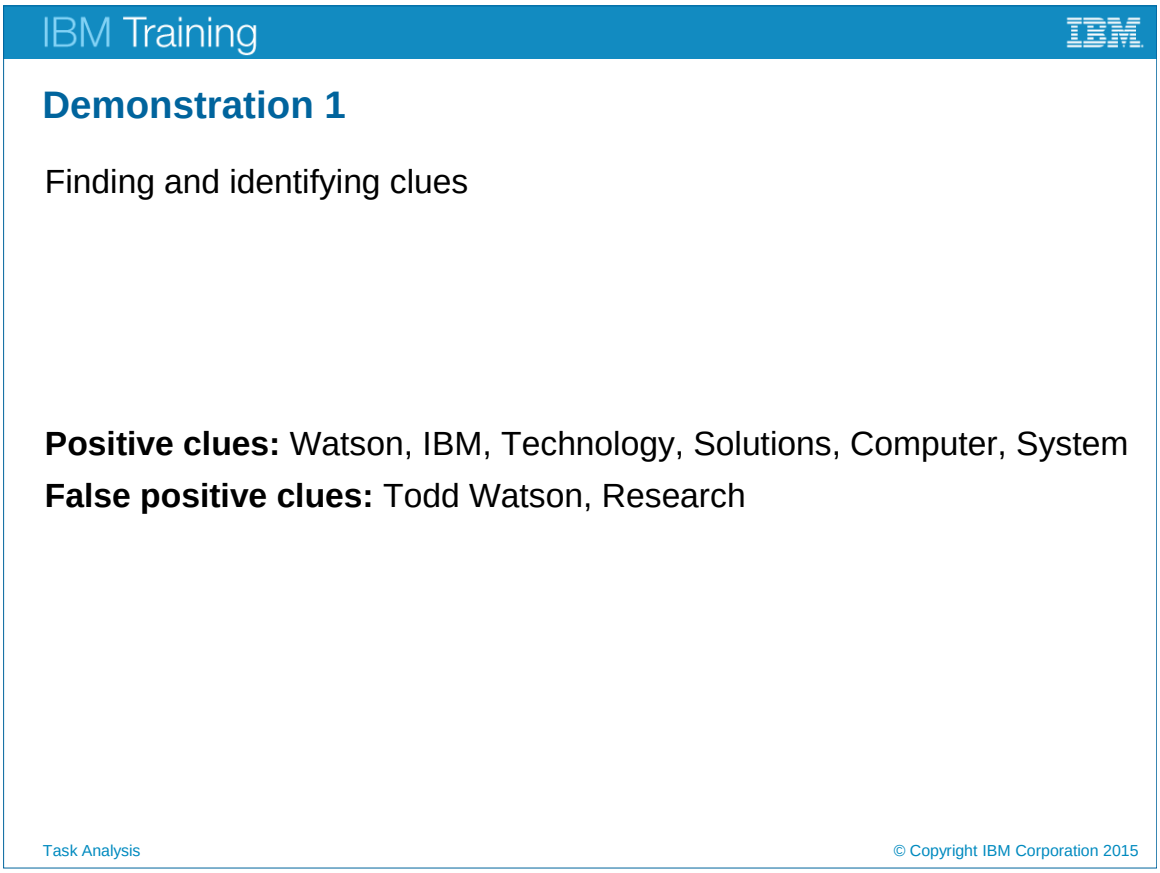
Load the data collection

Once the documents are loaded, you see them in the Document Viewer. When you run and test your extractors, the terms that matches what you are searching will be highlighted in the Document Viewer.



Identifying examples and clues

In this step, you identify examples of clues that contains the type of data that you ultimately want to extract. For example, you might select the text *million* and *billion* as clues to help you figure out the quarterly earnings. In the previous demonstration, the dictionary you started with had the clues for *college* and *university* to help find out the education level.



Demonstration 1: Finding and identifying clues

Demonstration 1: Finding and identifying clues

Purpose:
This demonstration will show you how to find and identify clues that are needed for the extractor. In real life, this process would typically be done with assistance from a subject matter expert, or someone who is familiar with the documents that you are examining. Prior to starting this demonstration, ensure that all the necessary Ambari services are up. If you had just completed Demonstration 1, you are in good shape. Otherwise, refer to demonstration 1 to get that set up.

User ids / Passwords

OS:	**biadmin/biadmin**
Root:	**root/dalvm3**
Ambari:	**admin/admin**
BigInsights Home:	**guest/guest-password**

Ambari Services Required:

- HDFS
- MapReduce2
- YARN
- Knox (also start the Demo LDAP service)
- BigInsights - Text Analytics
- BigInsights - Home

Task 1. Finding your way around the Web UI.

1. With the required services started, open up a new browser (or a new tab).
2. Go to the BigInsights - Home page. Use the bookmark saved in the Firefox browser, or this URL:

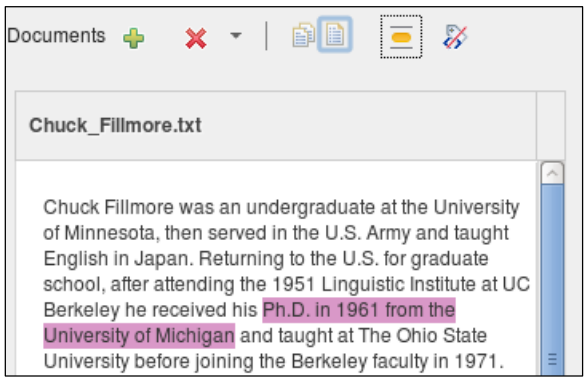
   https://ibmclass.localdomain:8443/gateway/default/BigInsightsWeb/index.html#/welcome
3. Click on Text Analytics to load up the Web UI.
4. You have used this in the first demo, but let's spend a little more time on the Web UI to make sure you know your way around. If you feel comfortable enough, you may skip this task. The left side of the UI has your **Projects** and **Extractors**. Click on the **Ed Demo** project to load it (if it wasn't already loaded).

   This loads all your extractors onto the canvas. It also loads the documents that were used in that project.

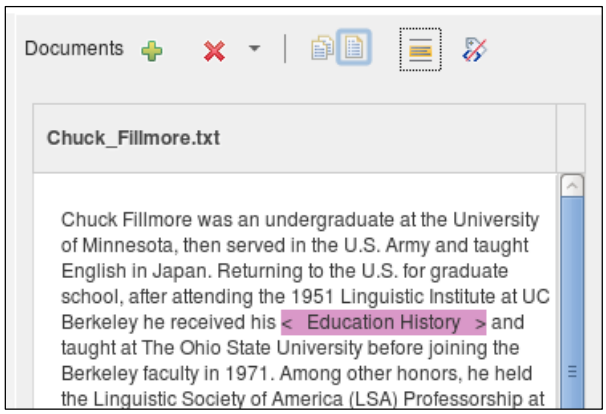
5. Click on the **Extractor** tab to see the list of the pre-built and custom-built extractors. You can drag and drop these directly onto the canvas to start using them.
6. On the canvas, select the **Degree** extractor.
7. Expand the **Extractor Properties** pane to see its settings. You may need to resize by click and dragging the pane. Play around with this to get comfortable in resizing the panes.

   Note: You can only resize if the panel is expanded.
8. Under the Extractor Properties, there are three sub-tabs: **General**, **Settings**, and **Output.** Click the **General** tab (if it isn't already on it).
9. On the **General** tab, you can edit the name, provide a description, or define some tags to assist in being more easily searchable among the Extractor catalogs. We will not do anything here, this is just for your information.
10. Click on the **Settings** tab. On here, you can modify the terms in the dictionary (in this case) or if it was a different extractor, modify the settings of that one.
11. Click on the **Output** tab. Here is where you can specify the columns from the extractor.
12. On the canvas, click on the **Education History** extractor and run it.
13. Go ahead and collapse the **Extractor Properties** and expand and resize the **Results** pane so that it is more visible.
14. Each tab on the results pane comes from a single extractor. In our case, we have a single union of multiple extractors, so we have single tab. Within that one tab, however, we have multiple results, one for each of the extractors that made up that union. Examine the results to see the various columns.
15. Click on any row and you will see that the results are highlighted within the document on the **Documents** pane (on the right).
16. Remember, you have the option to export your results as a CSV file for further analysis with a different tool.
17. On the **Documents** pane, you can toggle between single document view and multiple document views. Go ahead and click on it to see it in action.

18. Next to that is another button, **Show Extractor Name.** This is a nice little feature that tells you which extractor found the results. For example, select one of the rows from the Results pane.

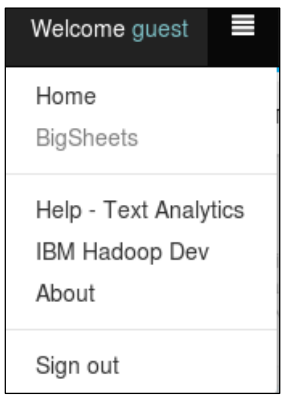


19. Now click the **Show Extractor Name** button to see which extractor it was:



Obviously, in this case, we only had one extractor, but if you ran with multiple extractors, you can use this to find out which one captured that result. This can help with debugging if you end up finding terms that should or shouldn't be part of the result set.

20. Finally, the third button is the **Remove tag / Remap tag.** This is used for documents where you may have tags, such as XML documents.
21. If you need additional help, at the upper right corner, there is a dropdown icon. Click on that and you can visit the help section for Text Analytics.

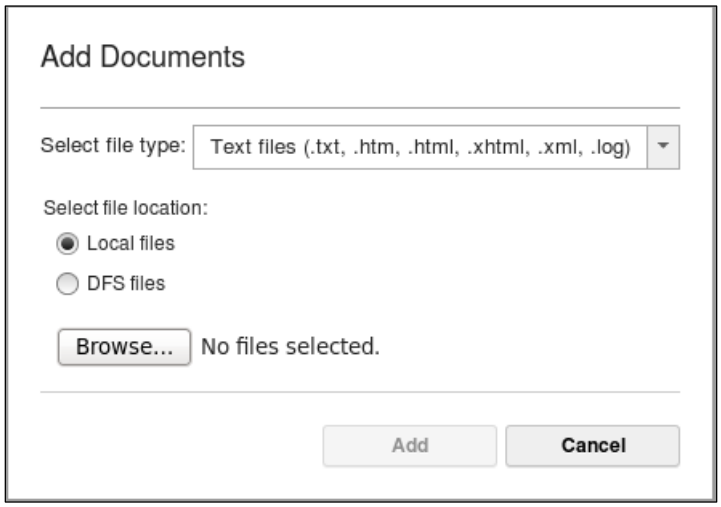


Task 2. Creating the Watson project.

1. On the **Project** pane, click the **green plus**.
2. Specify the name Watson for the project.

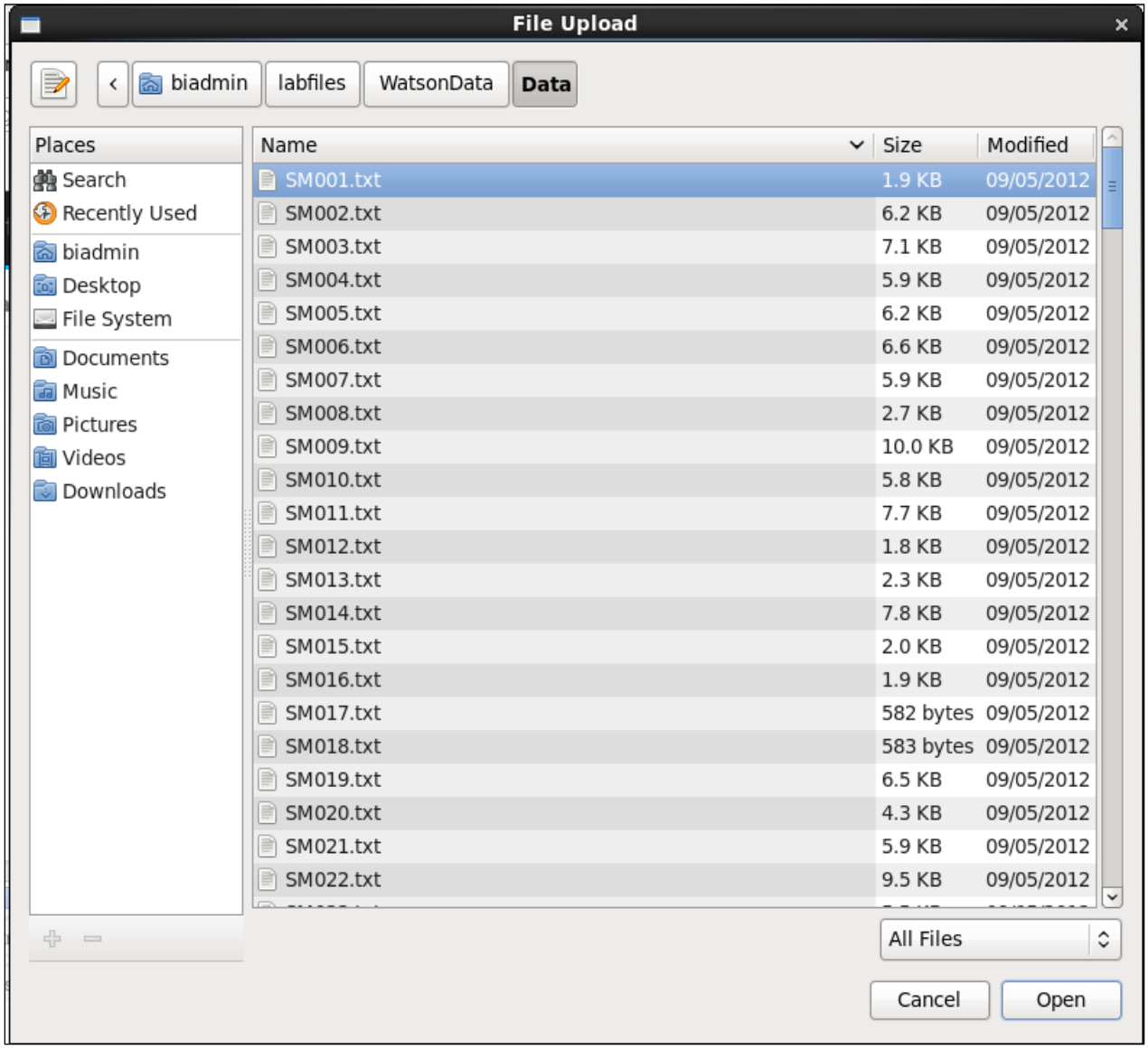
Task 3. Loading the data files.

1. On the **Documents** pane, click the **green plus**.



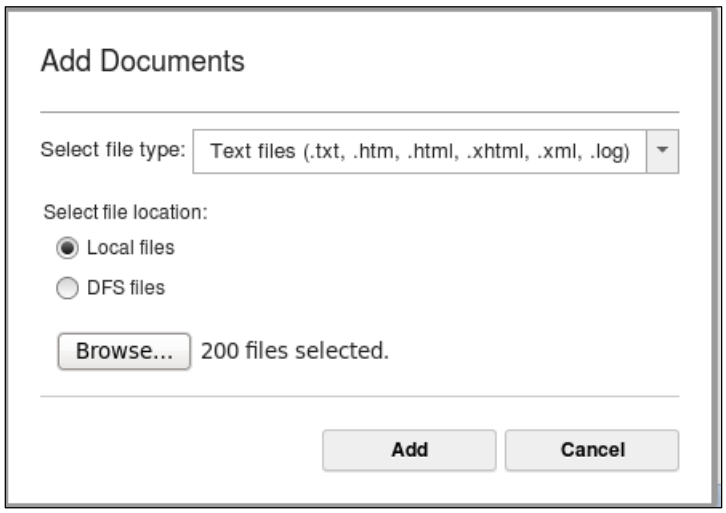
2. Specify the file type as **Text files** and the file location as **Local files**. Click **Browse** to select the files.

3. Navigate to **/home/biadmin/labfiles/WatsonData/Data/**.

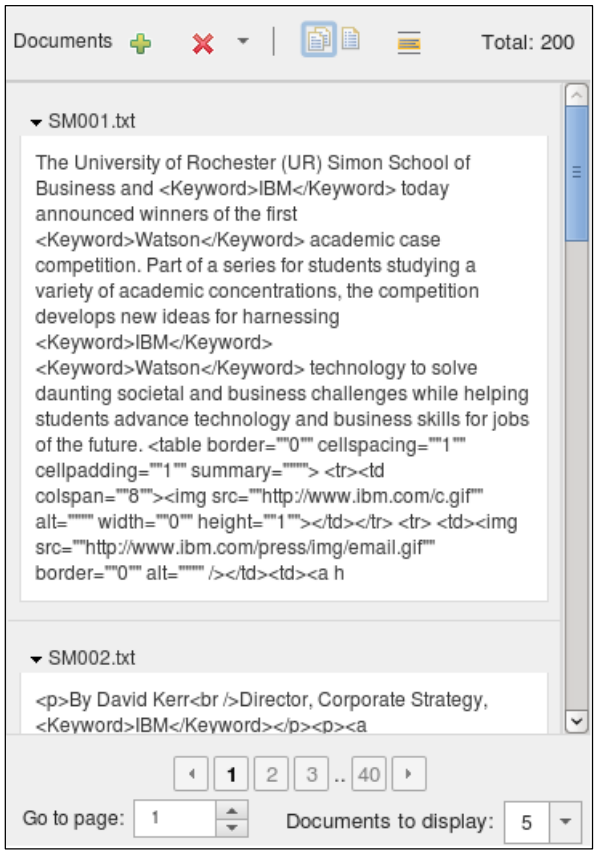


4. Select all the files. Use **CTRL + A** to select all the files and click **Open**.

5. Click **Add** to add the files.



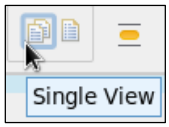
6. The documents are loaded.



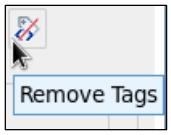
Task 4. Identifying and creating a list of the clues.

In this task, you will be creating a list of clues that you will use to create your extractor. Your test data consists of a number of files that are actually a collection of blogs and news posts retrieved from various social media sites using the BigInsights sample Boardreader application. Each post is stored in an XML encoded format. You use this test data to find examples of the type of information that you want to extract and build your extractors based on those examples.

1. Locate the file **SM001.txt**. Select that file and choose the **Single View** to show only one document at a time.



2. Next, make it easier to read by removing the tags by clicking on the **Remove tag** icon.



3. This is a copy of the text:

   The University of Rochester (UR) Simon School of Business and IBM today announced winners of the first Watson academic case competition. Part of a series for students studying a variety of academic concentrations, the competition develops new ideas for harnessing IBM Watson technology to solve daunting societal and business challenges while helping students advance technology and business skills for jobs of the future.

4. Since the goal of the task is to find social data that references IBM Watson, the first snippet of interest would naturally be the word Watson. Make a note of this word in a Notepad or a text editor of your choice. We'll keep a running note here:

   Positive clues: Watson

5. It is easy for you, as a human being, to scan through these files and find those that are referencing the Watson technology as opposed to someone's name or a place. But that same innate capability does not exist for a computer. You are going to have to give the computer both positive and negative clues for it to be able to recognize the appropriate Watson reference.

The first reference to Watson in the text was related to a competition. The second reference was IBM Watson technology. This is a reference in which we have an interest. And there are two clues that are of value, IBM and technology. It is the word Watson in context with these clue words that allow us to make the assumption as to the meaning of the word, Watson, used here.

Positive clues: Watson, IBM, Technology

6. Locate the SM010.txt file.
7. Examine the file and take note of the words *Solutions* and *computer.* These clues also relates to the Watson technology and will help the computer figure out if the Watson within the document is the Watson we want.

   Positive clues: Watson, IBM, Technology, Solutions, Computer
8. Locate the SM005.txt file.
9. Examine this file and take note of the word System.

   Positive clues: Watson, IBM, Technology, Solutions, Computer, System
10. Locate the SM011.txt file.
11. Examine the document and take note of the word Jeopardy

    Positive clues: Watson, IBM, Technology, Solutions, Computer, System, Jeopardy
12. Locate the SM063.txt.
13. Here we will look for some negative clues, or clues that may give false positives (e.g. returning Watson where it does not have anything to do with technology, but rather, a person's name or something of that nature).

    False positive clues: Todd Watson
14. Locate the SM121.txt. It's on page 25 if you are searching by page number.
15. In this file you have Watson Research Center in Yorktown Heights. Research would be another good false positive:

    False positive clues: Todd Watson, Research
16. At this point, we have enough information to work with to demonstrate the capability of BigInsights Text Analytics.

Results:
At the end of this demo, you should be able to identify clues that are needed for the extractors. You understand that typically, this process would involve someone who is familiar with the documents, such as a subject matter expert.

