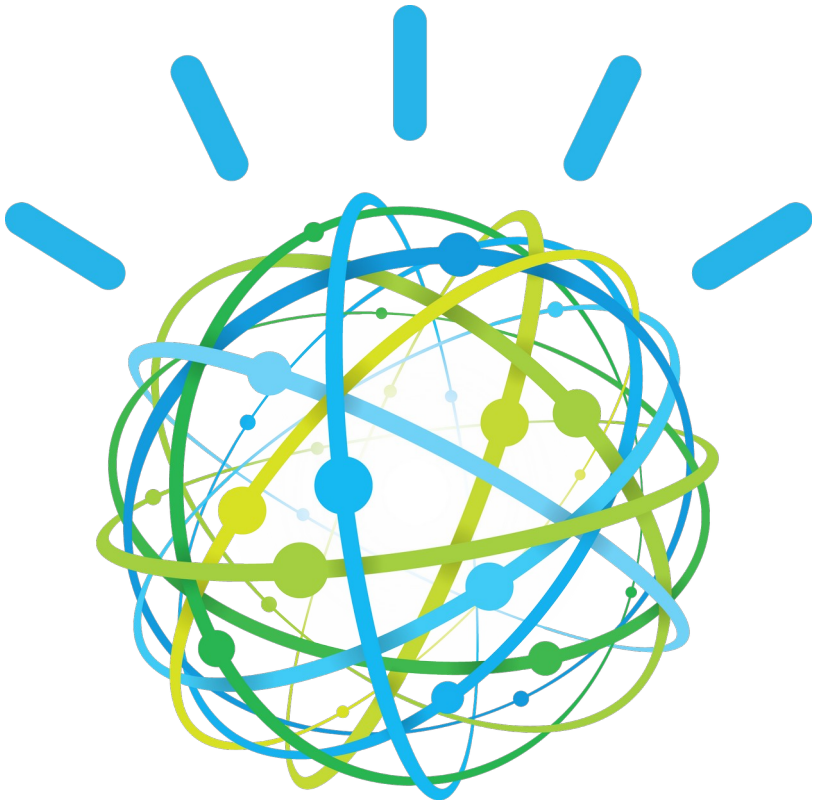
# IBM Watson Knowledge Studio

# Building a Machine-Learning Annotator

IBM Global Business Partners
Duration: 45 minutes
Updated: May 5, 2020
Klaus-Peter Schlotter
kps@de.ibm.com



Version 1.0

# Overview

The IBM Watson Developer Cloud offers a variety of services for developing cognitive applications. Each Watson service provides a Representational State Transfer (REST) Application Programming Interface (API) for interacting with the service. Some services, such as the Speech to Text service, provide additional interfaces.

With Watson Knowledge Studio you can teach IBM Watson® the language of your domain with custom models that identify entities and relationships unique to your industry in unstructured text. Build your models in a collaborative environment designed for both developers and domain experts, without needing to write code. Use the models in IBM Watson Discovery, IBM Watson Natural Language Understanding and IBM Watson Explorer.

The features that are available depend on the language. This list may change over time, so always check this in the online documentation.

The language model built in this tutorial will be applied Natural Language Understanding app created in Lab 4 that can run on your local workstation and be deployed on the IBM Cloud.

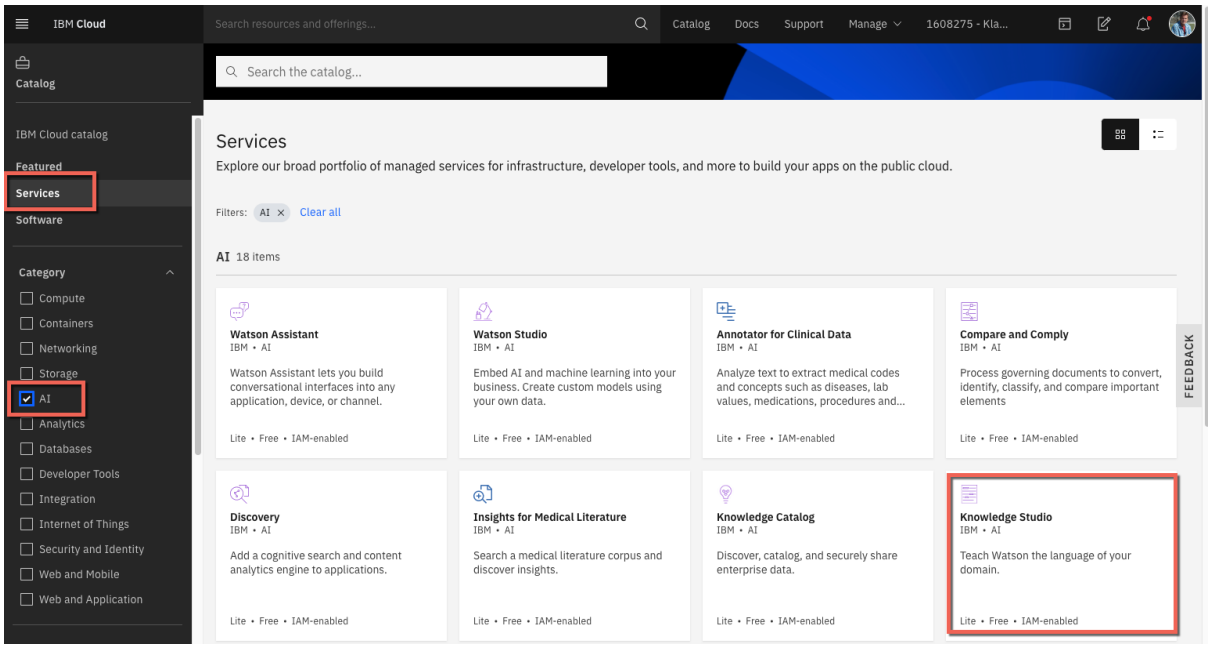
- In a first step, an instance of the Watson Knowledge Studio service will be created.
- In a second step, a Machine Learning annotator will be built and deployed to a Natural Language Understanding service in the IBM Cloud.
- Then the model will be tested with the Postman REST client.
- Finally the Node.js application created in Lab 4 will be used to incorporate the model into an application.

