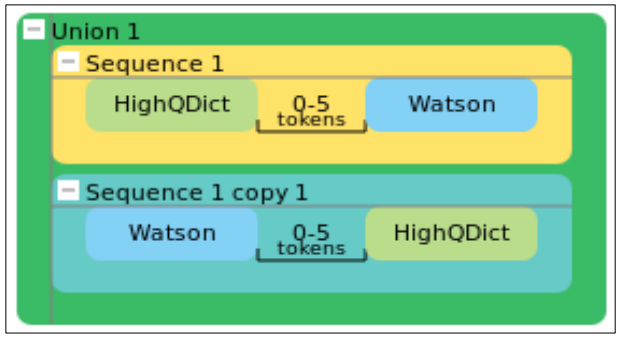
IBM Training

## 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.

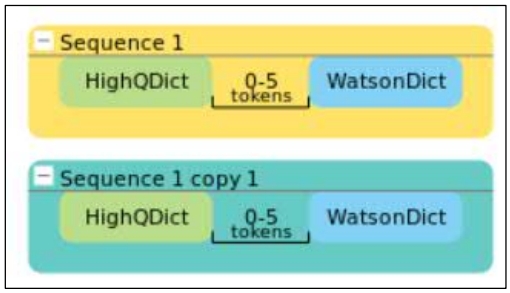
   This proximity rule will allow us to look for terms that are within five tokens of the words around it.
3. Join the **HighQDict** and the **WatsonDict** extractor with the proximity rule in the middle. Drag the proximity rule to the right of the **HighQDict**. You will see a blue line indicating that 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.

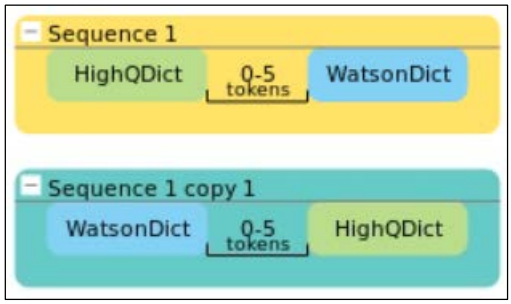
## Task 3. Making a copy of an extractor.

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.



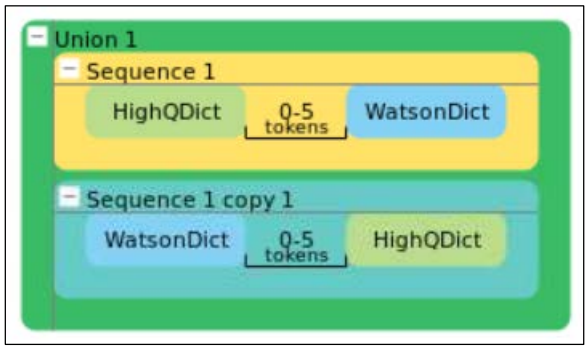

The **Paste as New Copy** makes a separate copy of the original extractor. Changes that you make to original extractor will NOT affect the copy.

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.*

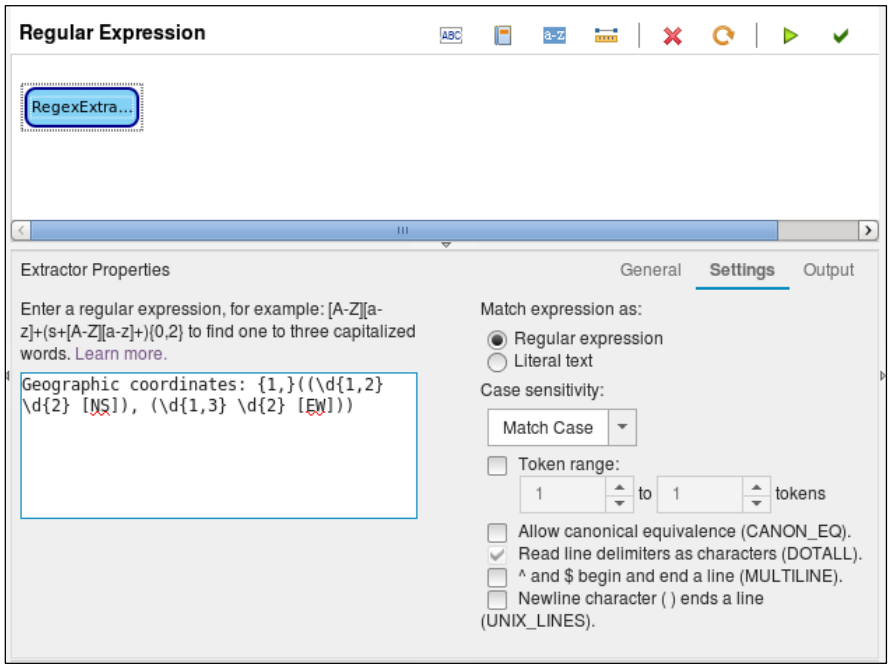
   A few lines further down, you should see

   *Geographic coordinates: 33 00 N, 65 00 E*

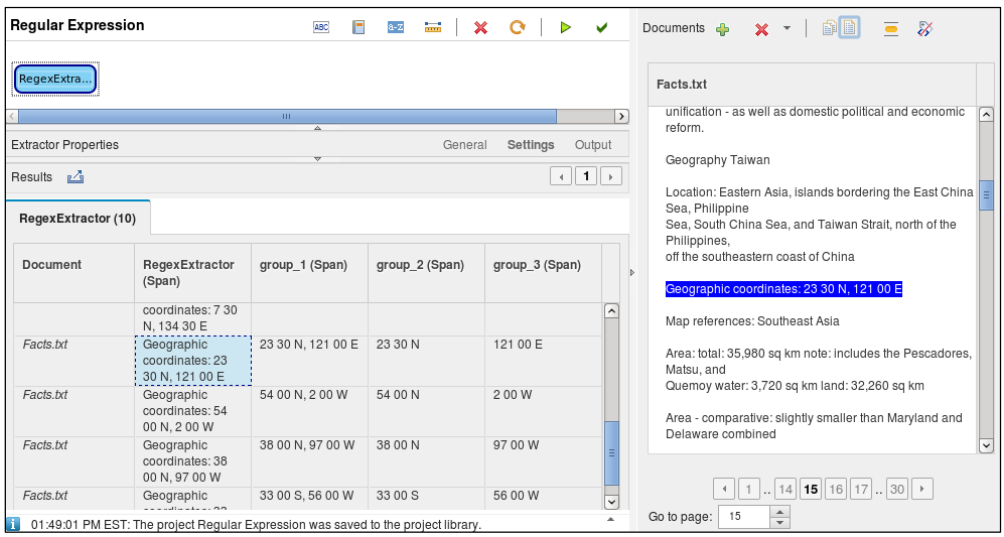
   You are going to create and use a regular expression extractor to extract the geographic coordinates.
7. Click the **New Regular Expression** button.
8. Name this extractor, **RegexExtractor.**

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.**