IBM Training

Unit summary

- Describe the approach of the task analysis phase
- List the task analysis steps
- Label clues in your documents that will help you to create your extractor

Unit summary

IBM Training

Annotation Query Language (AQL)

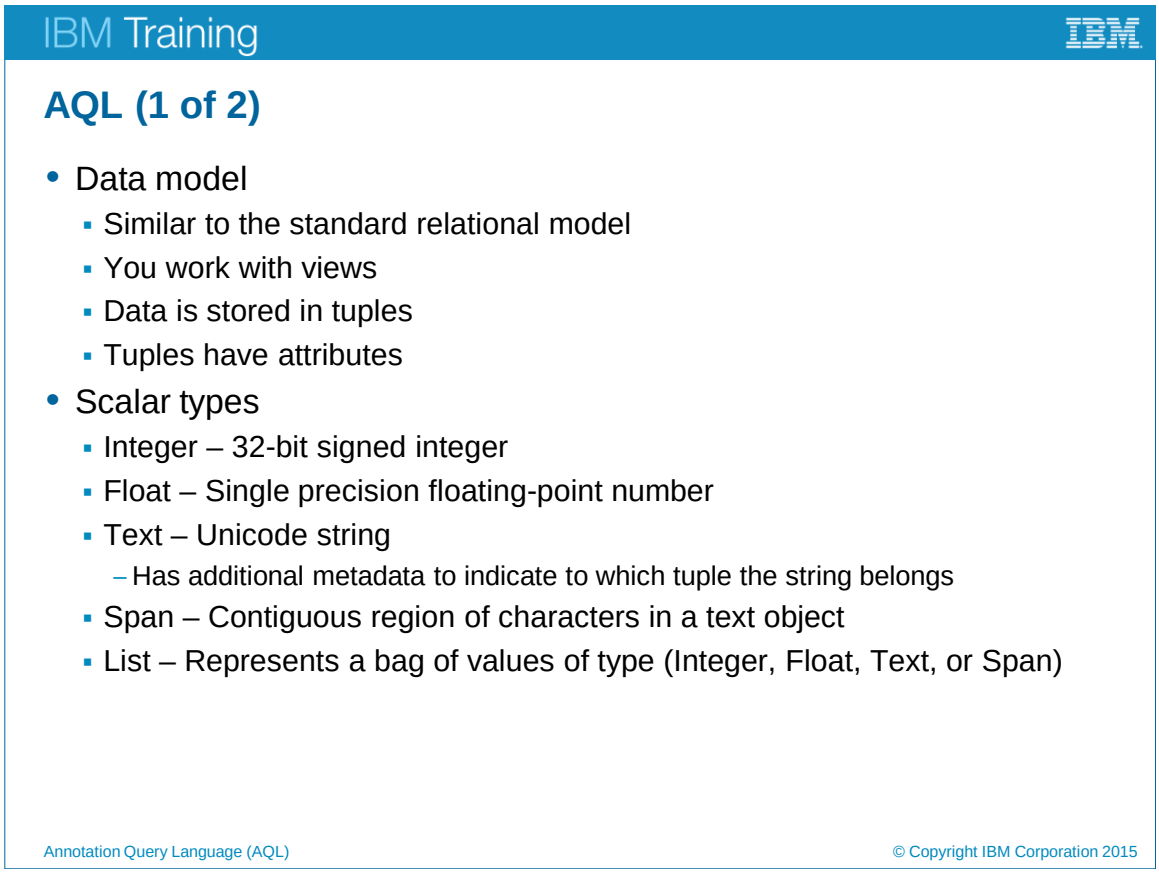
IBM BigInsights v4.1

IBM Training

Unit objectives

- Describe the AQL data model
- List the AQL components
- List the AQL objects that are used to create basic features
- Describe the Information Extraction Web Tool
- Describe the categories of the pre-built extractors

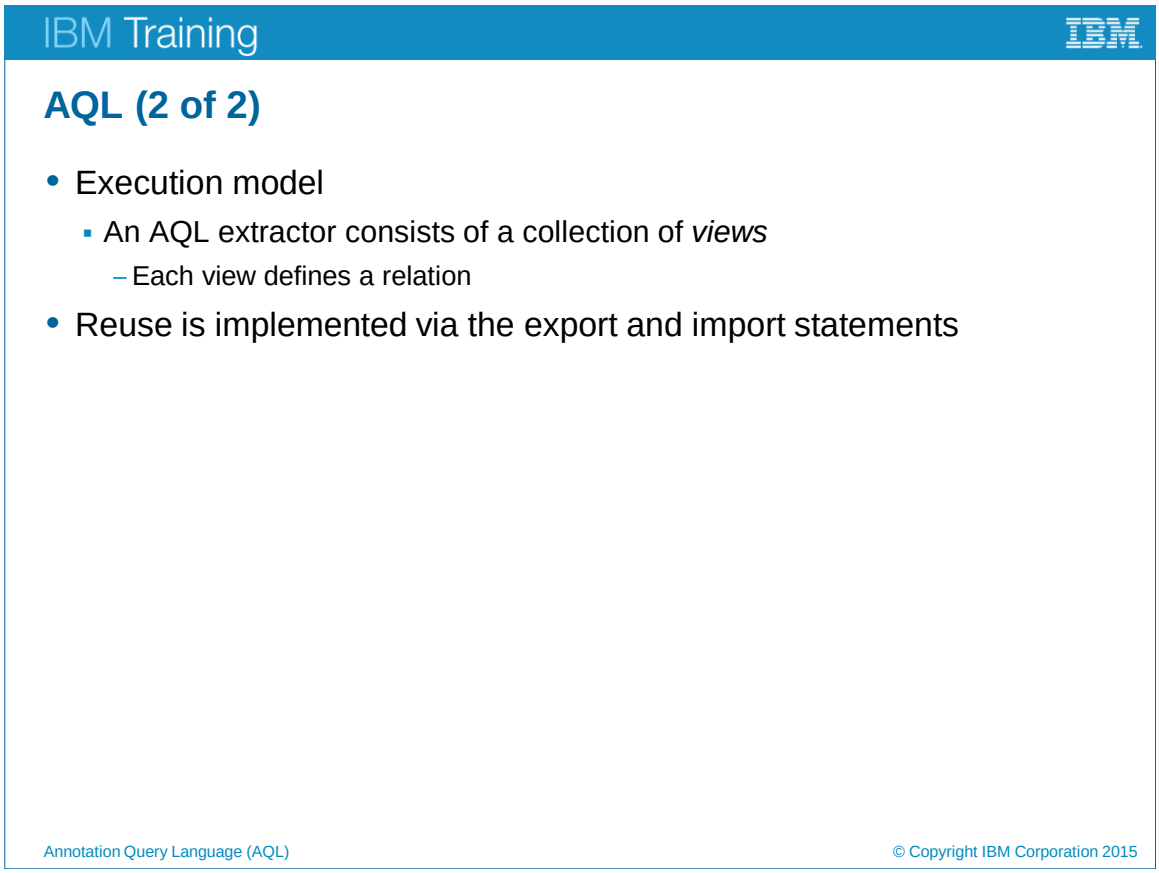
Unit objectives



AQL (1 of 2)

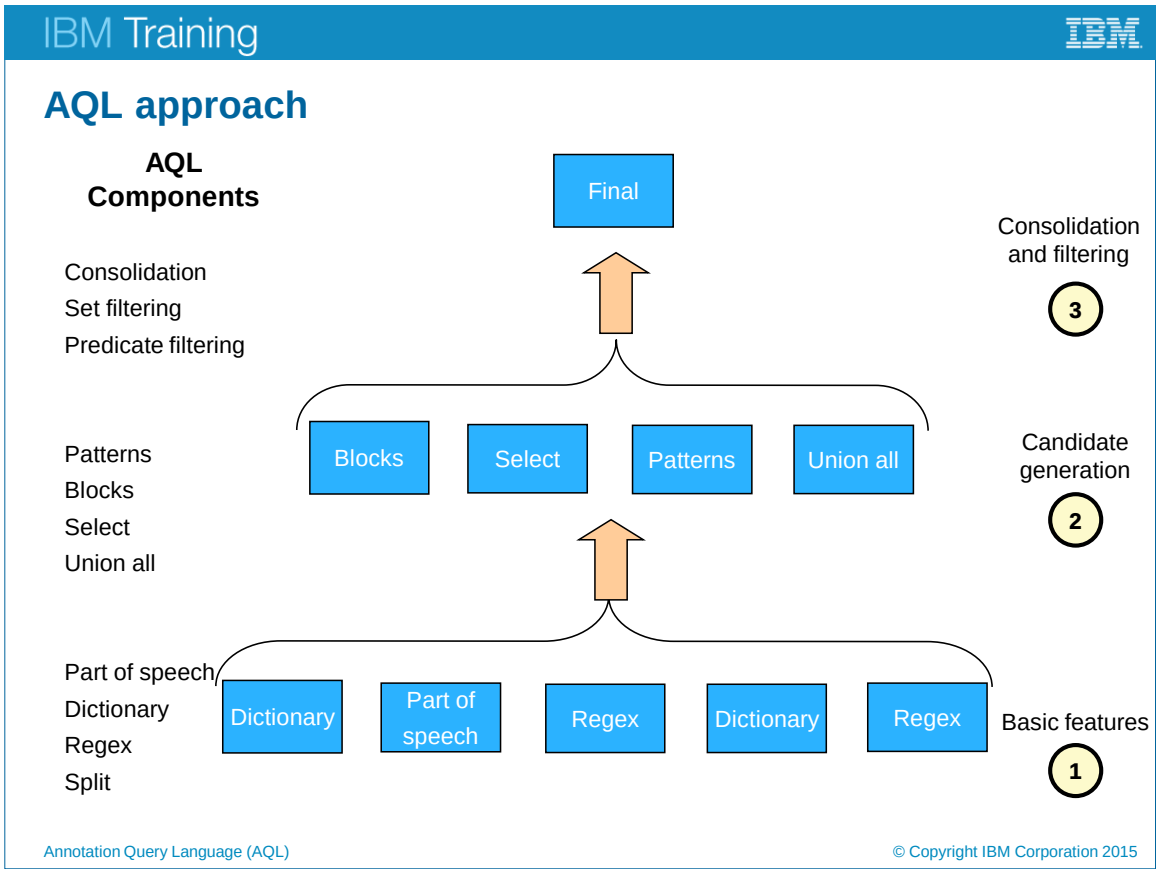
The unit will cover a bit about how the underlying AQL operates.

The data model for AQL is similar to the standard relational model used by SQL databases. To extract data, you create *views*. These are very similar to a table in a relational system. A view forms a *relation.* All data in AQL is stored in *tuples* or what you might think of as rows. Each tuple is make up of attributes, essentially columns in relational tables. All tuples in a relation must have the same *schema*, meaning the name and type of each attribute must be the same for all tuples in that relation.



AQL (2 of 2)

An AQL extractor consists of a collection of *views*, each of which defines a relation. The text analytics tooling within BigInsights makes it easy for you by keeping all of this under the covers. In fact, you will be working mainly with extractors through the web UI.

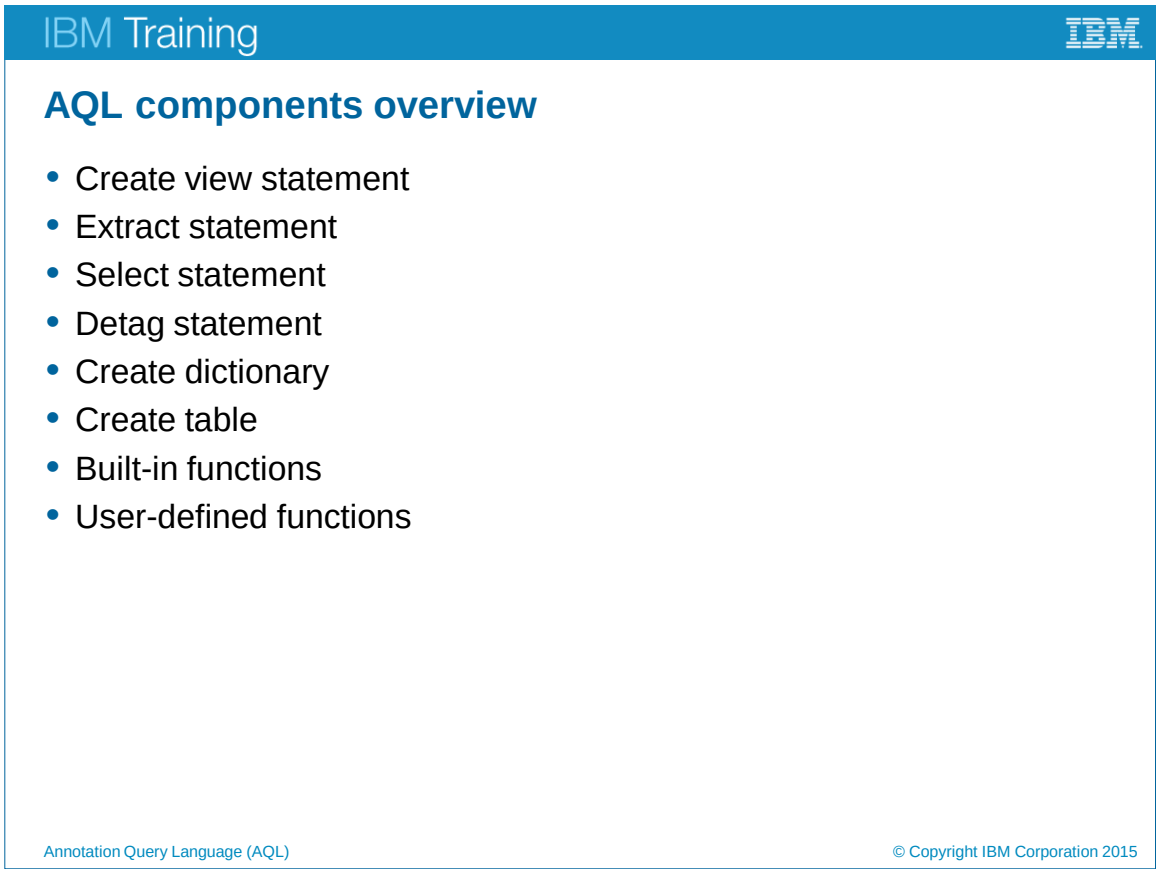


AQL approach

Creating AQL extractors is a multi-step, multi-layered process. You first start by creating fundamental components that are very specific in nature using the **basic features** of the language. These are along the lines of finding numeric strings in the data or locating all of the division names in the document,

These basic features are then used for **candidate generation**. At this level in the process you might be using multiple basic features in order to find occurrences of amounts, such as *$1.4 billion*. And then using that data to find the amounts that are associated with particular divisions. One of the things that you might find when generating candidates is that you extract more data than you require.

The third step is to then **consolidate or filter** the candidate results so that you only extract the desired data. The next couple units covers each of these steps starting with the basic features step in this unit.



AQL components overview

This is an overview of the AQL components for reference if you wish to create custom AQL code.

The **create view** statement defines a view and defines the tuples inside the view.

The **extract** statement provides functionality for extracting basic features from text. It makes use of regular expressions, dictionaries, splits, block, parts of speech, and sequence patterns.

The **select** statement allows you to create complex patterns out of simpler building blocks.

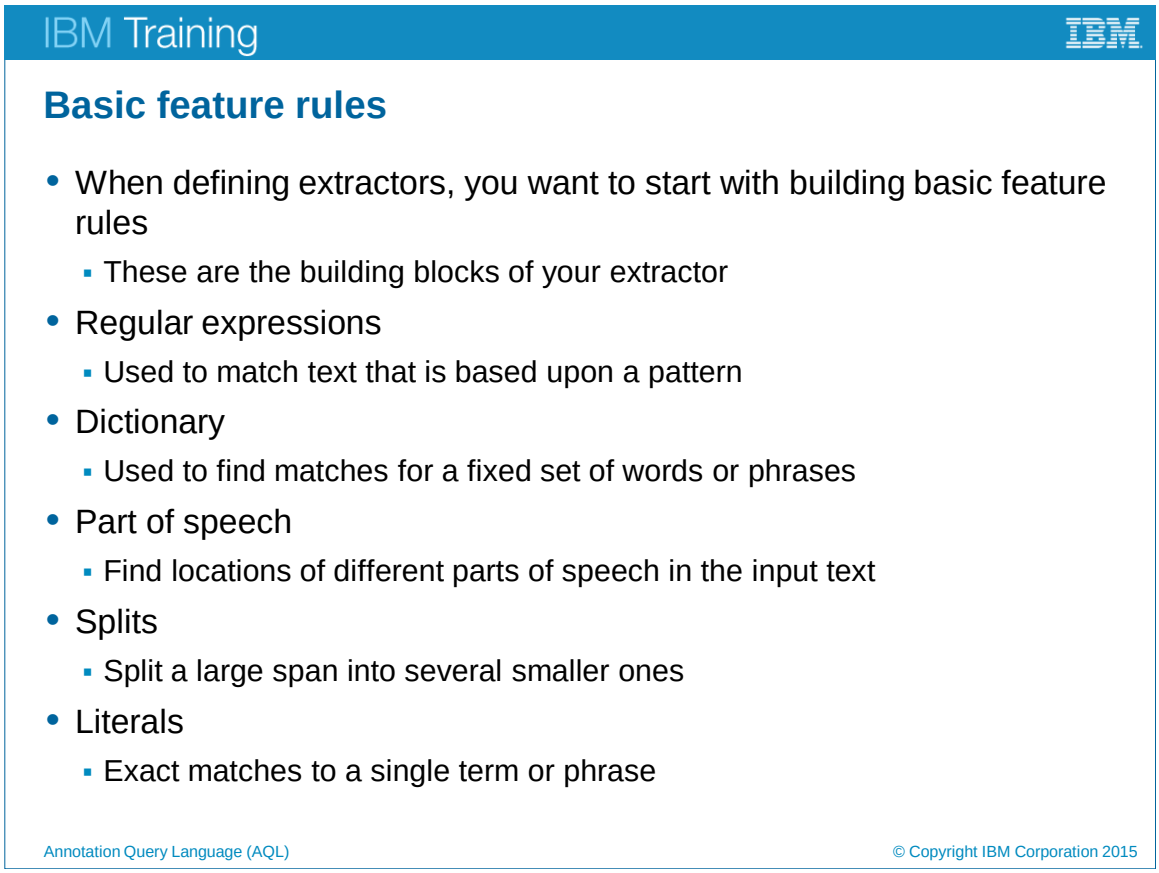
The **detag statement** removes HTML or XML tags from a document to simplify the analyzing of those types of documents.

The **create dictionary** statement allows you to define a dictionary of words and phrases that can then be used in extract statements.

The **create table** statement is similar to the same type of statement in SQL.

The **create external view** statement allows data to be passed from an external source into AQL at runtime.

There are a number of built-in functions provided with AQL that include predicate functions, scalar functions, and aggregate functions. Then there is the ability for one to code one's own function to be used in extraction rules.



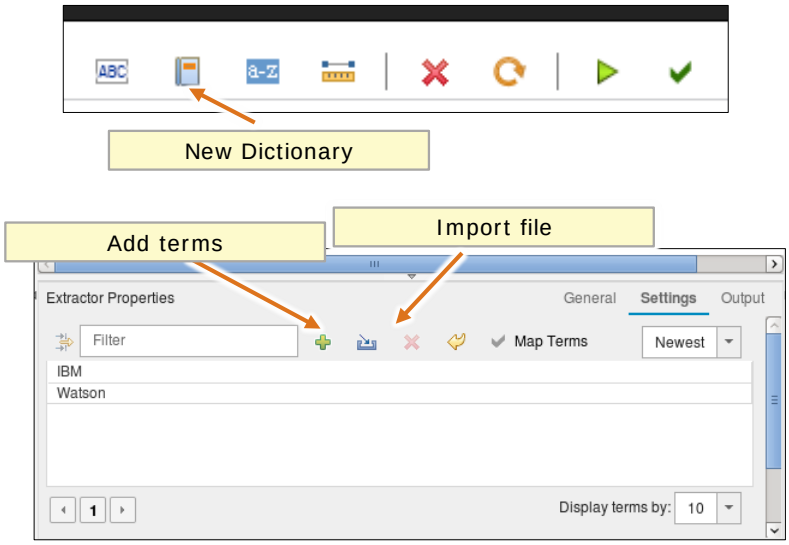
Basic feature rules

There is a best practices process that you should use when defining AQL extractors. You have already seen the beginnings of this process. That is selecting snippets of text and then locating clues within those snippets. Next you use basic feature rules to create the foundation of your extractor. These basic feature rules allow you to find numbers, unit, metrics, etc. from the document.

IBM Training

Creating dictionaries

- Dictionaries can be created from
  - External dictionary files
  - Inline dictionary declarations



Annotation Query Language (AQL)

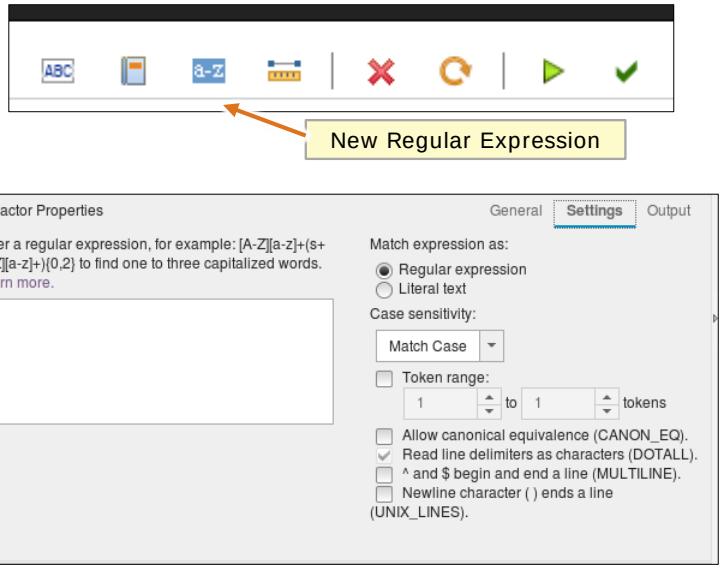
Creating dictionaries

Click the **New Dictionary** button on the toolbar. Then you specify the name of the dictionary. You then have one of two ways to load that dictionary. Either by directly typing in each term or specifying a file to load into the dictionary – this is all done through the **Extractor Properties**.

IBM Training

Regular expression

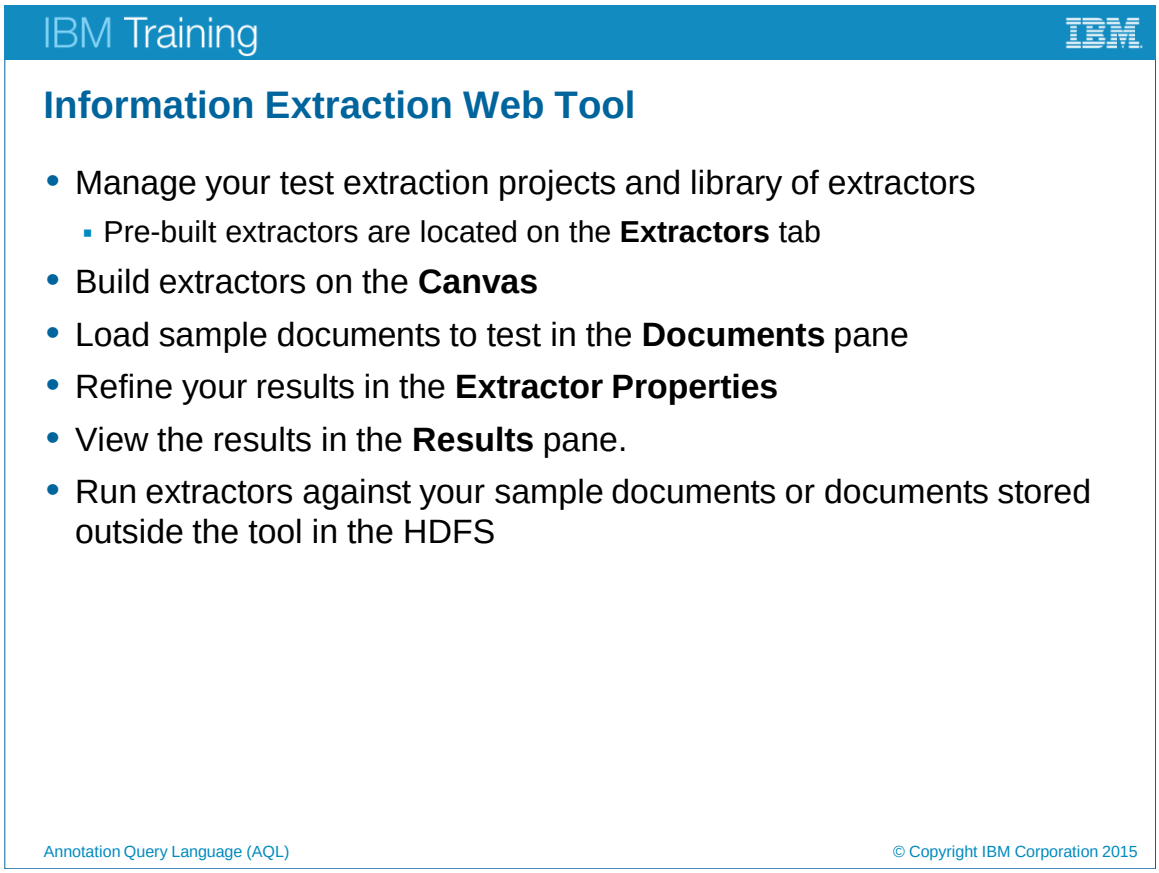
- Add a new regular expression in the **Extractor Properties**



Annotation Query Language (AQL)

Regular expression

Here's the screenshot to create a regular expression extractor. You input your own regular expression and the tool will extract based on your input.