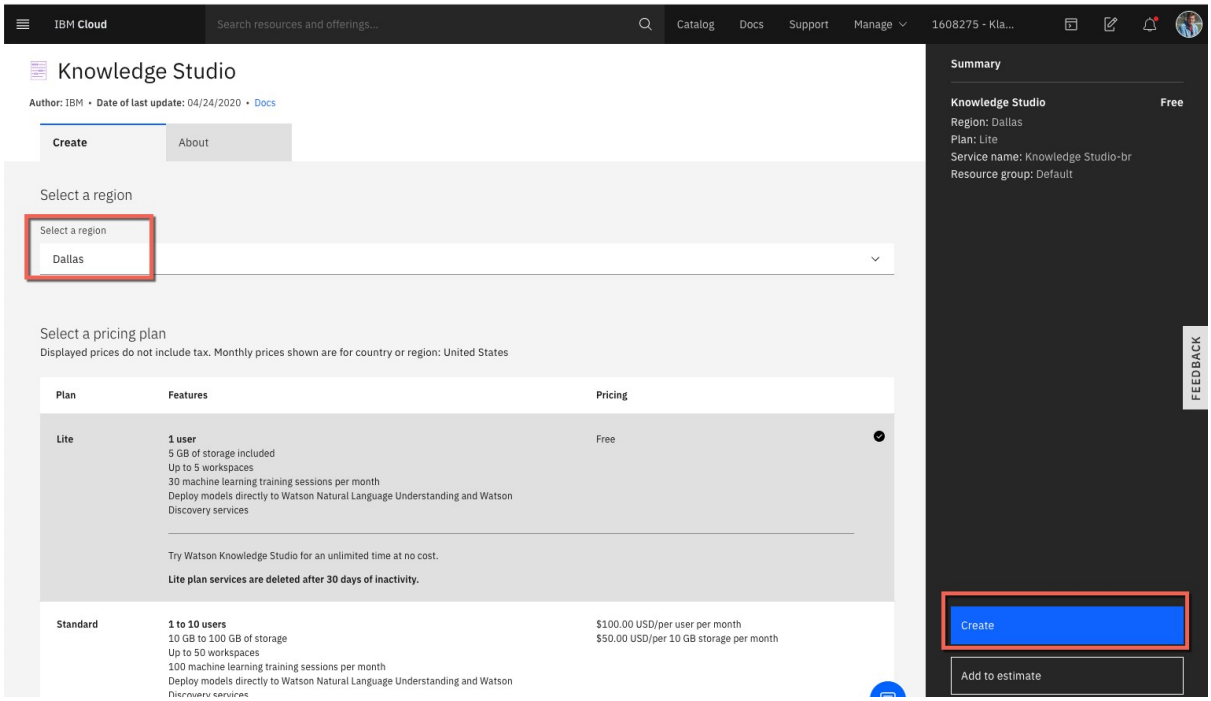
# Section 1: Create a Machine Learning Annotator

**Step 1** In your IBM Cloud Platform console **click** Create resource +

**Step 2** **Click** *Services* then **select** the *AI* category, **click** on the Knowledge Studio service

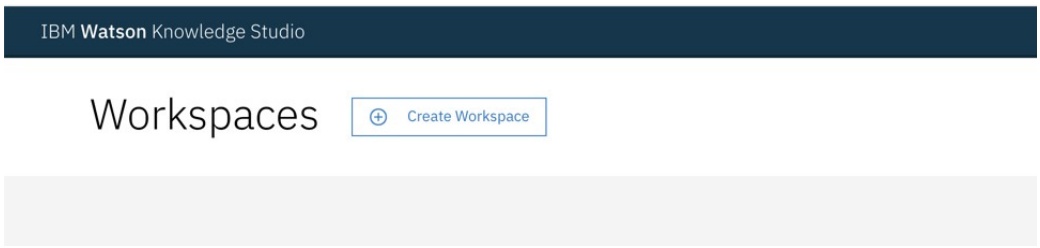



**Step 3** ***Select*** *a region*, **enter** a unique *name* (you can accept the default) and **select** the *Lite* plan. **Click** *Create*.