Information Extraction Web Tool

Formally, the name of the Text Analytics tool is called the Information Extraction Web Tool. You will commonly see this called the web UI or just the Text Analytics tooling within BigInsights. Essentially, in this release of BigInsights, everything you can do is within this web UI. You manage your projects and extractors on the **Projects** or **Extractors** tab. You play around with your extractors on the Canvas. The Documents pane allows to load sample documents or documents directly on the HDFS. You refine your extractors via the **Extractor Properties**. You test your extractors and see the output on the **Results** pane.

You can also develop extractors directly using AQL, but this is not covered within the scope of the course. There are more information and guidelines for creating extractors using AQL in the Knowledge Center.

IBM Training

Pre-built extractors (1 of 2)

Category	Use
Finance Actions	Extractors that identify and extract information about corporate financial activities, such as acquisitions and mergers or earnings reports and the parties involved.
Named Entity Recognition	Extractors that identify and extract information about people, locations, organizations, and contact methods.
Generic	Extractors that generally extract information on the basis of a single word or number, such as capitalized word or a currency amount.

Annotation Query Language (AQL)

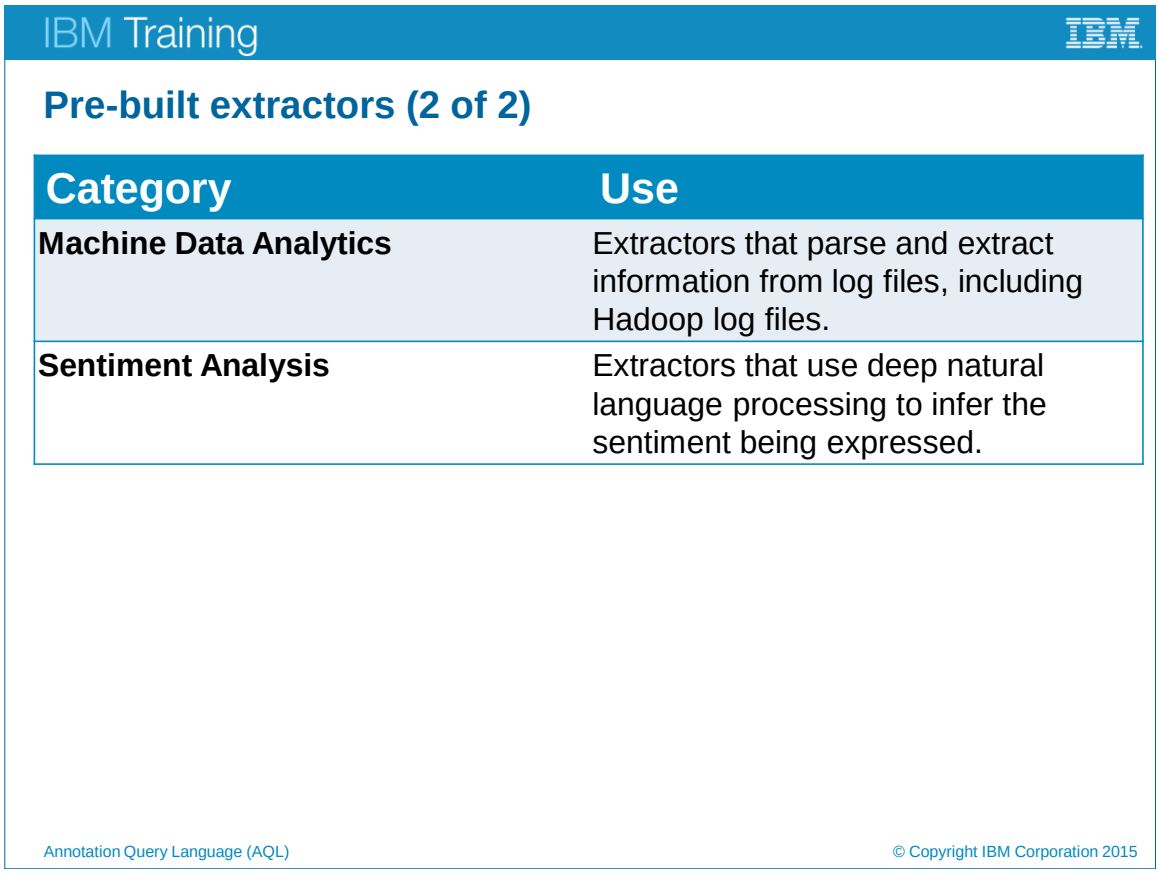
Pre-built extractors (1 of 2)

There are pre-built extractors included with the BigInsights Text Analytics tooling. The extractors falls into 5 different categories today. These categories are listed across two slides.

There are finance related extractors that extracts information about corporate financial activities.

There are named entity recognition to extract information about people.

There are generic extractors to extract basic information such as single word or number, or a currency amount.

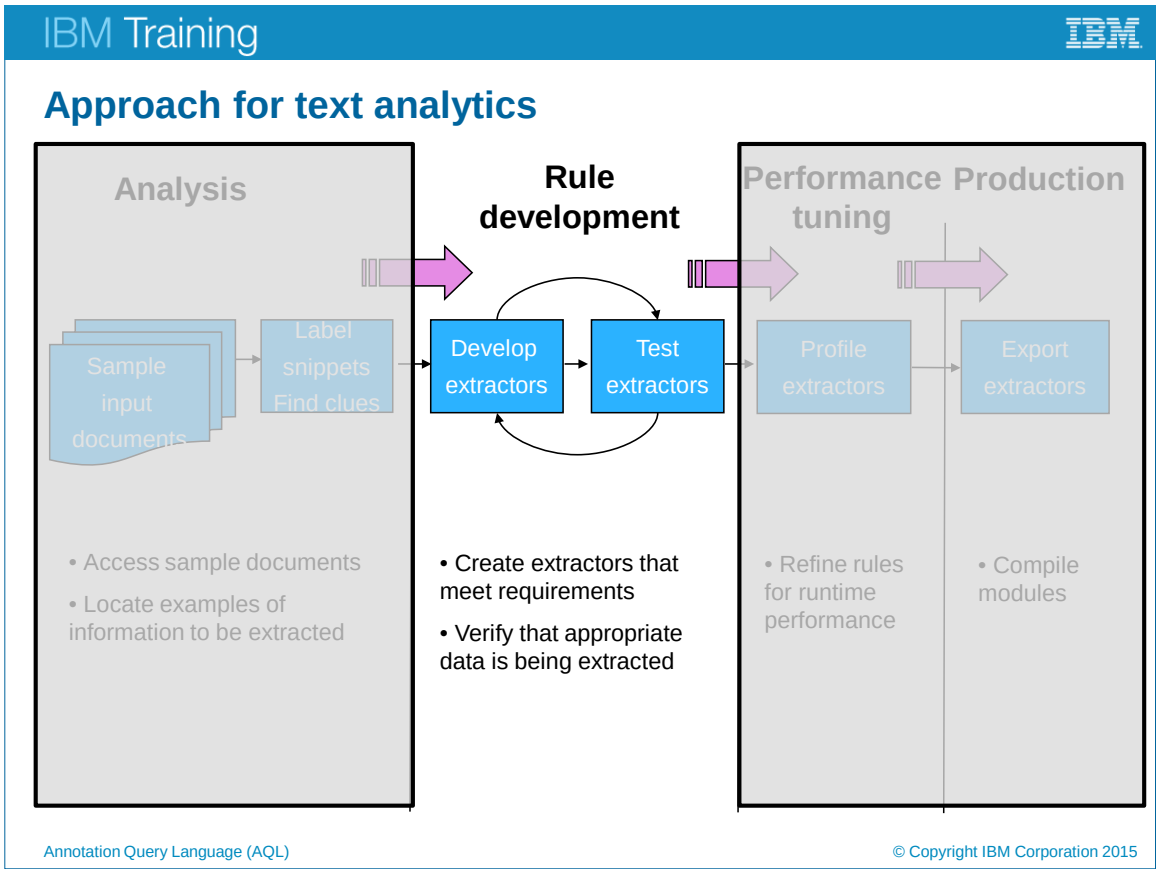


Pre-built extractors (2 of 2)

There is a category for machine data analytics with extractors that parse and extract from log files, including Hadoop log files.

There is a sentiment analysis category for extractors that use deep natural language processing to infer sentiment.

You will work with some of these in a later lab demo.



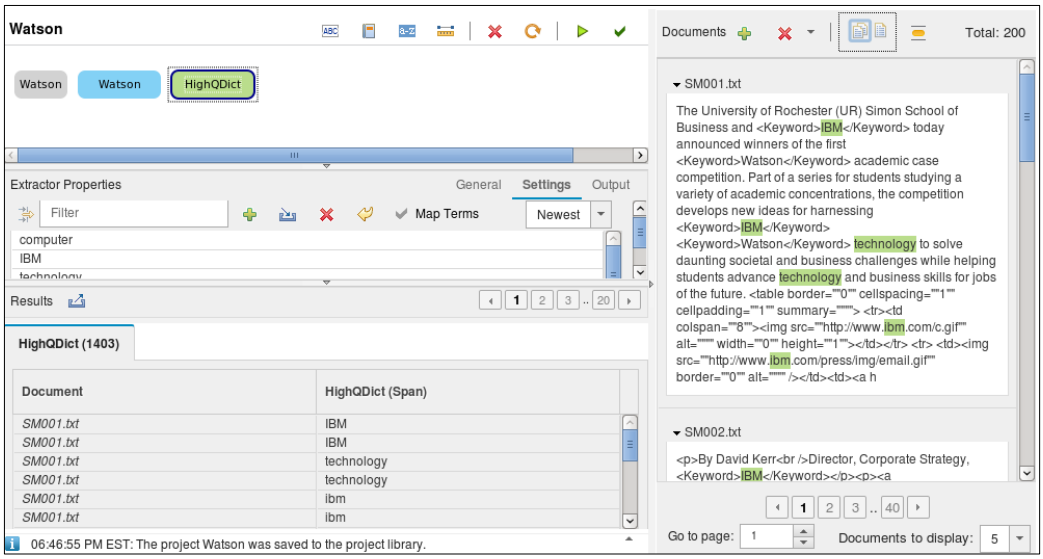
Approach for text analytics

In the following lab demo, you will be developing and testing the extractors with the Watson example that we started off in the previous demo.

IBM Training

Demonstration 1

Creating dictionaries for your Watson project



Demonstration 1: Creating dictionaries for your Watson project

Demonstration 1: Working with basic AQL features using the Web UI

Purpose:
The purpose of the demo is to get you started with the extractors using the Web UI. You will build dictionaries using the clues that you identified in the previous demo. Everything is done by dragging and dropping the extractors onto the canvas.

User ids / Passwords

OS:	**biadmin/biadmin**
Root:	**root/dalvm3**
Ambari:	**admin/admin**
BigInsights Home:	**guest/guest-password**

Ambari Services Required:

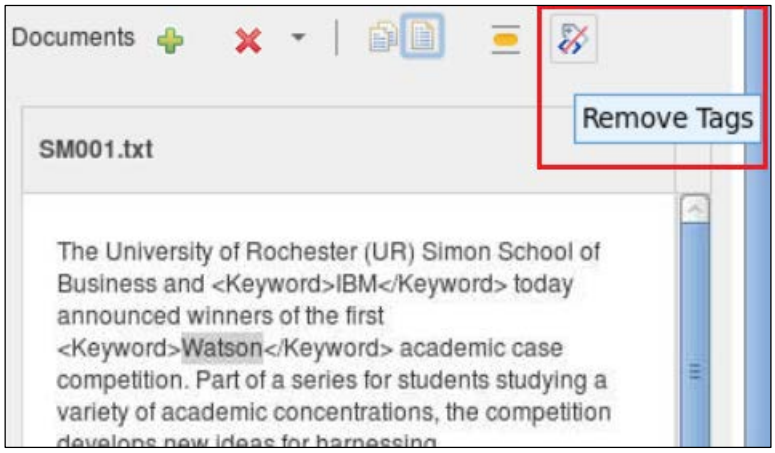
- HDFS
- MapReduce2
- YARN
- Knox (also start the Demo LDAP service)
- BigInsights - Text Analytics
- BigInsights - Home

Task 1. Creating and running a literal.

1. Continue on with the same project, **Watson**.
2. On the toolbar on top of the canvas, click the **New Literal** button.
3. Type the word **Watson** in the literal extractor**.**
4. With the literal selected, click the **Run** button (green arrow).

5. It should return 298 rows where the word Watson was found within the documents.

   If you do not get 298 rows, it's probably because you didn't remove the tags when you loaded the document. Remove the tags by clicking on the Remove tag button:



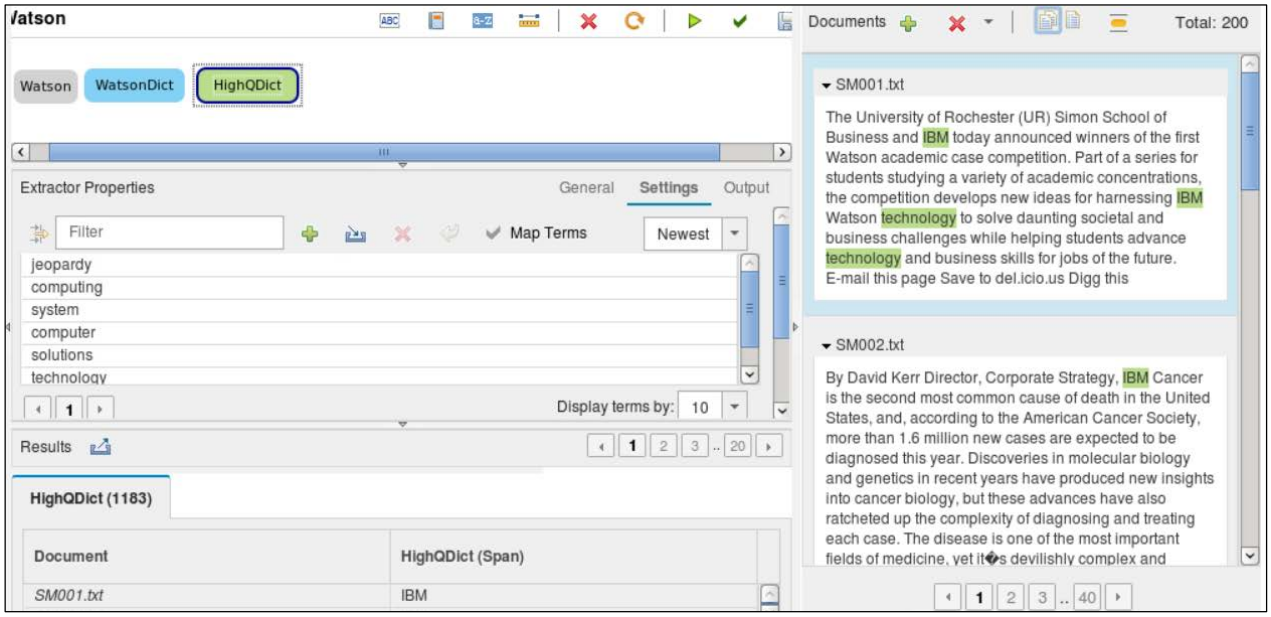
Task 2. Creating and running a dictionary.

1. Create a new dictionary. Click the **New Dictionary** button.
2. Name the dictionary: **WatsonDict**
3. Add the word Watson. Type **Watson** in the field under the **Extractor Properties.**

4. Run the extractor by clicking on the **green arrow**.
5. The results returned should be 298 rows as well, since we searched on the exact same term.
6. Now create another dictionary called **HighQDict**, with all the terms you identified in the previous lab. Add the word **computing** to the list as well. Here they are:

   Positive clues: IBM, technology, solutions, computer, system, jeopardy, computing.
7. Run the HighQDict extractor.

8. You should get 1183 results returned:



Results:
you should be able to build dictionaries using the clues that you identified in the previous example.

IBM Training

Unit summary

- Describe the AQL data model
- List the AQL components
- List the AQL objects that are used to create basic features
- Describe the Information Extraction Web Tool

Unit summary

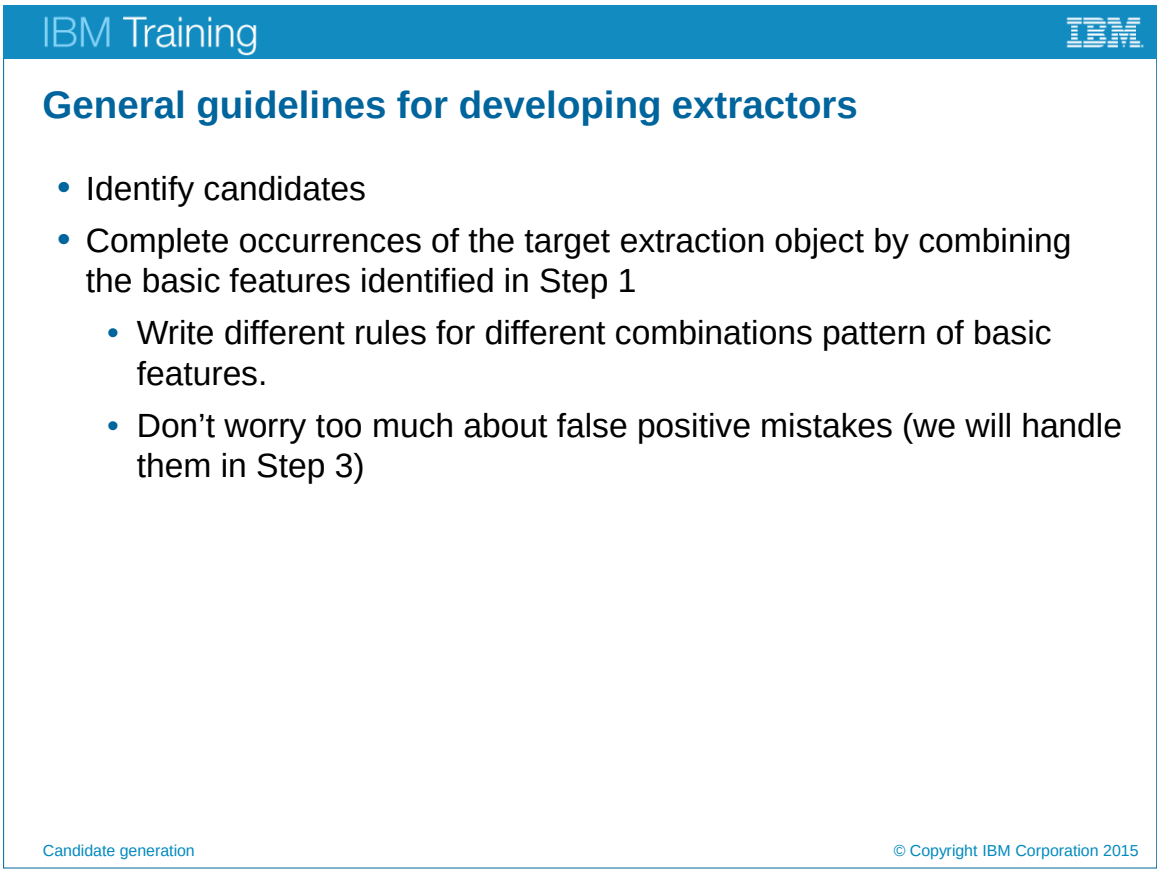
Candidate generation

IBM Training

Unit objectives

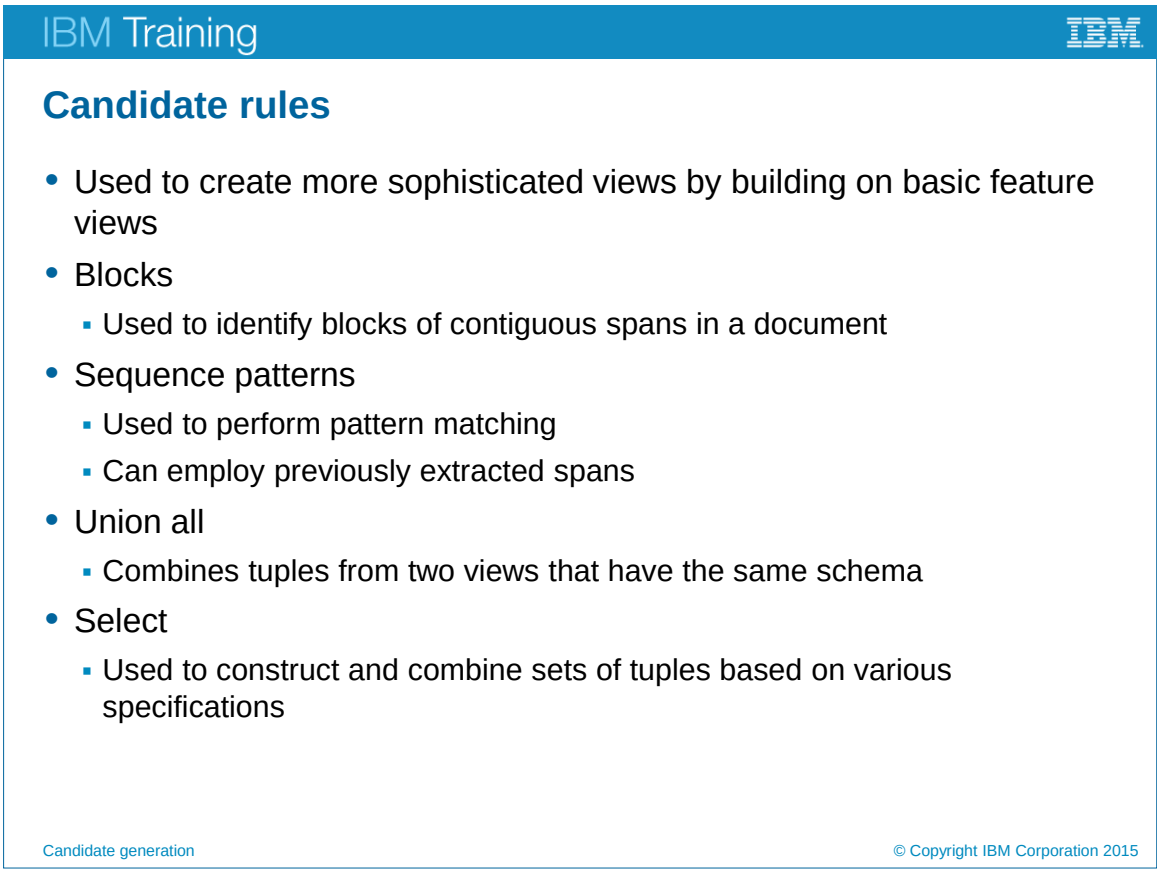
- Understand the general guidelines for developing extractors
- List the AQL candidate generation components
- Explain the use of sequence patterns, proximity rules and union

Unit objectives



General guidelines for developing extractors

Remember from the previous unit, that there are three general steps for developing AQL for text analytics. Working from the bottom up, you have the (1) basic features, (2) candidate generation, and (3) consolidation and filtering. Step 1 is where you extract the basic information for the documents. Step 2 is where you combine those extractions into something more meaningful. It is ok in this step to get false positives (more terms or occurrences than there really should be). In step 3, you will consolidate and filter out everything to get a clean set.



Candidate rules

Remember that all these can be done through custom AQL coding, but in our web UI, you have the ability to use the canvas and not have to worry about coding any AQL.

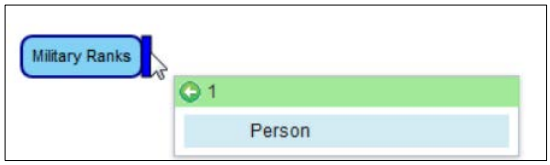
Basic features return individual building block components. You can think of each object created using a basic feature as being a brick when building a house. By itself, it is of very little use. But when combined with other bricks, you can then have a wall of the house which is a much more significant component of the house. Candidate rules allow you to combine basic feature objects to create more sophisticated views.

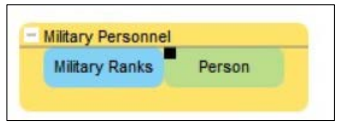
For example, finding all of the occurrences of the world *million* in your document, in and of itself, does not help very much. The same is true for finding all numbers. But being able to find a number that is followed by the world *million* now becomes more helpful.

IBM Training

Sequence patterns

- Assess the text for patterns that provide context for terms of interest
- To define a sequence pattern:
  - Create the individual extractors for all needed terms
  - Drag and drop one extractor onto another to form a sequence
- Example:
  - Create a dictionary called **Military Ranks** that includes terms such as Warrant Officer, Sergeant, and Lieutenant.
  - Drag the **Person** extractor onto the canvas following the **Military Ranks** dictionary to indicate that a new sequence finds ranks then names.





Candidate generation

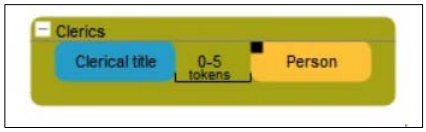
Sequence patterns

- Create individual extractors for all needed terms by extending provided extractors, or creating dictionaries, regular expressions, literals, and so forth.
- Drag and drop one extractor to another extractor on the canvas, aligning your cursor to reflect the order in which the term appears in the text pattern. A dark, bold blue line to the left or right of the extractor on which you are dropping the new extractor indicates the relative positions of the extractors. After you drop the new extractor, a box surrounds the two extractors to indicate the sequence. The box has a temporary title, Sequence n.
- Optional: Select the sequence on the canvas and rename it in Extractor Properties under General.
- Optional: If needed, repeat steps 1 and 2 to add additional elements to the pattern.

IBM Training

Proximity Rule

- Special type of element in a sequence pattern
- Each word or character is referred to as a *token*
- Specify the maximum number of tokens that might occur between the desired terms
- Example:
  - Create a dictionary called **Clerical title** that includes terms such as *Rabbi*, *Father*, and *Archbishop*
  - Drag the **Person** extractor to the right side of the **Clerical title** dictionary.
  - Right-click on **Clerical title** and click **Add After > Proximity Rule**. To capture terms such as *Archbishop of Canterbury, Robert Runcie*, specify the minimum and maximum number of tokens between words, in this case 0-5



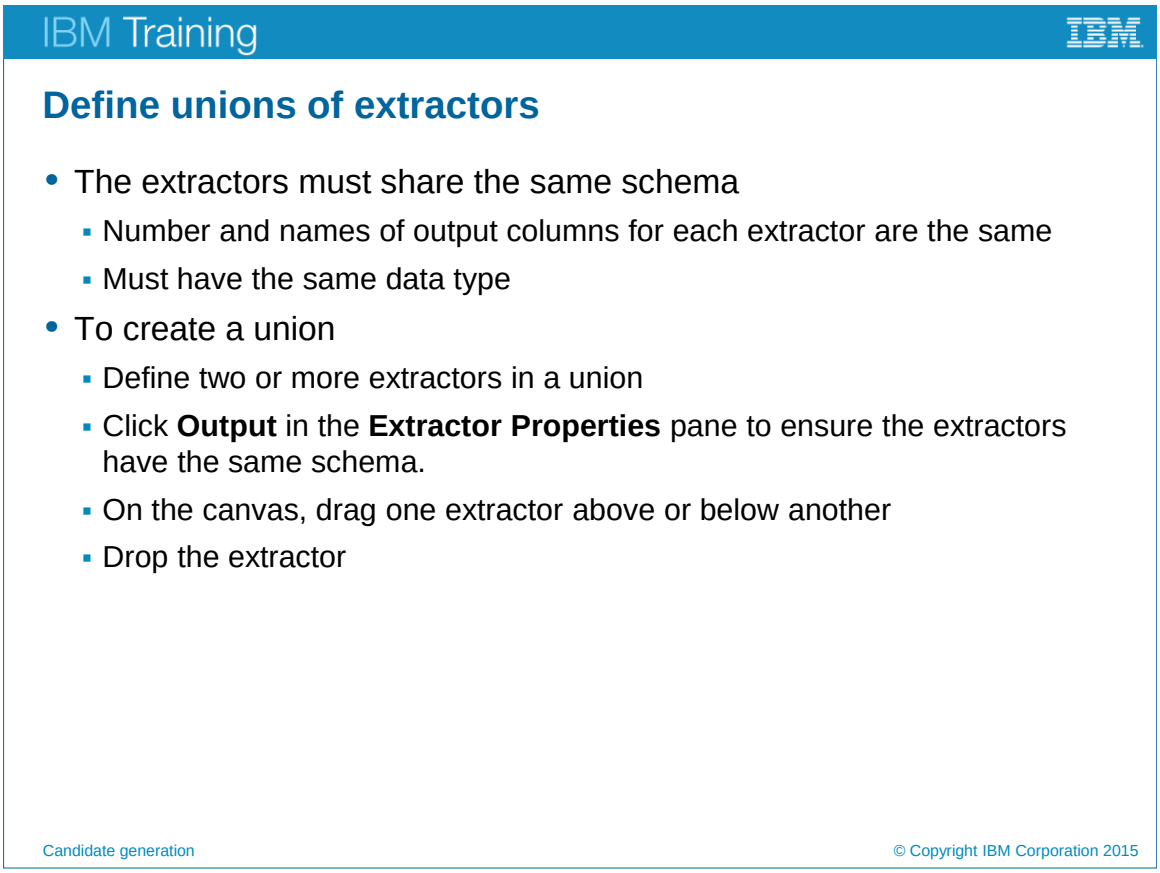
Candidate generation

Proximity Rule

Proximity Rule is a special type of element in a sequence pattern. It allows you to specify the minimum and maximum number of tokens that might occur between the desired terms. For example, suppose you want to extract clerical titles and the person to which it belongs.

You create a dictionary called **Clerical title** that includes terms such as *Rabbi, Father*, and *Archbishop*. Drag the **Person** extractor to the right side of the **Clerical title** dictionary and let go when you see the blue bar. This will create a sequence of these two entities. Right-click on the **Clerical title** and click **Add After > Proximity Rule** to indicate that you want to capture phrases that occur between 0-5 tokens. In this example, you want to be able to extract *Archbishop of Canterbury, Robert Runcie.*

As a second example, select tweets that refer to Twitter names of industry analysts with a big data term. To accomplish this, create two dictionaries, one of twitter names of analysts and a second of big data terms and combine them on the workspace canvas with a proximity of one to 25 tokens.



Define unions of extractors

The concept of unions should not be totally foreign to you. If you have some experience with SQL, you understand that a union of any two or more elements requires that their schemas be the same. It is no different here. The extractors to be combined in a union must have the same schema, so as part of the process of creating a union, you must ensure that the number and names of the output columns for each extractor are the same. They also must have the same data type. In this case, data type is span, number, string, character, date, or time. Note that you cannot edit these properties for an extractor or sequence while in a union.

To create a union, you must first have two or more extractors or sequences. Click Output on the Extractor Properties pane of each extractor to ensure that they all have the same schema and data type. On the canvas, drag one extractor above or below another and then drop the extractor to create the union.

IBM Training

Example of a union of extractors (1 of 2)

- The term *parties* in a contract can refer to two individuals, two orgs, or one of each.
- Drag both the **Person** and the **Organization** extractors from the **Extractor** pane onto the canvas
- On the canvas, select each extractor and on the **Output** tab of the **Extractor Properties**, rename the output columns so that the column definitions are the same for both

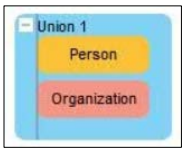
Candidate generation

Example of a union of extractors (1 of 2)

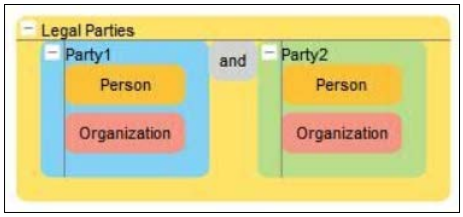
Here's an example. The term *parties* in a contract can refer to two individuals, two organizations or one individual and one organization. We want to find occurrences of all the legal parties. Drag both the **Person** and the **Organization** extractors from the Extractor pane onto the canvas. On the canvas, select each of the extractors and on the **Output** tab of the **Extractor Properties** pane, rename the output columns so that the definitions are the same for both.

IBM Training

Example of a union of extractors (2 of 2)

- Drag the **Organization** extractor to the drop down below the **Person** extractor 

- Select Union 1 and name it Party 1
- Copy Union 1 and create a second union and name it Party 2
- To look for the parties in a contract, combine to two unions in a sequence
- Name the new sequence Legal Parties



Candidate generation

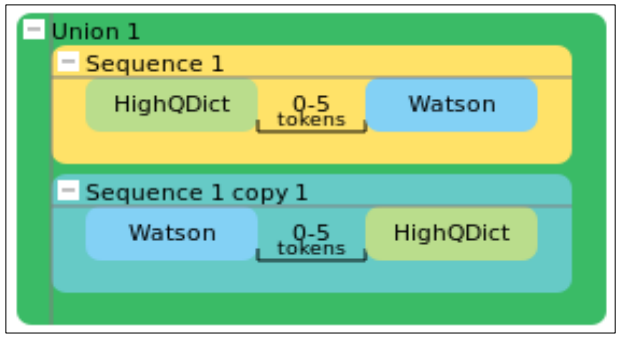
Example of a union of extractors (2 of 2)

Then drag the **Organization** extractor to drop down below the **Person** extractor. The order actually does not matter here - it's a union. Name Union 1 as Party 1. Make a copy of Union 1 to create a second union and name it Party 2. Combine both unions with the literal *and.* Name this final sequence Legal Parties.

IBM Training

Demonstration 1

Generating candidates



Candidate generation

Demonstration 1: Generating candidates

Demonstration 1: Generating candidates

Purpose:
In this demo, you will learn how to build candidates to tailor your extractors.

User ids / Passwords

OS:	**biadmin/biadmin**
Root:	**root/dalvm3**
Ambari:	**admin/admin**
BigInsights Home:	**guest/guest-password**

Ambari Services Required:

- HDFS
- MapReduce2
- YARN
- Knox (also start the Demo LDAP service)
- BigInsights - Text Analytics
- BigInsights - Home

Task 1. Creating yet another dictionary.

1. Click the **New Dictionary** button.
2. Name the dictionary, **ResearchDict**.
3. Add the terms, **research center** and **research centre**.

   You will use this dictionary later to eliminate occurrences of the word Watson that has to do with the research centers.

Task 2. Creating a proximity rule.

1. Click the **New Proximity Rule** button.
2. Specify **0-5** tokens for the proximity rule.

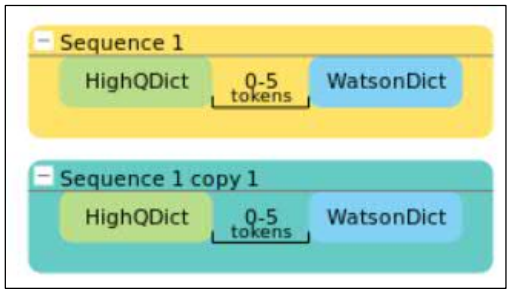
   This proximity rule will allow us to look for terms that are within five tokens of the words around it.
3. Join the **HighQDict** and the **WatsonDict** extractor with the proximity rule in the middle. Drag the proximity rule to the right of the **HighQDict**. You will see a blue line indicating that the two items will join when you let go of the mouse button. A **Sequence 1** box will appear with the two items in it.

4. Now, drag the **WatsonDict** extractor to the right side of the proximity rule and combine it.

The Sequence we just created looks for terms with the word Watson following the list of positive clues that we identified. To accurately capture all possibilities, we need to also define a sequence that has the word Watson preceding those same words.

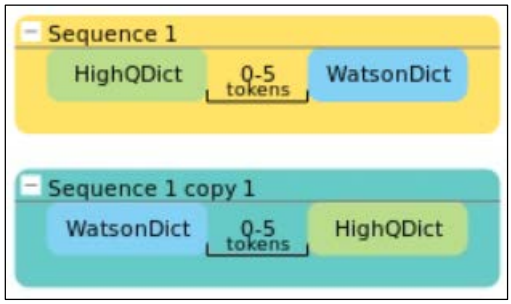
Task 3. Making a copy of an extractor.

1. Right-click on the **Sequence 1** extractor and select **Copy** from the menu.
2. Right-click somewhere on the canvas (outside of the Sequence 1 extractor) and select **Paste as New Copy** to create a new sequence.



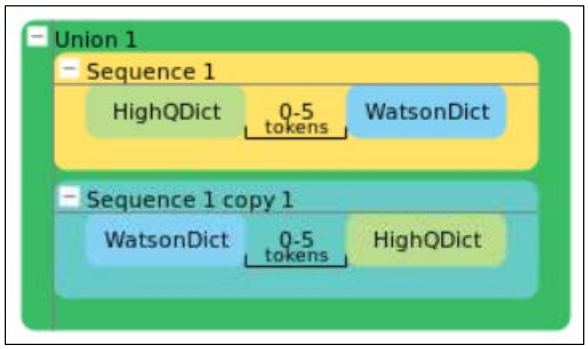
The **Paste as New Copy** makes a separate copy of the original extractor. Changes that you make to original extractor will NOT affect the copy.

3. Now we need to change the order of the extractors in **Sequence 1 copy 1** to **WatsonDict** , **0-5**, **HighQDict**. Drag and drop to rearrange the order.



Task 4. Combining extractors through a union.

1. Drag and drop **Sequence 1 copy 1** to the bottom of the **Sequence 1** extractor to create a union of the two.



2. Run this new **Union 1** extractor. You should get 178 rows returned.

   Note: If you are not getting 178 rows, check that the tags have been removed from the Documents.

Task 5. Using regular expressions in extractors - Extra credit.

This is an extra credit task in the sense that it does not continue with the normal Watson storyline that we have been doing. This task is to show how you can take advantage of the regular expression extractor.

1. Create a new project. Call it **Regular Expression**.
2. Add a document. Click the **green plus** button.
3. Browse for the file located under **/home/biadmin/labfiles/TextAnalytics/**
4. Add the file **Facts.txt**
5. Examine the **Facts.txt** file. Open this file up on your local file system to be able to see more of the content. The Document pane only shows a subset of the full document.
6. About 29 lines down in the file, you should see

   Geography Afghanistan.

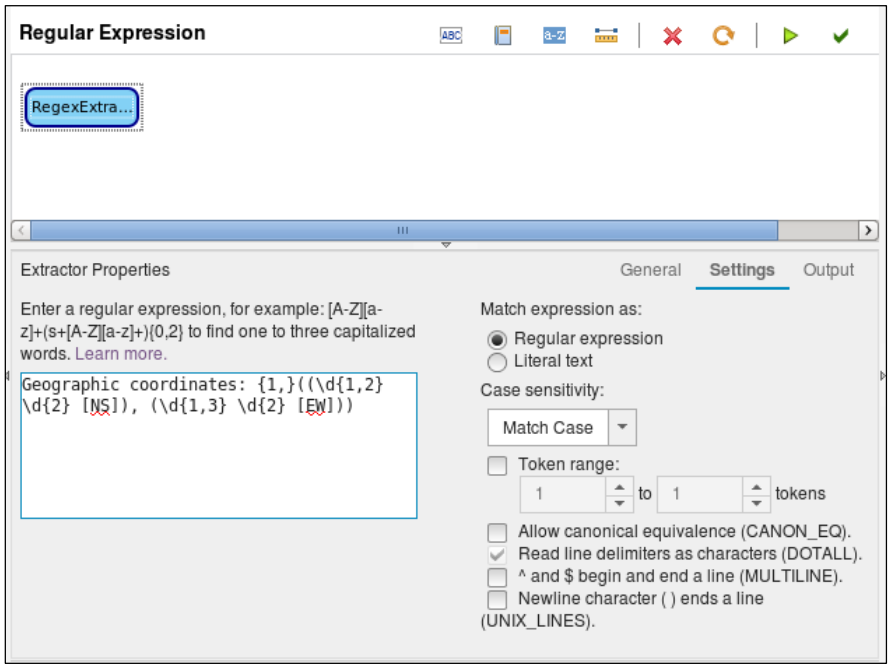
   A few lines further down, you should see

   Geographic coordinates: 33 00 N, 65 00 E

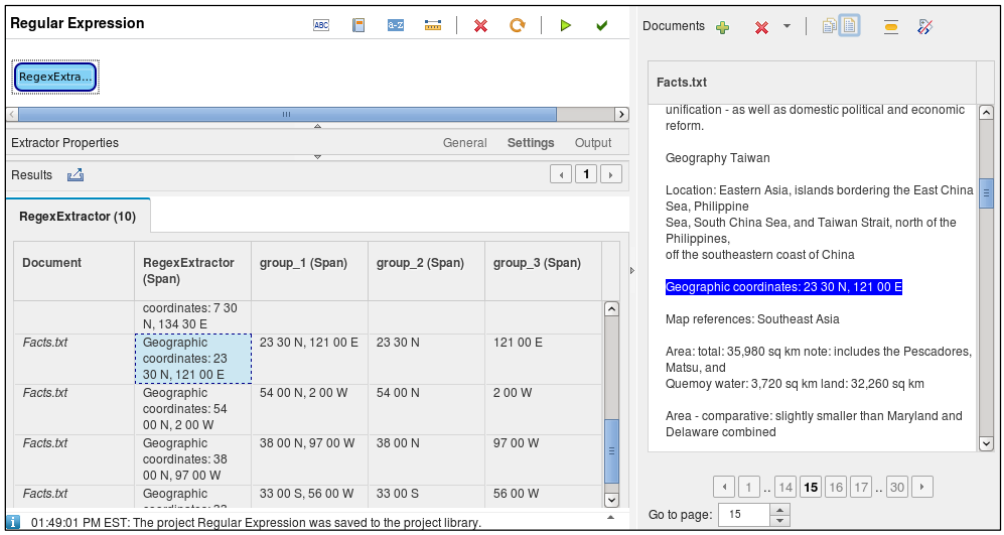
   You are going to create and use a regular expression extractor to extract the geographic coordinates.
7. Click the **New Regular Expression** button.
8. Name this extractor, **RegexExtractor.**

9. In the **Extractor Properties,** on the regular expression field, enter in this regular expression that will extract the geographic coordinates.

   Geographic coordinates: {1,}((\d{1,2} \d{2} [NS]), (\d{1,3} \d{2} [EW]))



10. Keeping everything else the same, go ahead and run the extractor.
11. You can view the 10 rows that were returned and see where within the file they are located.



Results:
You have learned to build candidates to tailor your extractors. Optionally, if you went through the regular expression section, you should be able to use the regular expression extractor to extract texts based on the provided regular expression.

IBM Training

Unit summary

- Understand the general guidelines for developing extractors
- List the AQL candidate generation components
- Explain the use of sequence patterns, proximity rules and union

Unit summary

Filter and consolidation

IBM Training

Unit objectives

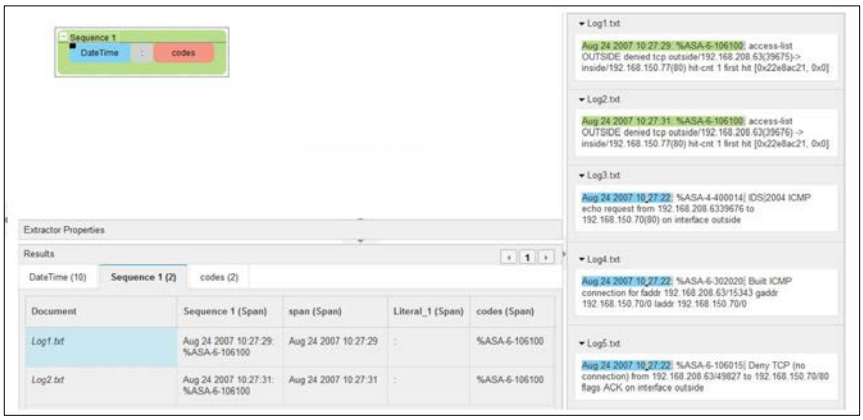
- Run an extractor and refine the results
- Eliminate duplicates and overlaps
- Filter the results
- Export the results

Unit objectives

IBM Training

Run an extractor and refine results

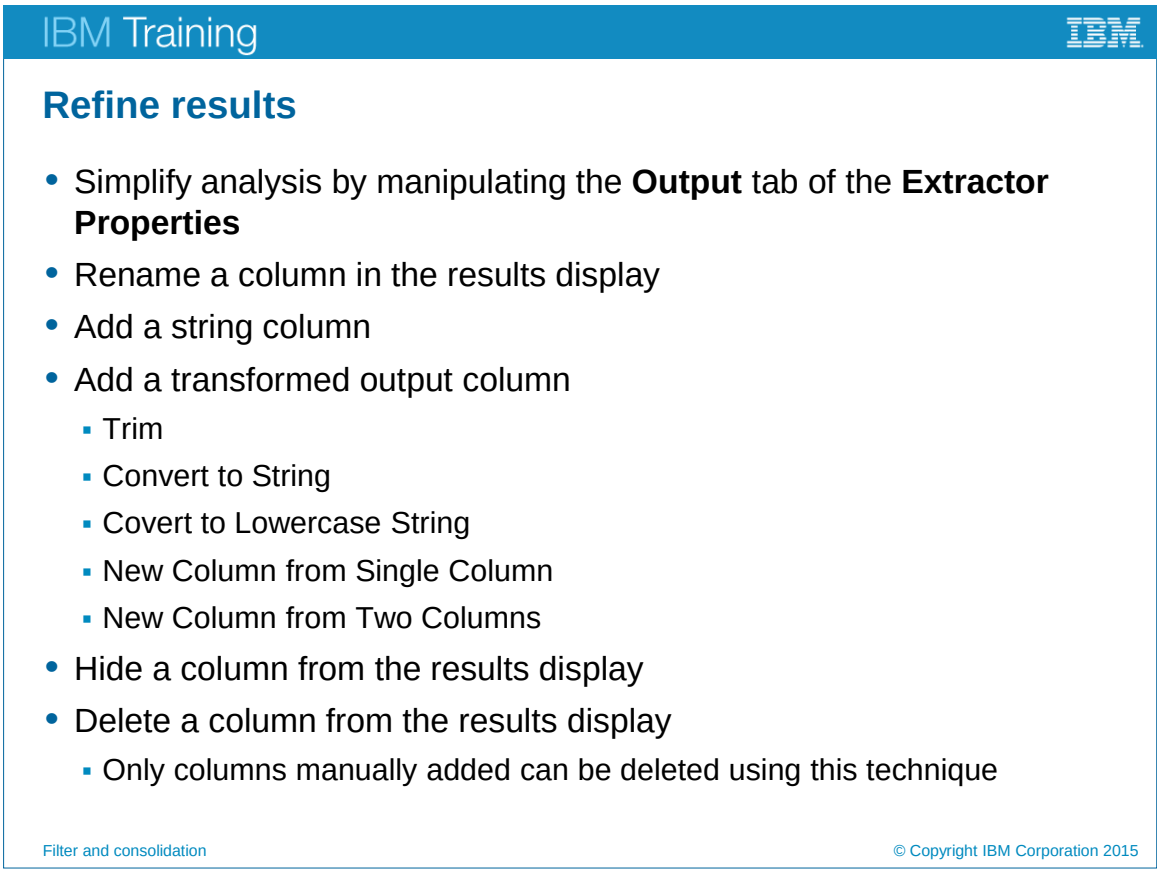
- Test your extractor by running from the workspace
  - Right-click the extractor and click **Run Selected** from the menu
  - Results appear in the **Results** pane as well as the **Documents** pane
- Review and refine as needed
- Save it to the extractor library to be used in other projects



Filter and consolidation

Run an extractor and refine results

Test the extractor and then refine the results as needed. Once you have refined the results to your satisfaction, you save the extractor to the library to be used by other projects.



Refine results

Rename a column in the results display

- On the canvas, right-click the extractor that generated the results and click Edit Output.
- From the column menu, select Rename or simply double click the column.
- Enter the new column name to be displayed in the results.

Add a string column

- On the canvas, right-click the extractor that generated the results and click Edit Output.
- Click the Manage Columns menu in the left column of the table.
- Click New Column.

Add a transformed output column – for example: converting it to all lowercase

- On the canvas, right-click the extractor that generated the results and click Edit Output.
- Click the drop-down menu in the header of the column that you want to transform and select the type of transformation that you want to do.

Hide a column from the results display

Important: If you hide output columns, which are part of the sequence that is being matched, then when you use that sequence inside another sequence, the output columns affect the match criteria for the outer sequence. For example, if you create a sequence called Money, which is a sequence of a literal $, followed by a number, followed by a dictionary match, and you update the output to hide the $, then if you useMoney inside of a larger sequence, the outer sequence match for Money will not look for the $. A better approach would be to use a filter to narrow the results to those preceded by $.

- On the canvas, right-click the extractor that generated the results and click Edit Output.
- Click the Manage Columns menu in the left column of the table.
- Clear the check boxes for the columns you want to remove from the results display. These columns are hidden from the results, although the content is still be extracted.

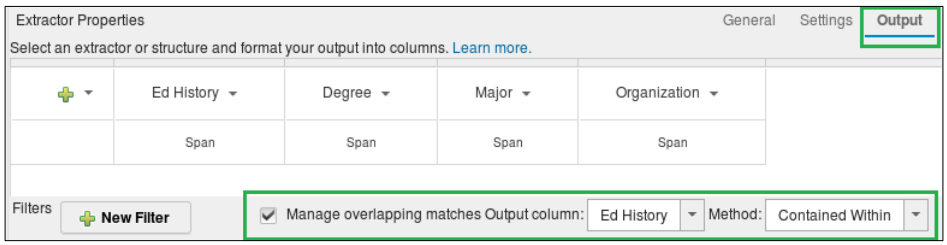
Delete a column from the results display

- On the canvas, right-click the extractor that generated the results and click Edit Output.
- Click the Manage Columns menu in the left column of the table.
- Click Delete Column and select the check boxes for the columns you want to remove from the results display.

IBM Training

Eliminate duplicate and overlapping results

- An extractor generates multiple rows for the same text
- Remove or consolidate duplicate entries by providing consolidation rules
  - Right-click the extractor that generated the results
  - Click **Output** in the **Extractor Properties** pane
  - Select **Manage overlapping matches**
  - From the **Output column** list, select the column that is causing the duplicates
  - Select one of the values from the **Method** list



Filter and consolidation

Eliminate duplicate and overlapping results

Occasionally, an extractor will generate duplicate results for the same text because the text matches more than one dictionary entries. In these cases, you will want to remove them by providing some consolidation rules. Essentially, you specify the column that is causing the duplicates and then select one of the five methods:

Contained Within to keep the longest result.

Not Contained Within to keep the shortest result.

Contains But Not Equal to keep unique results of the same length.

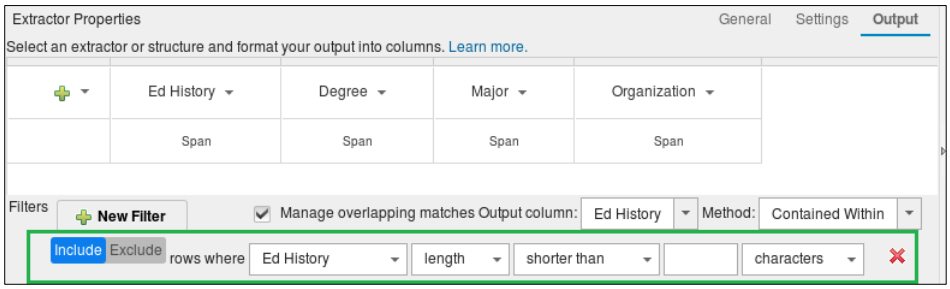
Exact Match to keep one instance of each result.

Left to Right to keep the longest result, with the greatest number of terms from left to right.

IBM Training

Refine results using filters

- An extractor sometimes produces unwanted results
- Refine the results using filters
  - Right-click the extractor and click **Edit Output**
  - Click the **New Filter** button
  - Select the Include or Exclude option
  - Specify the parameters of the filter



Filter and consolidation

Refine results using filters

Sometimes extractors will produce unwanted results and this is where you can filter them out. Essentially, you can choose to either include or exclude certain terms that you specify in the filter. The next slide shows an example of this.

IBM Training

Example of a filter

- Your Military Ranks extractor might product a match for the text *Chief Warrant Officer John Doe*, but you do not want to include results that have the word *except* preceding the match
  - Create a dictionary with the term *except* and any other terms you want to exclude
  - Open the **Output** tab
  - Click **New Filter** and select **Exclude**
  - Select **range** and **occurs after**
  - Select the dictionary
  - Select the column and **between 0** to **2 tokens**

Filter and consolidation

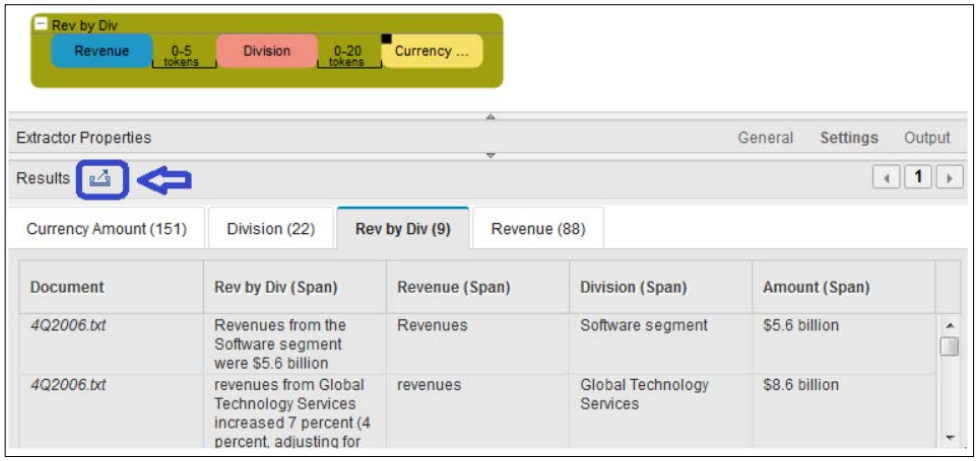
Example of a filter

This filters excludes any matches that have the word *except* within 0-2 tokens before a match

IBM Training

Export refined extractor results

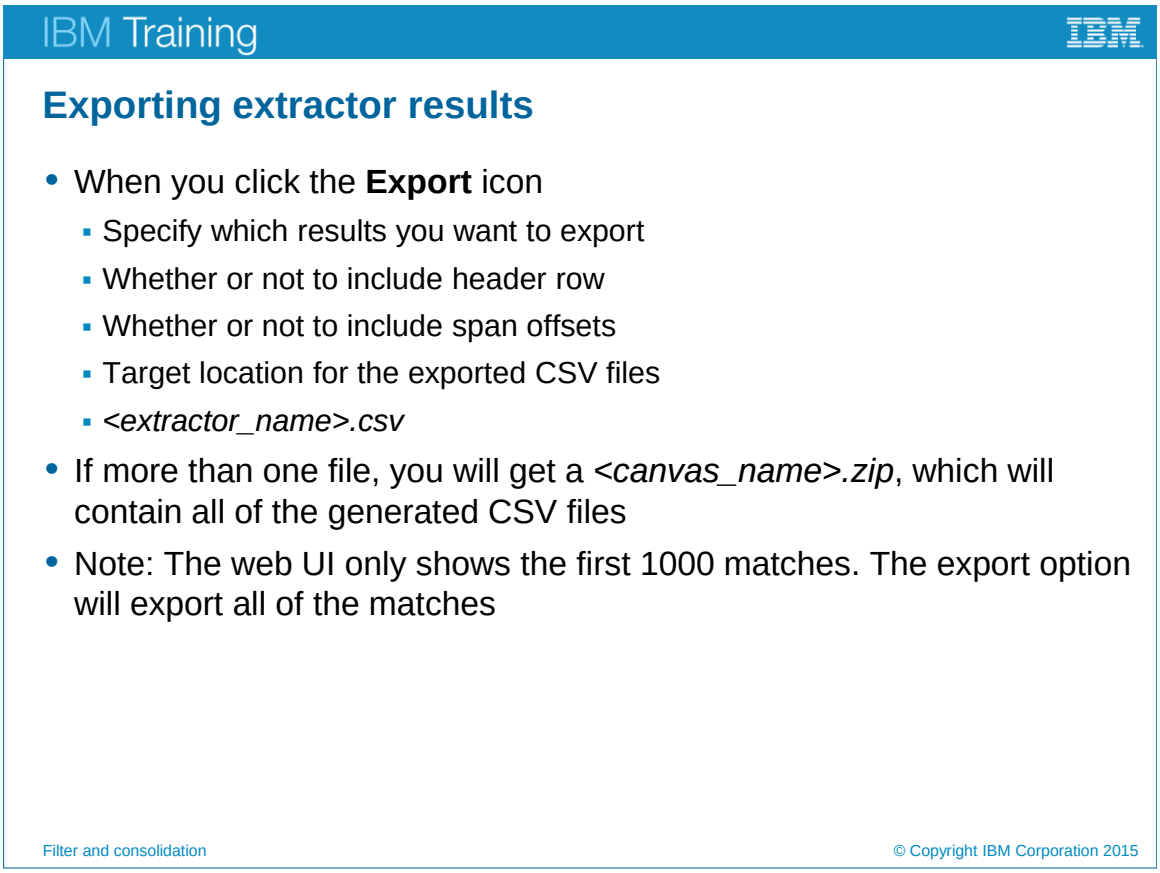
- Export results into CSV files for viewing in external applications
  - Extracted from a full set of documents
  - Do not need the extractors for a different project



Filter and consolidation

Export refined extractor results

You can export your results out a CSV file for viewing in external applications so that you do not need the specific extractors for the other projects.



Exporting extractor results

Which results to export. You can choose to export results from one or more extractors. The list of available extractors will match the list of result grids which are currently shown in the Results pane (i.e. the list of extractors which were most recently executed on the current canvas). You must select at least one extractor, or the OK button in the dialog will be disabled.

Whether or not to include a header row. Check this box if you would like each exported CSV file to include a header row. The header row for a given CSV file will contain the names of the output columns for the extractor whose results are exported to that file.

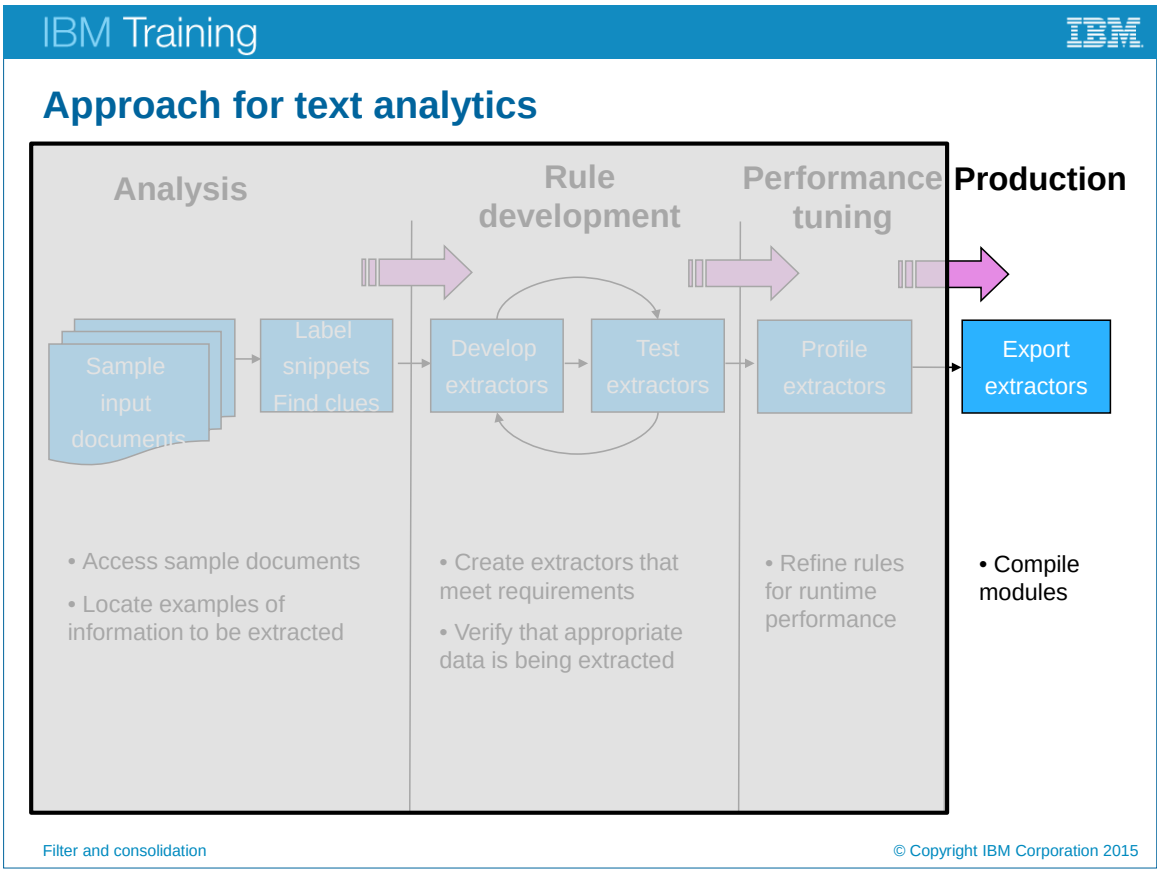
Whether or not to include span offsets within the column values for columns which return spans. If true, the data values for span columns will include begin and end offsets, as integers within brackets. For example, instead of *revenues*, a column value might be *revenues [3972 - 3980]*.

Target location for the exported CSV files. You can save the exported CSV files to your local machine, or you can upload them to a directory of your choosing on a distributed file system (DFS). The latter option is only available when the web tool is running in an environment which has access to a DFS.

Note: The web tool shows only the first 1000 matches for each extractor, and those matches are sorted by document name and span offsets. However, the Export option will export all of the matches found for the specified extractors, and those matches will not be sorted. The total number of matches which will be exported for each extractor is displayed in parentheses after the extractor name in the Results pane.

If you choose to save the files to your local machine, your browser will attempt to download a generated file named *<canvas_name>.zip*, which will contain all of the generated CSV files.Note: The default name for the zip file is always *<canvas_name>.zip*. If you have your browser configured to prompt you for a location before downloading files, you can change the name of the zip file in the download prompt. Otherwise, the file will be immediately saved to your default download location. If a file named *<canvas_name>.zip* already exists in your default download location, the behavior is browser-dependent.

If you choose to upload the CSV files to a DFS directory, all of the CSV files will be written directly to the specified directory, as individual .csv files. (They will not be packaged into a zip file.) If a generated CSV file has the same name as a file which already exists in the specified directory, the existing file will be overwritten.



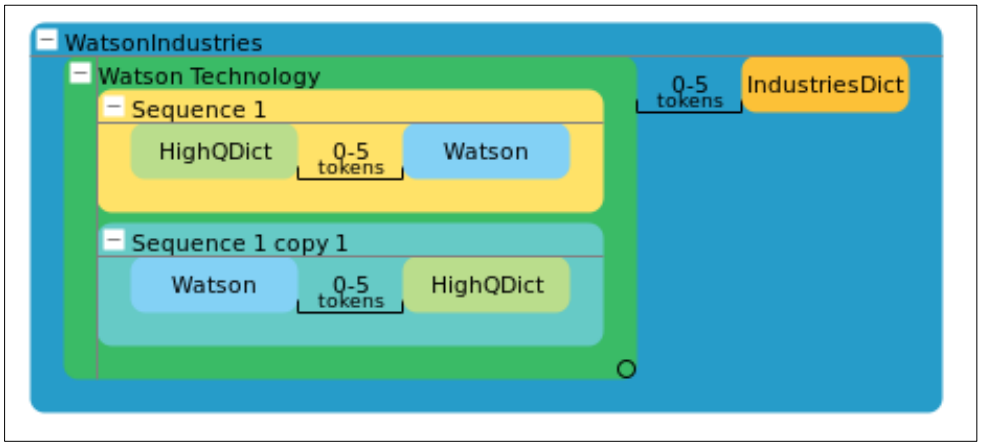
Approach for text analytics

In this demonstration, you will focus on the filtering and consolidation of the extractors and see how to export them.

IBM Training

Demonstration 1

Filtering and consolidating



Filter and consolidation

Demonstration 1: Filtering and consolidating

Demonstration 1: Filtering and consolidating

Purpose:
In this demonstration, you will filter and consolidate the results of the extractors

User ids / Passwords

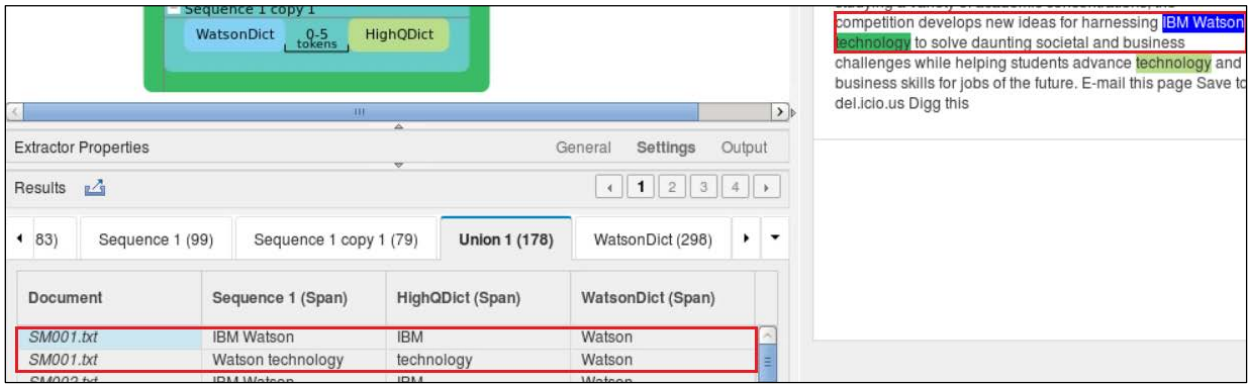
OS:	**biadmin/biadmin**
Root:	**root/dalvm3**
Ambari:	**admin/admin**
BigInsights Home:	**guest/guest-password**

Ambari Services Required:

- HDFS
- MapReduce2
- YARN
- Knox (also start the Demo LDAP service)
- BigInsights - Text Analytics
- BigInsights - Home
- BigSheets (optional)

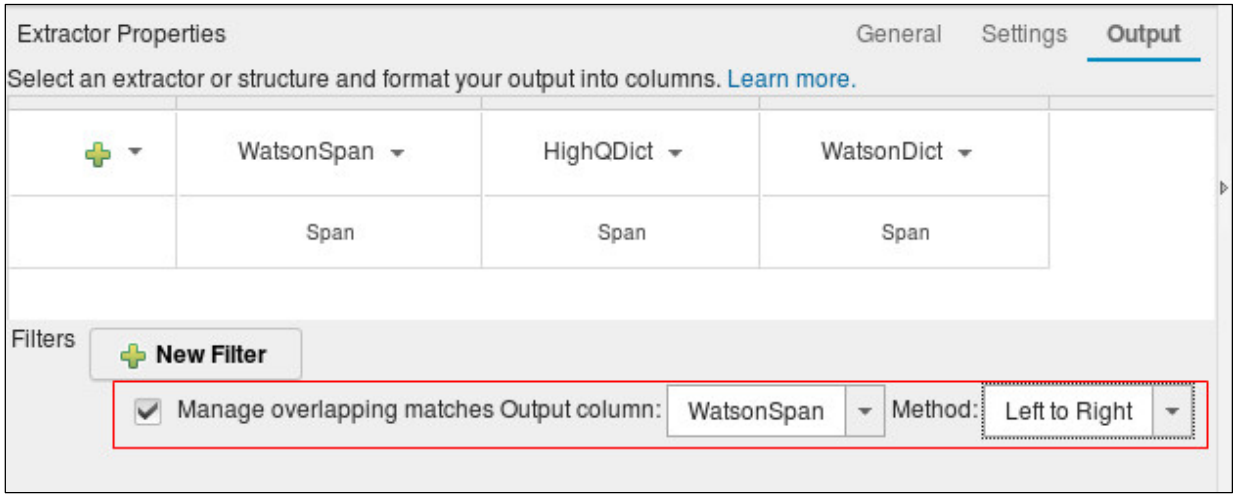
Task 1. Removing duplicate results.

1. We are going to resume with the Watson project. Open it up from the **Projects** pane.
2. Run the **Union 1** extractor again. You should get 178 rows returned.
3. Now, look at the results more closely and you see some duplicates in the first file, SM001.txt. The word Watson was selected twice. Once for the IBM clue and again for the technology clue. We only need one occurrence of this.

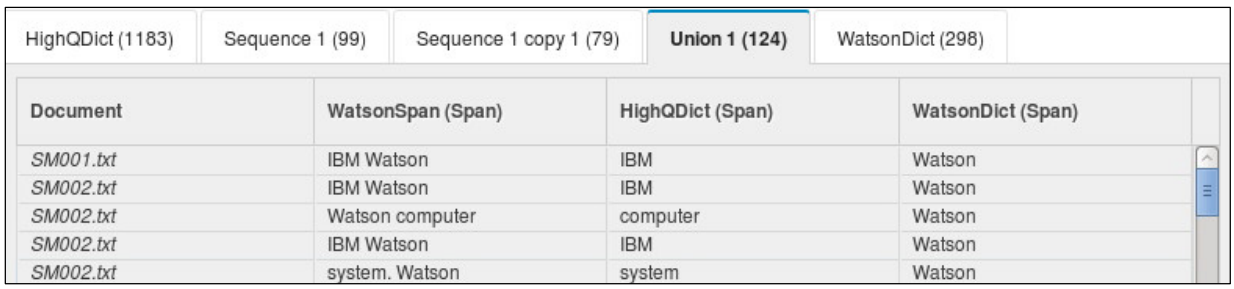


4. Take a look at the **Extractor Properties**.

5. Rename the **Sequence 1** column. On the **Output** tab**,** click the **Sequence 1 dropdown** and select **Rename**.
6. Name it **WatsonSpan**.
7. Click the Manage overlapping matches on the **WatsonSpan** column using the Method **Left to Right**.



8. Run the extractor and note that there are now 124 rows returned. More importantly, the duplicates have been removed from the results.



Task 2. Creating filters to remove instances of the Watson research center.

1. Under the same **Extractor Properties** on the **Output** tab, click the **New Filter** button.
2. Select **Exclude** rows where **WatsonSpan range occurs before** Extractor **ResearchDict** Column **ResearchDict between 0** to **3 tokens.**

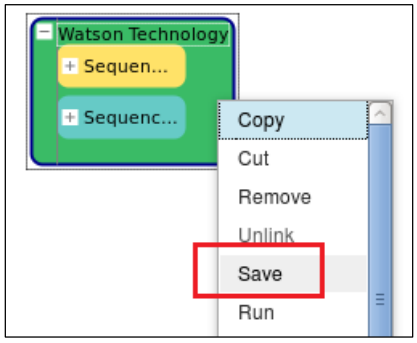
3. Run the extractor and see that the occurrence has been removed. 123 rows returned.

Task 3. Using regular expression to filter out names.

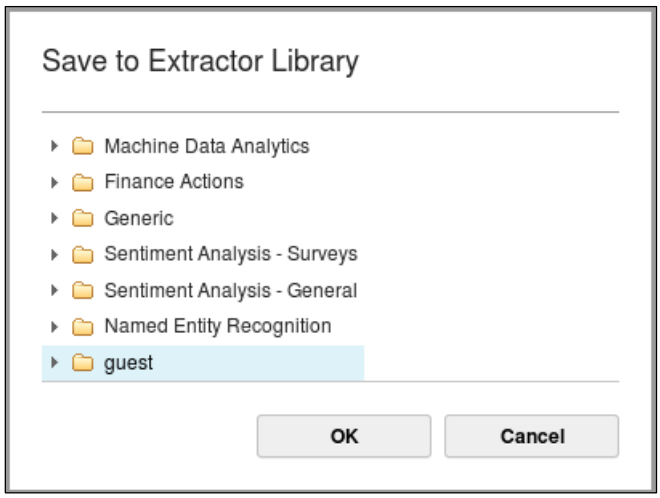
1. SM126.txt (on page 26 on the documents pane) has IBM Founder Thomas J Watson. We cannot eliminate this unless we come up with a regular expression for it.
2. I will leave this exercise up to you. Create a regular expression extractor which eliminates names such as the one located on the SM126.txt file.

Task 4. Saving the extractor.

1. Rename the Union 1 extractor as **Watson Technology**.
2. Save the extractor. Right-click and select **Save.**



3. Save it as **Watson Technology** under the **guest** folder**.**

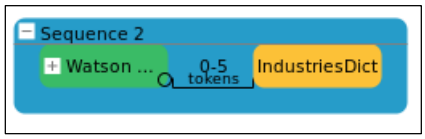


4. Now you can drag and drop this extractor from the catalog to use it in other projects.

Task 5. Working with Watson industries.

1. Continuing with the same project, create a new dictionary: **IndustriesDict**
2. This time, import the dictionary from the **/home/biadmin/labfiles/WatsonData/Dictionary/Industries.dict** file

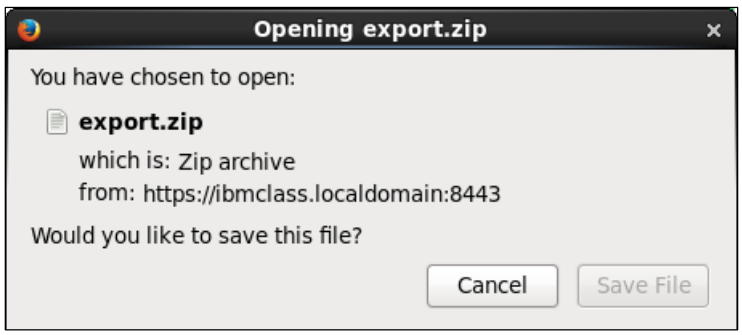
3. Create a proximity rule of **0-5 tokens**.
4. Join the **Watson Technology** extractor with the **IndustriesDict** using the proximity rule.



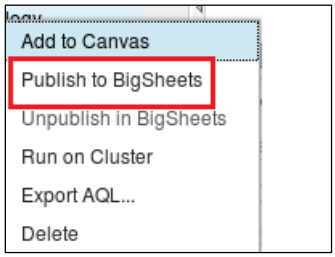
5. Run the extractor.
6. Likewise, you can create a union of these and put IndustriesDist preceding the proximity rule. I leave this optional exercise up to you.
7. Rename and Save this Sequence 2 extractor as **WatsonIndustries**

Task 6. Exporting the results.

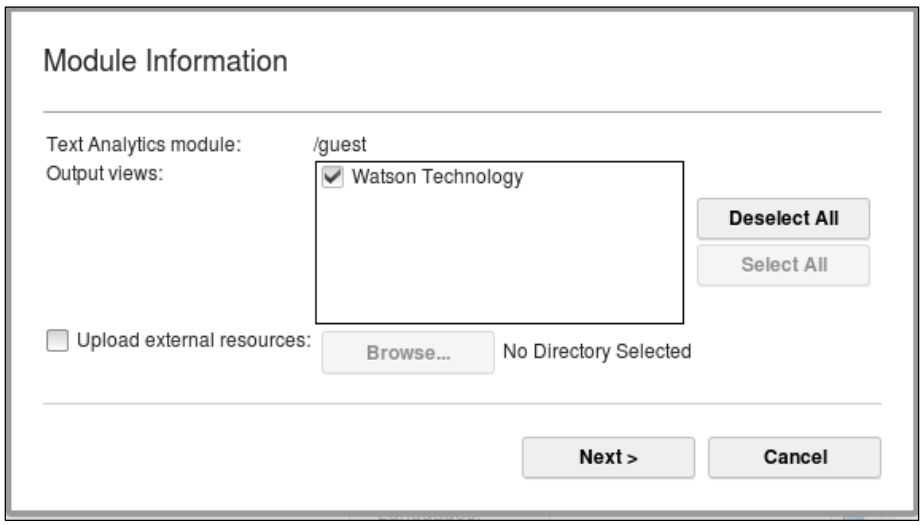
1. On the Extractor catalog, right-click any of the extractors under the guest folder and select the Export AQL… option to get the AQL code out as a zipped file.



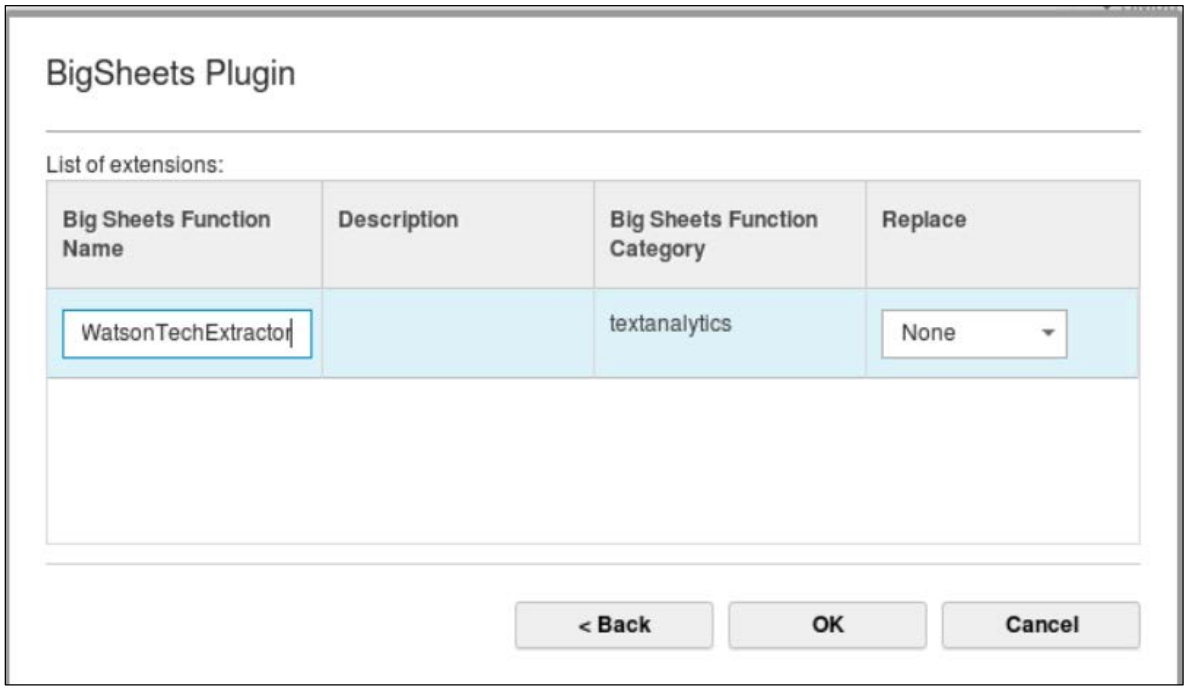
2. Click Cancel to the Opening export.zip dialog. We are not going to export.
3. You can export the extractor as a function within BigSheets. The BigSheets service has to be installed and started for the function to work. If you want to try this now, go ahead and start up the **BigInsights-BigSheets** service in Ambari.
4. Once the BigSheets service has started, restart the **BigInsights-Home** service.
5. Refresh the **BigInsights-Home** page in the **Firefox** Browser. You should see the BigSheets panel enabled (no longer greyed out).
6. Click the **Text Analytics** link to go back into your projects.
7. Back on the Extractor catalog on the left side, under the guest folder, right-click the **Watson Technology** extractor and select **Publish to BigSheets**.



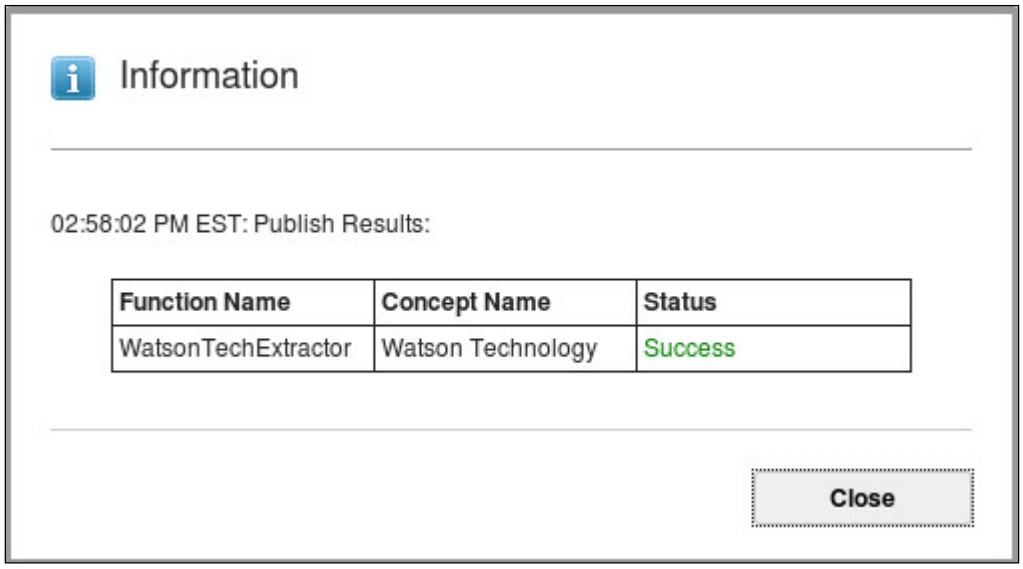
8. Click **Next>**.



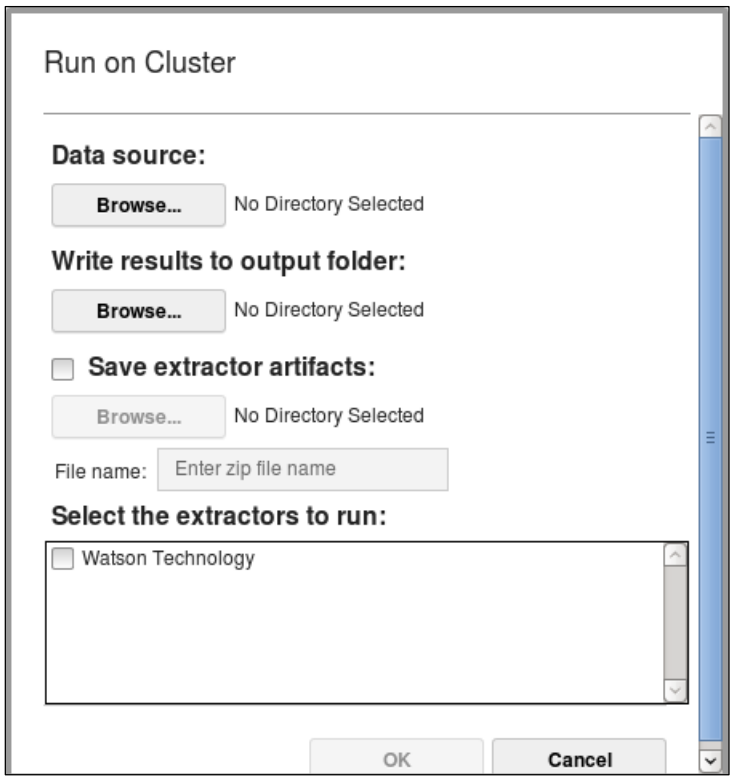
9. Rename the extractor. Double click the **Watson_Technology** name. Name it **WatsonTechExtractor**.



10. Click **OK** to publish it.
11. Click the **Close** button.



12. From the **Extractors** tab **guest**>**Watson Technology** right click and select **Run on Cluster**.
13. A final way to use your extractor outside of **Text Analytics** is to run it on the Cluster. Select the **Run on Cluster** option.



14. Specify the options that you wish and run the extractor. I leave this as an optional exercise for you.

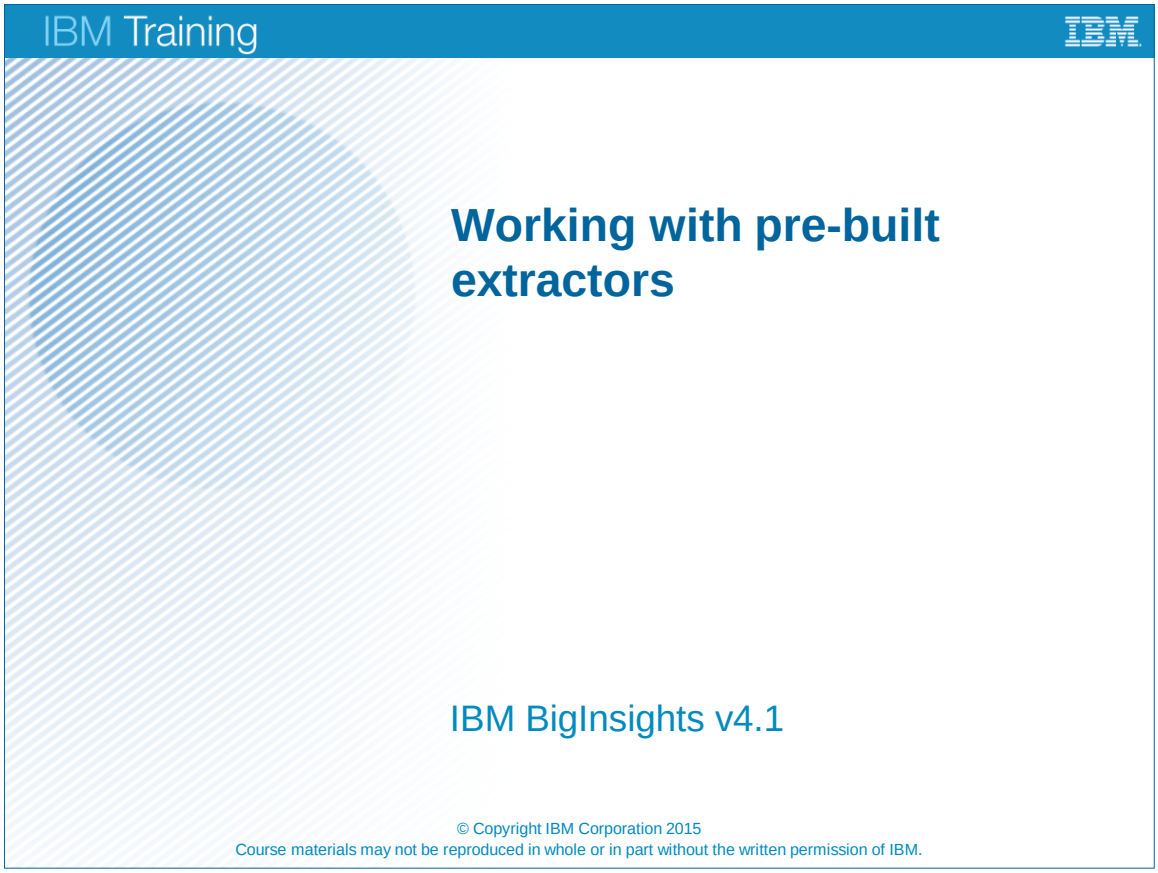
Results:
You have learned to filter and consolidate the results of the extractors.

IBM Training

Unit summary

- Run an extractor and refine the results
- Eliminate duplicates and overlaps
- Filter the results
- Export the results

Unit summary



Working with pre-built extractors

IBM Training

Unit objectives

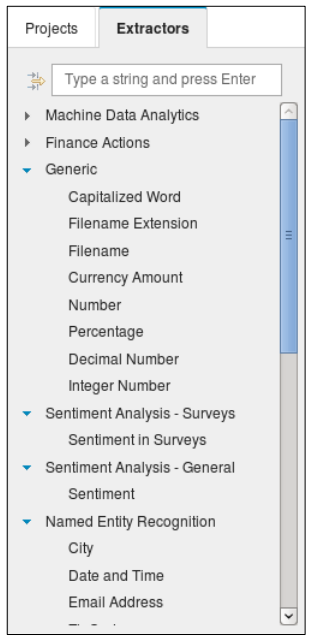
- List the categories of the pre-built extractors
- Export extractors
- Describe tokenization and multilingual support

Unit objectives

IBM Training

Pre-built extractors

- Common extractors for in various domains
- Extract specific information from input text
- Located on the Extractor catalog
- Five general categories
  - Named entity extractors
  - Financial extractors
  - Generic extractors
  - Sentiment extractors
  - Other extractors



Working with pre-built extractors

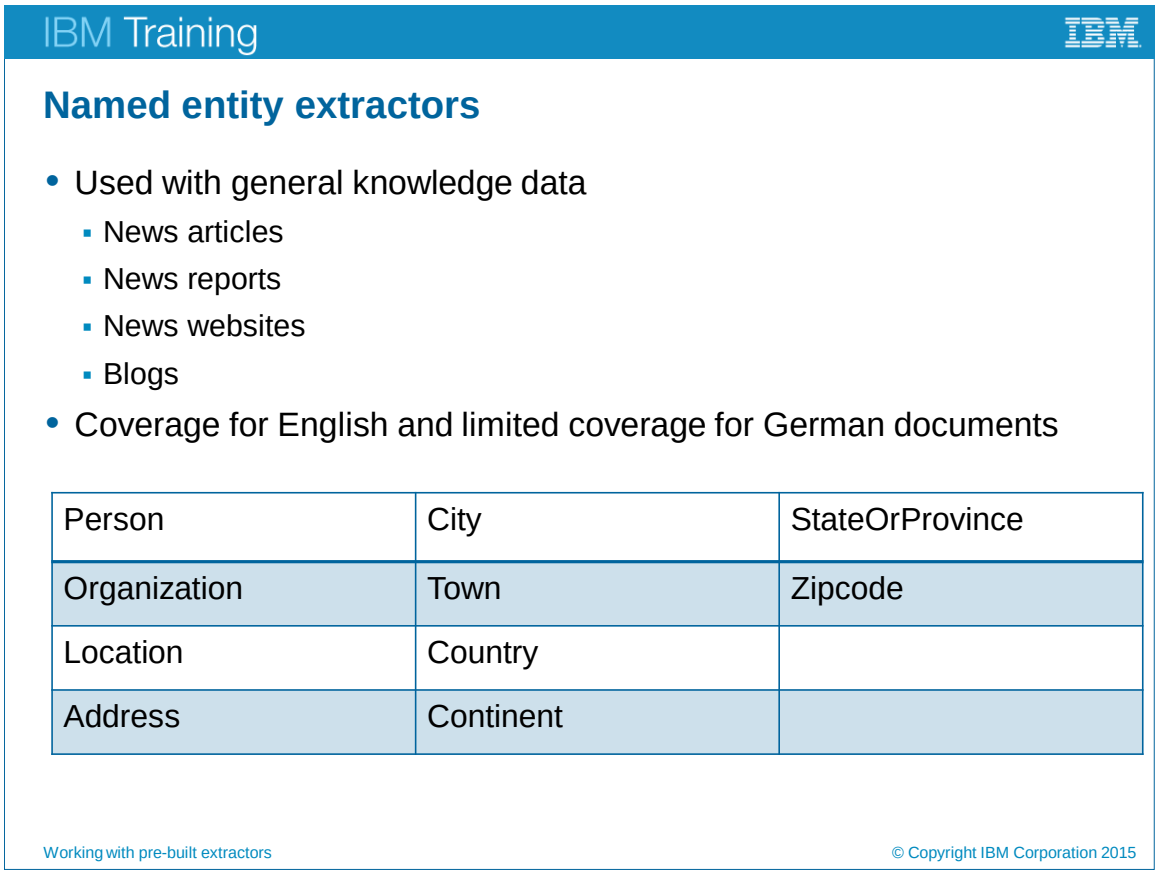
Pre-built extractors

BigInsights Text Analytics comes with a catalog of pre-built extractors that are ready to use. All you do is drag and drop them from the **Extractor Catalog** onto the canvas and specify the Extractor properties as needed. Then they are ready to go. These are some of the common extractors that are created for various domains and you can use them without having to worry too much about how they are created.

There are five general categories of pre-built extractors.

- Named entity extractors
- Financial extractors
- Generic extractors
- Sentiment extractors
- Other extractors

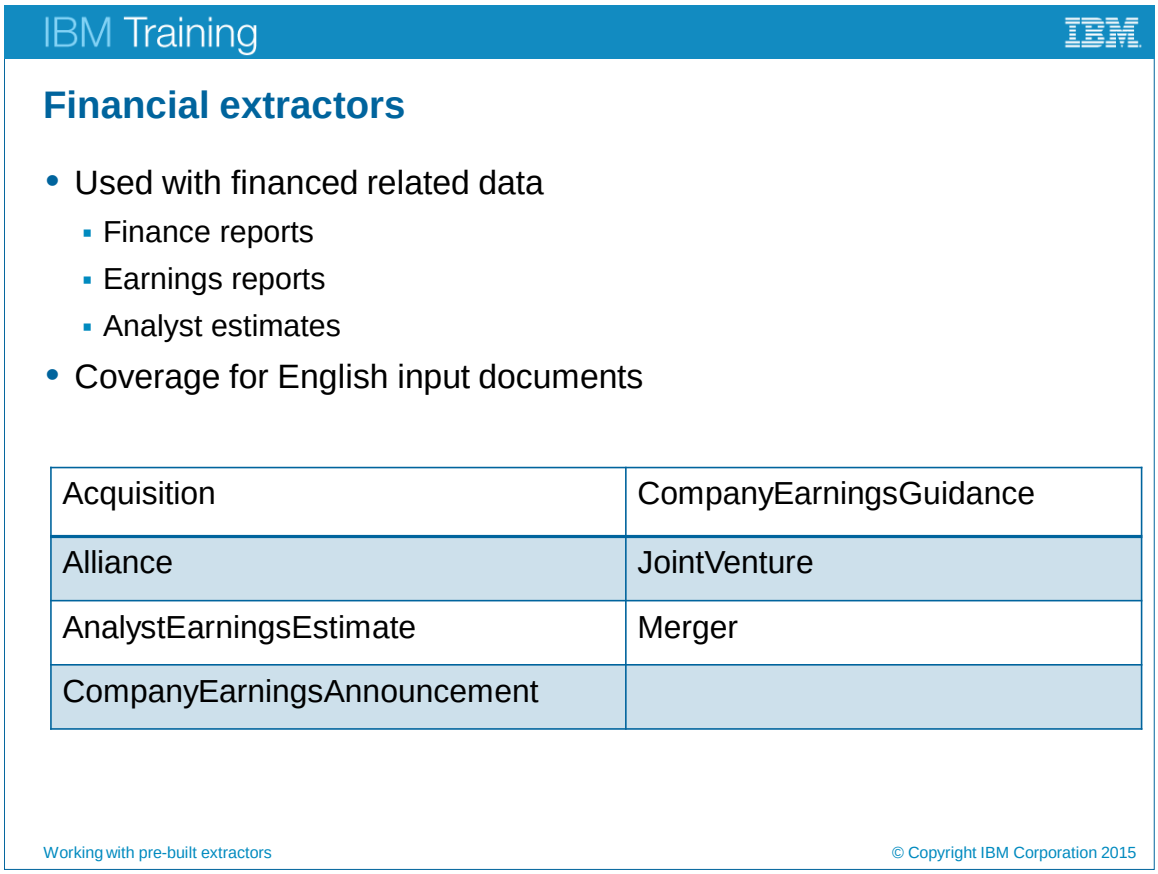
More details on these extractor categories on the following slides.



Named entity extractors

Named entity extractors are used with general knowledge data such as news articles, reports, websites or blogs. There is coverage for English and limited coverage for German documents using these extractors. Listed here are the actual extractors themselves.

- Person
- Organization
- Location
- Address
- City
- Town
- Country
- Continent
- StateOrProvince
- ZipCode

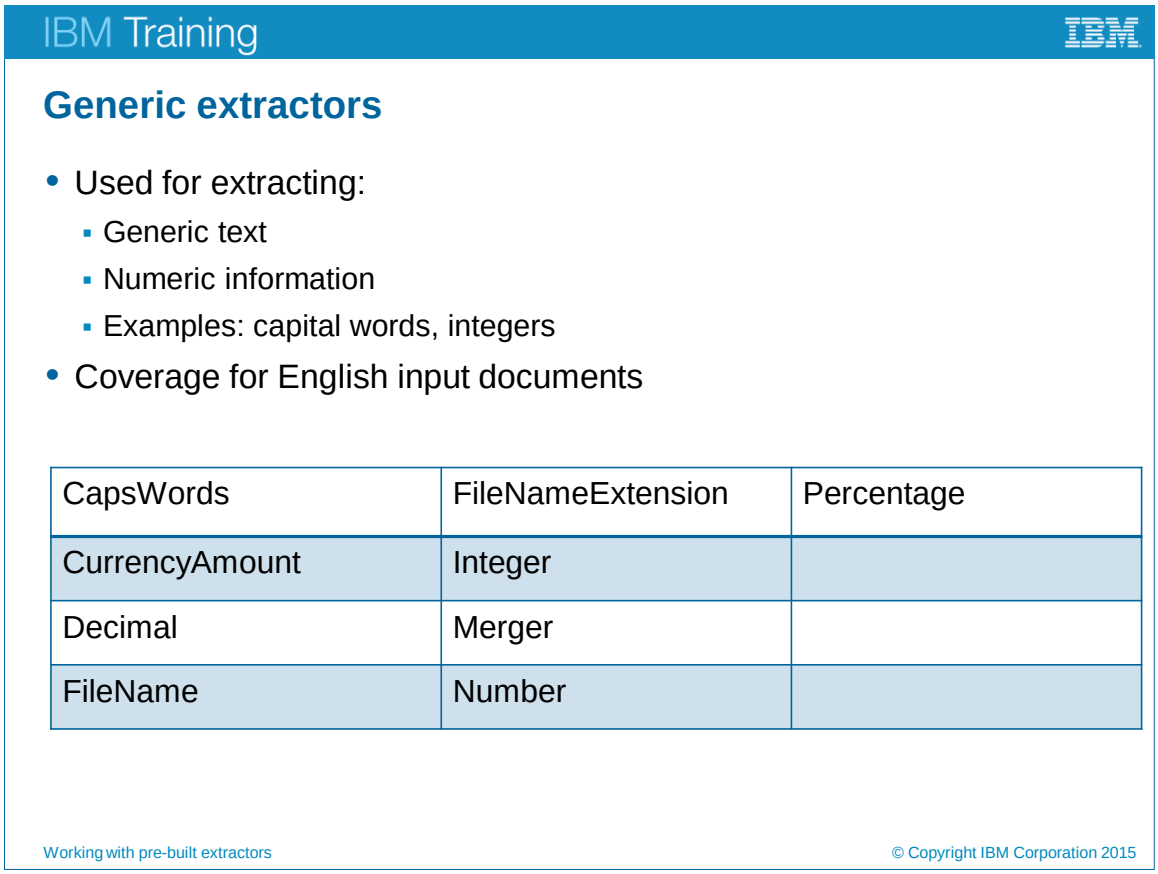


Financial extractors

Financial extractors, as you can guess, are used for financial related data including finance reports, earning reports, or analyst estimates. There is only coverage for the English input document. These extractors are:

- Acquisition
- Alliance
- AnalystEarningsEstimate
- CompanyEarningsAnnouncement
- CompanyEarningsGuidiance
- JointVenture
- Merger

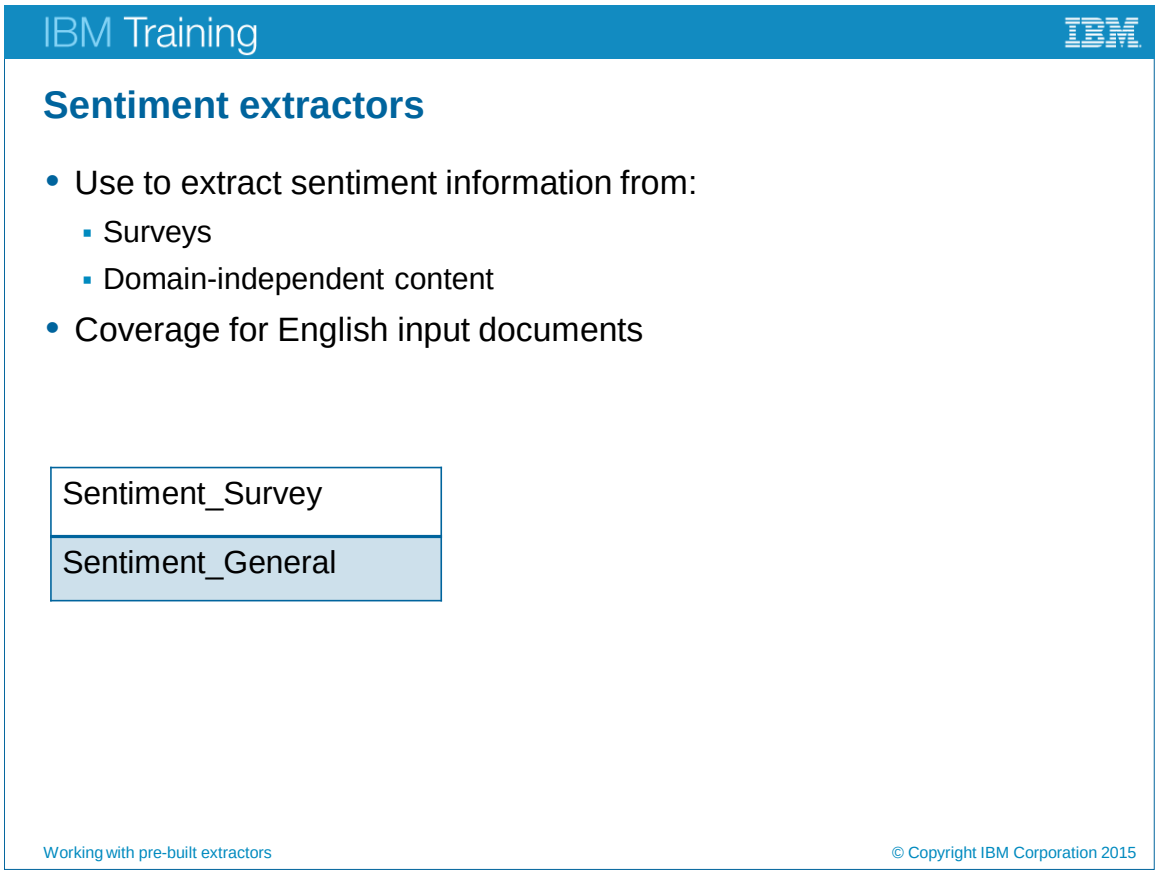
You can find more details on these extractors in the IBM Knowledge Center under BigInsights v4.1



Generic extractors

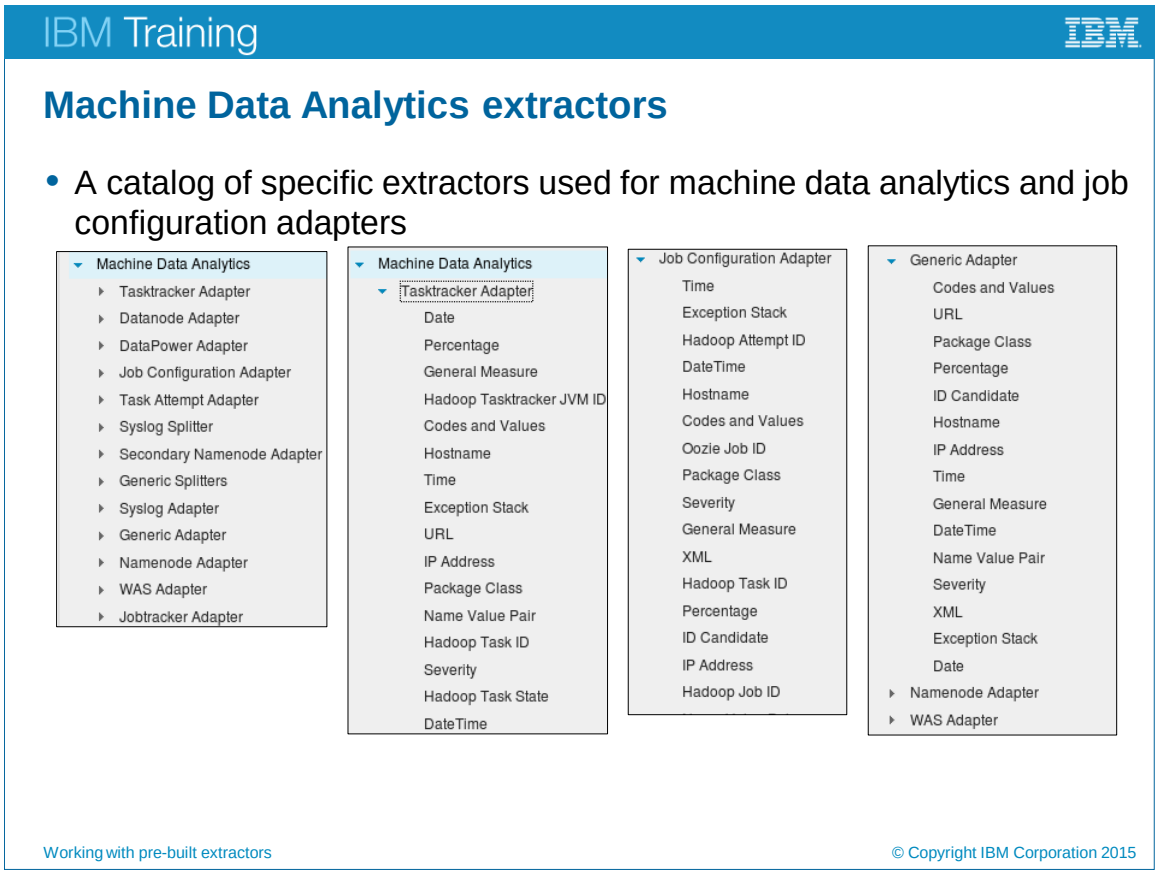
Generic extractors are used for extract text and numeric information such as capital words or integers. Again, there is only coverage for English input documents here.

- CapsWords
- CurrencyAmount
- Decimal
- FileName
- FileNameExtension
- Integer
- Merger
- Number
- Percentage



Sentiment extractors

Sentiment extractors are used to extract sentiment from surveys or domain-independent content. Only coverage for English input documents is available.



Machine Data Analytics extractors

This category contains a catalog of specific extractors used for machine data analytics and job configuration adapters. If you need to find information such as extracting http status codes and their values from web pages or analyzing exception stacks. There's a ton of different extractors under this category. I recommend you go to the Knowledge Center for more information on them.

IBM Training

Other extractors

- Used to extract domain independent information:
  - Dates
  - URLs
  - Emails
- Coverage for English input document

DateTime	URL
EmailAddress	
NotesEmailAddress	
PhoneNumber	

Working with pre-built extractors

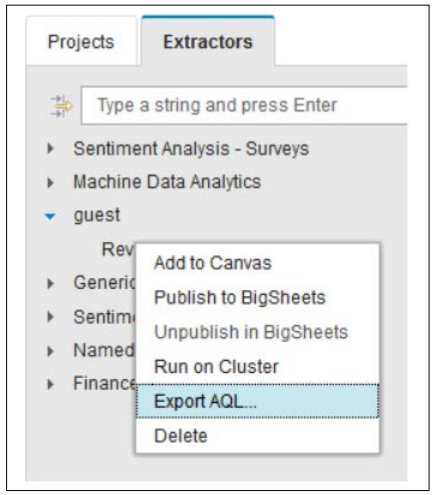
Other extractors

There are other extractors that you can use to extract domain independent information such as dates, URLs, emails. Only coverage for English input document.

IBM Training

Exporting extractors

- Export extractor for use by external applications
  - Exporting to AQL
  - Exporting as MapReduce jobs
  - Exporting as BigSheets functions



Working with pre-built extractors

Exporting extractors

AQL

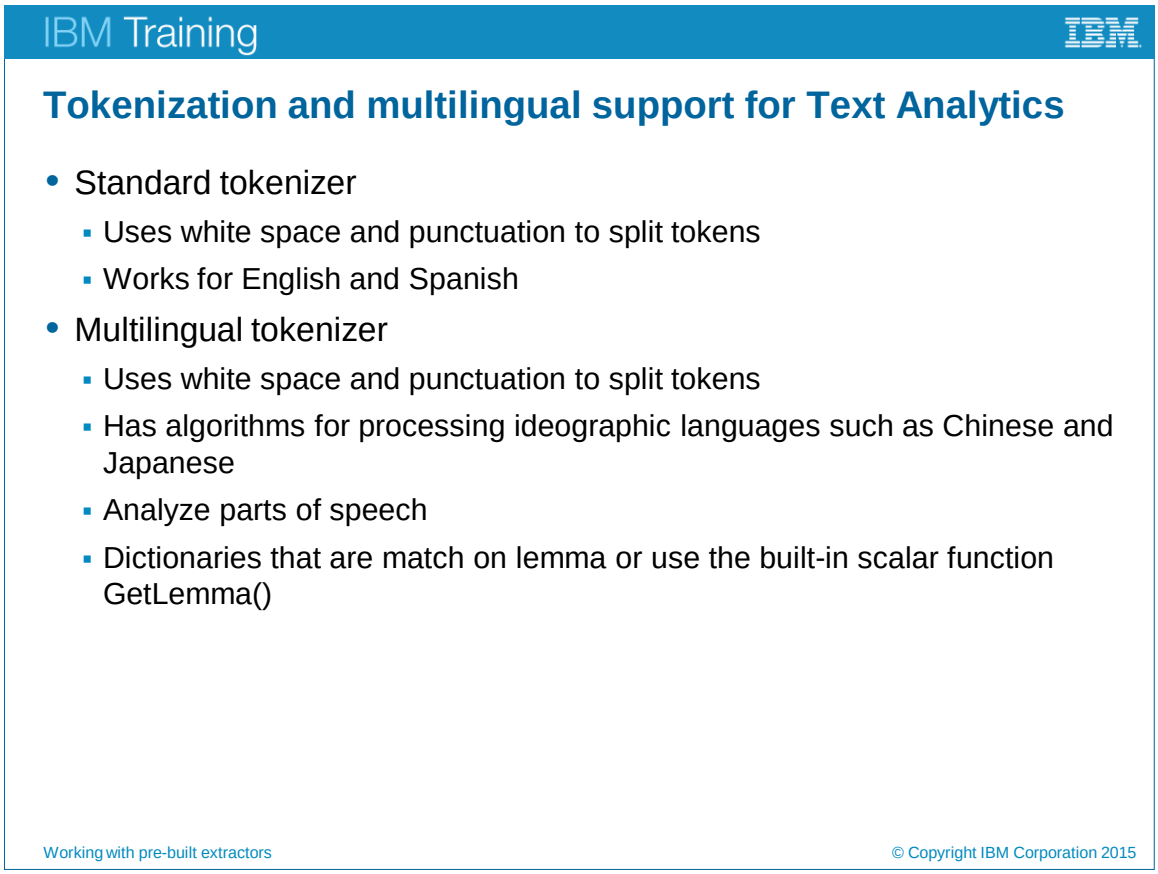
When you execute extractors in the web tool, they are transformed into Annotated Query Language (AQL) statements which are in turn then compiled and executed against your sample documents. If you want to view, edit, or execute the generated AQL outside of the web tool, you can use the Export AQL... option to obtain the AQL for your extractors.

MapReduce jobs

When the web tool is running in an environment that has access to a distributed file system (DFS), you can use the Run on Cluster option to export your extractors as map/reduce jobs.

BigSheets functions

When the web tool is running in an environment that has access to the BigSheets value-add service, you can use the Publish to BigSheets option to export your extractors as BigSheets functions. Published BigSheets functions can be executed from within the BigSheets web application, just like any other BigSheets function.



Tokenization and multilingual support for Text Analytics

Standard tokenizer

The Standard tokenizer uses white space and punctuation to split tokens. Since a white space tokenizer is efficient, you can use this tokenizer for alphabetic languages like English and Spanish. The Text Analytics runtime component uses the Standard tokenizer by default. BigInsights Text Analytics does not support part-of-speech extraction when tokenization is performed by using the Standard tokenizer. It also does not support dictionaries that are matched on lemma, and the GetLemma() scalar function.

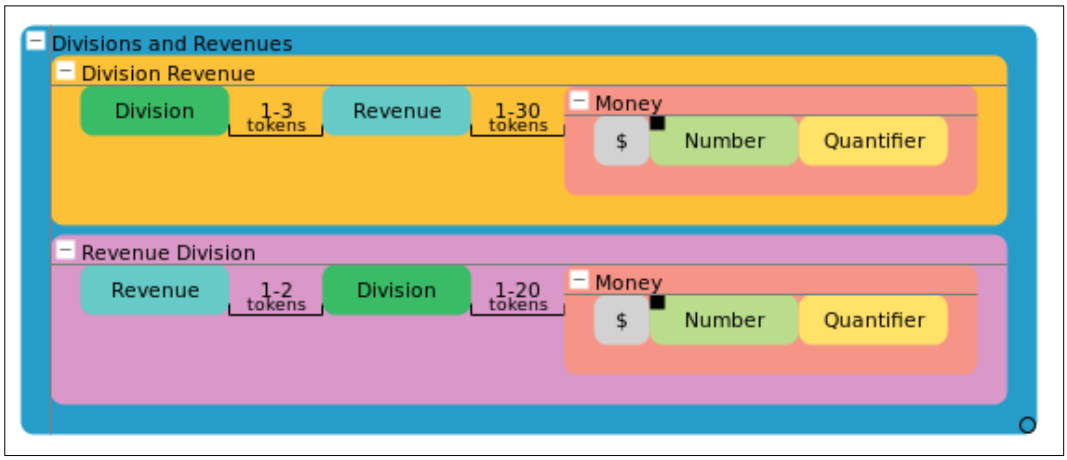
Multilingual tokenizer

The Multilingual tokenizer uses white space and punctuation to split tokens, but also has algorithms for processing ideographic languages such as Chinese and Japanese. Use the Multilingual tokenizer to process Chinese or Japanese text or to analyze parts of speech, or if you have dictionaries that are matched on lemma, or use the built-in scalar function GetLemma().

IBM Training

Demonstration 1

Analyzing quarterly reports using Text Analytics



Working with pre-built extractors

Demonstration 1: Analyzing quarterly reports using Text Analytics

Demonstration 1: Analyzing quarterly reports using Text Analytics

Purpose:
In this demo, you will work with some of the prebuilt extractors to analyze quarterly reports.

User ids / Passwords

OS:	**biadmin/biadmin**
Root:	**root/dalvm3**
Ambari:	**admin/admin**
BigInsights Home:	**guest/guest-password**

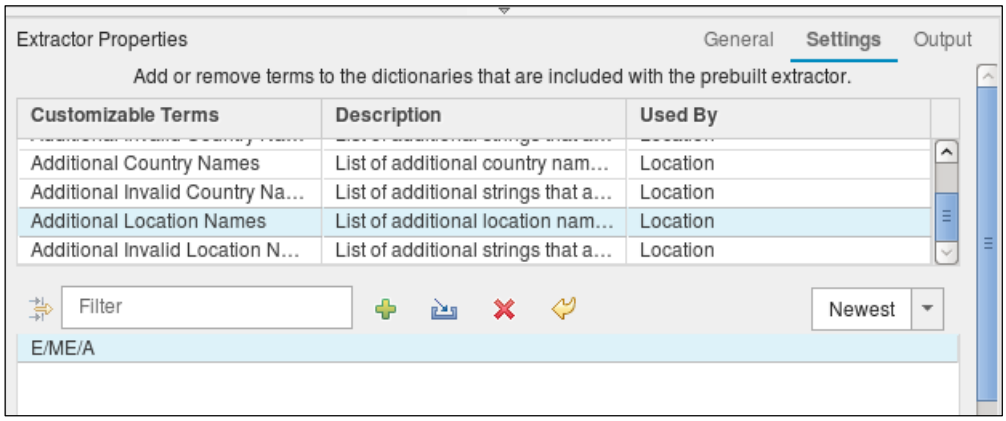
Ambari Services Required:

- HDFS
- MapReduce2
- YARN
- Knox (also start the Demo LDAP service)
- BigInsights - Text Analytics
- BigInsights - Home

Task 1. Setting up your project.

1. Create a new project called **Quarterly Reports**.
2. Add the five documents to the project from this location: **/home/biadmin/labfiles/TextAnalytics/IBMQuarterlyReports**
3. On the **Extractor** tab, expand the **Named Entity Recognition** category.
4. Drag and drop the **Location** extractor onto the canvas.
5. Run the extractor on the documents and you should get 63 rows returned.
6. Scroll through the results until you see the first result for the document listed for *4Q2007.txt*. Double click on that entry to see the results in the document pane. Notice that the acronym E/ME/A is not highlighted. This is an acronym that IBM uses for Europe/Middle East/Africa. You can modify the prebuilt extractor so that it accurately captures what you need.

7. On the **Extractor Properties** under the **Settings** tab, select the **Additional Location Names** and add the **E/ME/A** acronym to that list.



8. Run the extractor again and you see that there are now 64 matches.

Task 2. Analyzing documents and identifying examples.

For this process, you would typically enlist the help of someone who knows the document well to help you identify the examples of clues to search. Since you are interested in extracting revenue by division, you must read through to find spans of text that contain this information. Look for patterns and clues in the text to help improve the accuracy of the extractor.

An example that you might find is a phrase such as *Revenues from Software were $3.9 billion*. This has three important features:

"Software" is a division name

"$3.9 billion" is a revenue amount

"revenue"

You will use these features as context to identify instances of revenue by division.

It is a good idea to decompose the clues to the lowest level. This allows for flexibility and also it lets the extractor performs all the hard work of combining all the clues. Consider that *Money* has three basic features, a currency sign, followed by a number, followed by a quantifier such as million or billion.

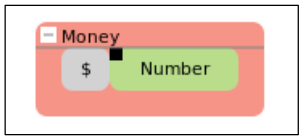
Two patterns that you may have picked up are:

Revenues for division were $x.x

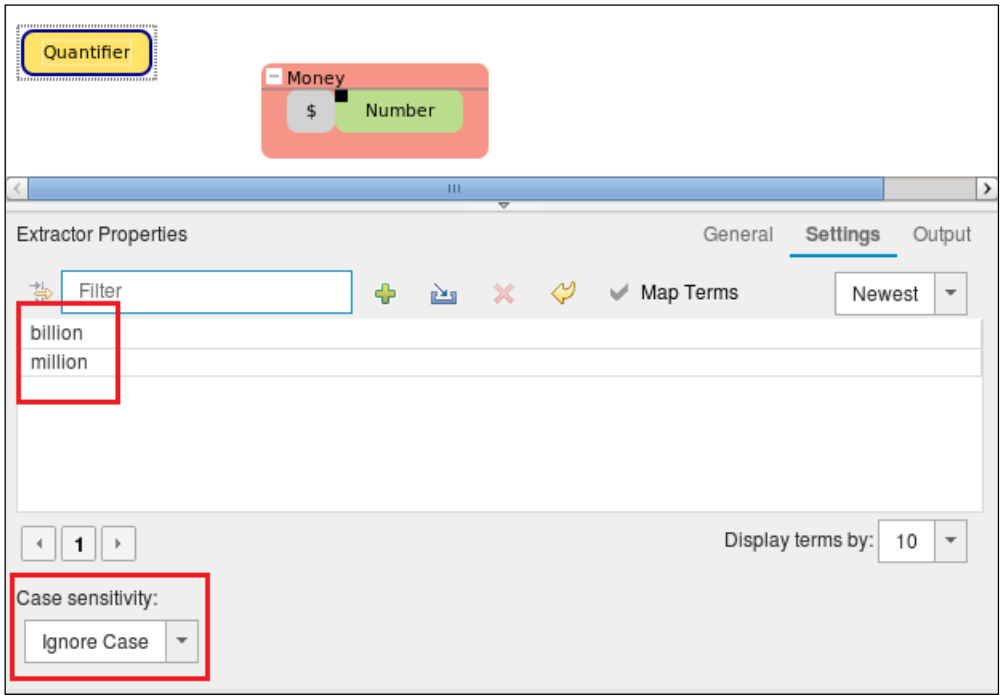
Division revenues were $x.x

Task 3. Creating and testing extractors.

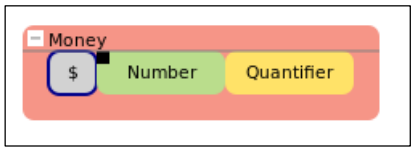
1. You will create the basic features on the canvas. Create one feature for each of the three basic features of money. Click the **New Literal** button. Type a dollar sign ($).
2. Run that extractor and you should see 389 results.
3. On the **Extractor**s catalog, expand the **Generic** category and drag the **Number** extractor to the canvas.
4. Run that and you will see 1722 results.
5. Create a sequence of these two by dragging the literal to the left of the Number extractor.
6. Rename it from **Sequence 1** to **Money**.



7. Run the Money extractor to test it out.
8. To add the Quantifier extractor, you will use a dictionary. Click the **New Dictionary** button. Name it **Quantifier.**
9. Enter in two terms for the dictionary: **million** and **billion**.
10. Ensure that the case sensitivity is set to **Ignore case** under the **Settings** tab.



11. Drag the **Quantifier** extractor into the **Money** extractor to complete the sequence.

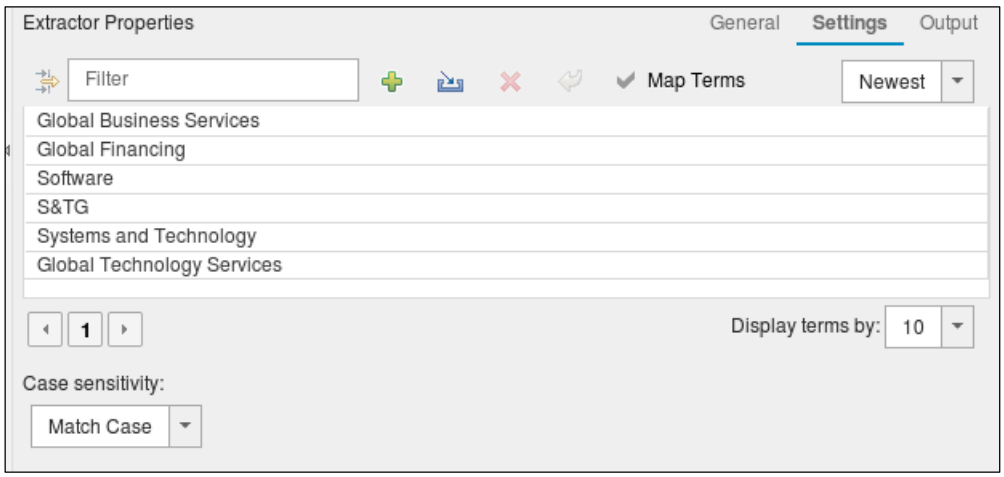


12. Run it and you should see 333 matches returned. You'll also see each tab for the rows specific to the individual extractors.

    You have now located all the instances of money. Next task is for you to combine with revenues and divisions.

Task 4. Writing and testing extractors for candidates.

1. Create a new dictionary. Name it **Revenue.**
2. Add two terms to it: **revenue** and **revenues**.
3. Run this extractor to test it out. There should be 238 matches.
4. Create a new dictionary for **Division** names.
5. Add the following terms to it: **Global Technology Services, Systems and Technology, S&TG, Software, Global Financing,** and **Global Business Services.**
6. Run the extractor. You should get 142 rows.
7. Notice that in the results, the terms software and global financing are picked up as division names. Because they are in lowercase, they are likely not division names. The problem can be fixed by choosing the **Match Case** option for the extractor.



8. Run the extractor again and you should get 98 rows now.

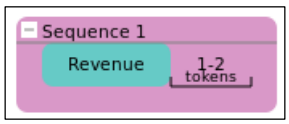
You have now extracted the three key basic features: money, revenue, and division. The next step is to extract candidates that match the two patterns that you identified earlier. To generate candidates, you combine extractors into sequences, building on the extractors that you created in the previous tasks.

If you remember, the two patterns are: revenues for division were $x.x and division revenues were $x.x

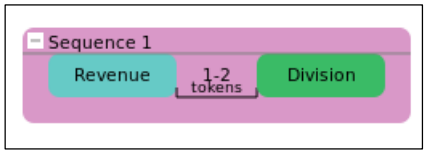
The first pattern looks for examples where the word revenue is followed by a division name and then a money amount, with some number of tokens in between. This is the conceptual view of the first pattern.

<Revenue><1 to 2 Tokens><Division><1 to 20 Tokens><Money>

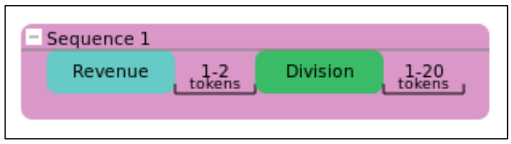
9. Add a new proximity rule with 1-2 tokens.
10. Drag the proximity rule to the right of the Revenue extractor to create a new sequence.



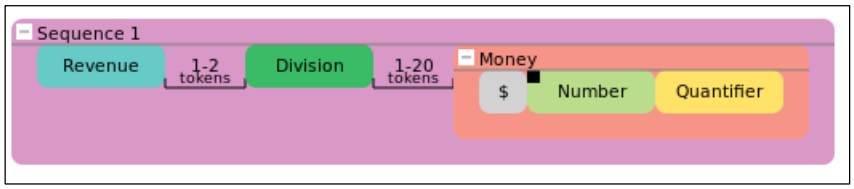
11. Drag the Division extractor to the right of the proximity rule to add to the sequence.



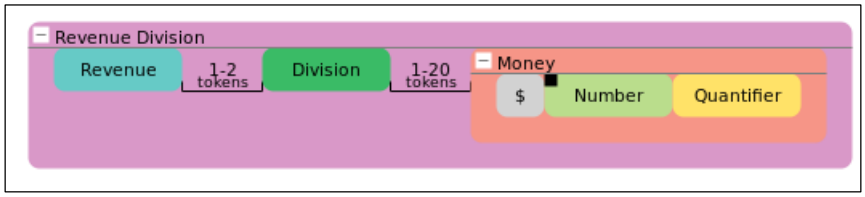
12. In order to create the next proximity rule, right-click on the **Division** extractor and choose **Add After** and then **Proximity Rule**. Fill in 1-20 in the text box.



13. Drag the **Money** extractor into the sequence.



14. Rename **Sequence 1** to **Revenue Division**.

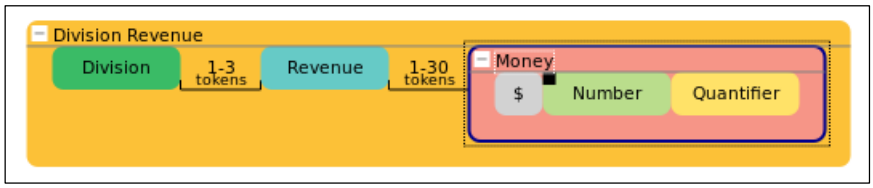


15. Run it and you should see 22 results.
16. For the second pattern, it looks for examples where a division name is followed by the word revenue and a money amount. For example, *Global Financing segment revenues* increased *3 percent* (flat, adjusting for currency) in the fourth quarter to *$620 million*. When you look at these patterns in the document, you find that there are between 1 and 3 words between *Division* and *Revenue*, but perhaps as many as 30 between *Revenue* and *Money*.

    Conceptually, this looks like:

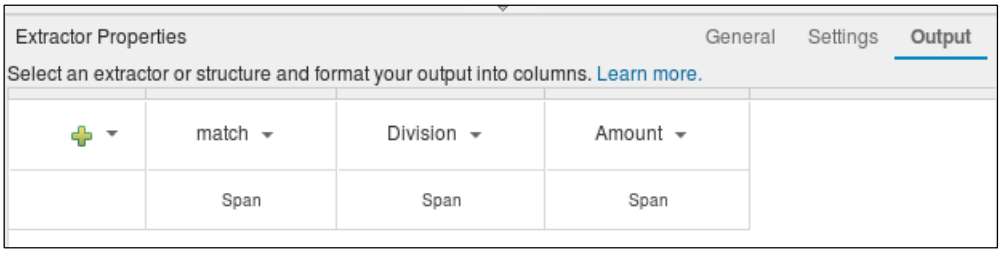
    `<Division><1 to 3 Tokens><Revenue><1 to 30 Tokens><Money>`
17. Right-click on the **Revenue** extractor and choose **Copy.**
18. Right-click on the canvas and choose **Paste.**
19. Do the same for the **Division** and **Money** extractors.

    You should now have linked copies of the three extractors. Remember, linked copies are affected when you change one. If you needed a new copy, you would select Paste as New Copy. Notice that the linked copies are the same color.
20. Create a new sequence with these three extractors and proximity rules to create an extractor for the second pattern. Name it **Division Revenue**.

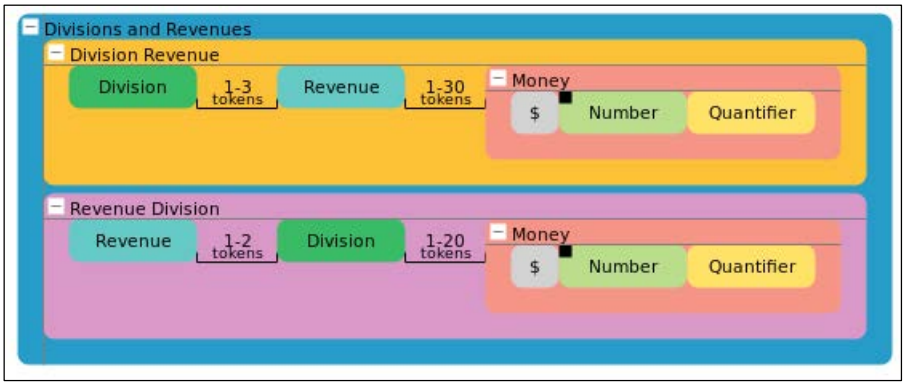


21. Run the extractor. You should get 40 matches.

22. Union these two extractors together to yield a full picture. First, the columns must match. Modify the output specification to make them match.
    a) Go to the Output tab of the Properties pane.
    b) Deselect the **Revenue** column. We don't need this column. Click on the dropdown next to the green plus.
    c) Rename the **Division Revenue** column to **match**.
    d) Rename the **Money** column to **Amount**.



23. Both extractor's output column must match.
24. Create the union by dragging and dropping one of them on top of the other.
25. Change the name of the union to **Divisions and Revenues.**



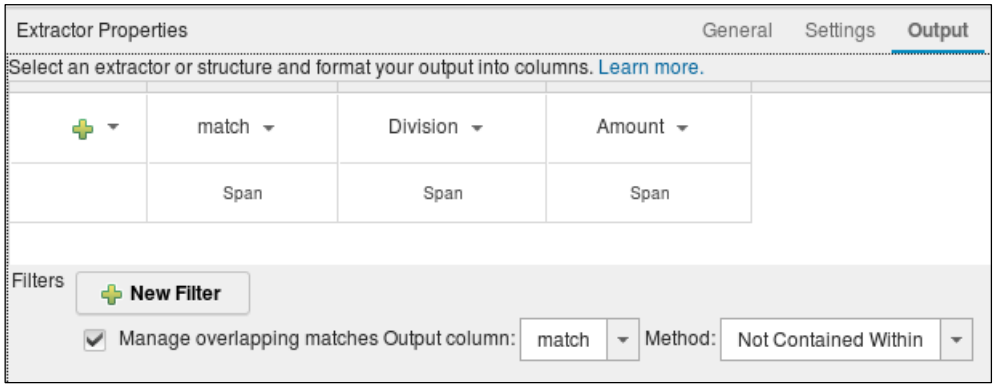
26. Run the extractor and you should get 62 matches.

Task 5. Creating and testing final extractors.

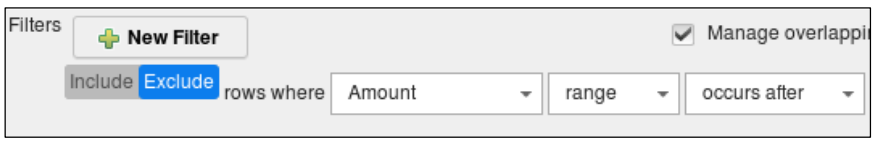
The first part of consolidating is to remove duplicate information. As you scroll through the 62 results, notice the last two entries for *4Q2006.txt* that one of the results is contained within the other results, causing duplicated matches.

1. Right-click the union extractor and select **Edit Output**. This will bring you to the Output tab of the Extractor Properties pane.
2. Click **Manage overlapping matches**.

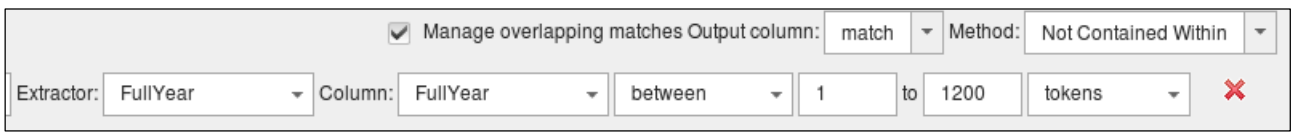
3. Choose output column **match** and Method **Not Contained Within**. This specifies that we only want matches which are not contained within another match.



4. Run the extractor again and verify that none of the 49 matches are contained within another.
5. Look at the results again. Notice that there are two values for the Software division in *the 4Q2006.txt* file. Looking more closely, one of these results was for 4Q, and the other for the full year.
6. On examining the document, we see that the unwanted results have their Money amount within a proximity of 1200 tokens from a phrase like `Full-Year 2006 Results`. To match multiple years, we can create a regular expression to match this clue for unwanted results. Click the **New Regular Expression** button.
7. Type in **FullYear**
8. Type in **Full-Year \d{4} Results** as the regular expression.
9. Run the regular expression to test it. You should see five results, one per document.
10. Select the **Divisions and Revenues** extractor. Under the Output tab, click the **New Filter** button.
11. Click **Exclude,** because we want to exclude some rows.
12. Choose the **Amount** column, the **range** type, and the **occurs after** option.



13. Select the **FullYear** extractor and **FullYear** Column choose **between** and fill in **1** and **1200** for the **tokens**.



14. Run the extractor and verify that only 25 results should be shown now.

    This view contains exactly the information that you need for further analysis. When you apply text analytics to more complex documents, and when you are extracting more sophisticated information, you would expect to spend time improving the precision and recall of your extractor. You can also profile your extractor to understand and improve its performance characteristics.

Task 6. Finalizing and saving the extractor.

1. Click on the extractor and click the **Save** icon.
2. Select the **guest** category to save the extractor.
3. You can choose to **Export AQL**, **Publish to BigSheets** or **Run on the Cluster**.

Results:
In this demonstration you have learned to use some of the pre-built extractors to analyze IBM quarterly reports to figure out the revenues from each division.

IBM Training

Unit summary

- List the categories of the pre-built extractors
- Export extractors
- Describe tokenization and multilingual support

Unit summary

IBM Training