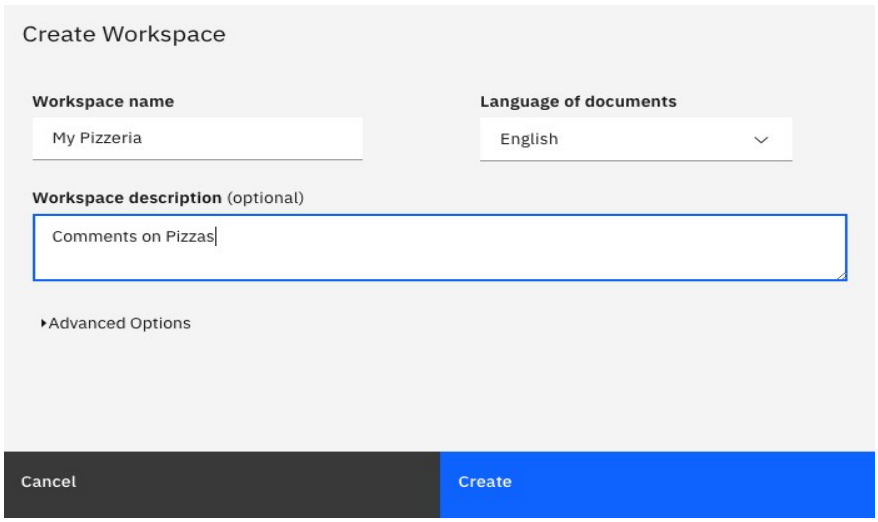


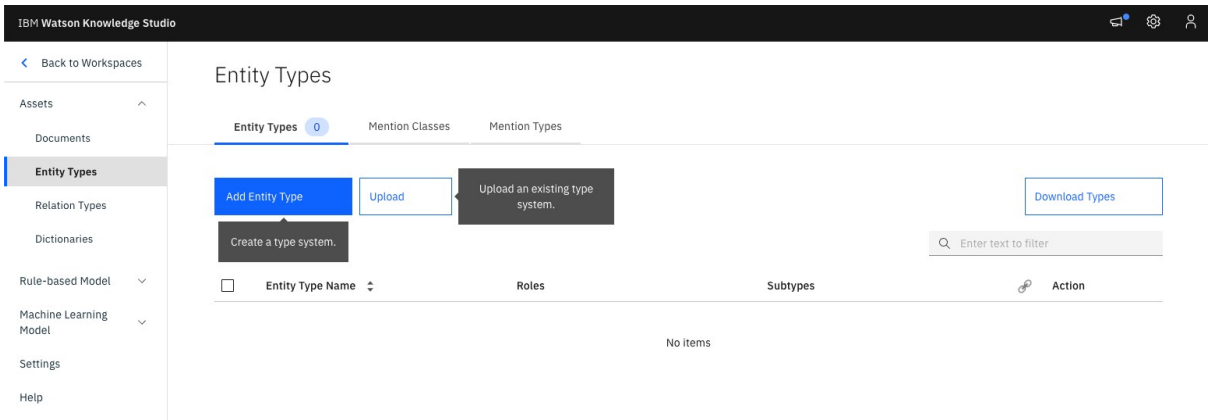
**Step 4** In the service instance **click** Launch Watson Knowledge Studio to open Knowledge Studio.




**Step 5** **Click** Create Workspace +, then Create entities and relations workspace . **Name** the workspace and **select** *English* as the language. **Click** *Create*.




**Step 6** The workspace opens with *Entity Types* page displayed.

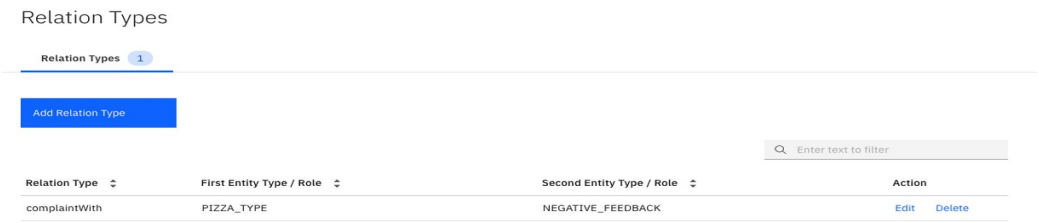


**Step 7** On the *Assets → Entity Types* click Add Entity Type. As *Entry Type Name* **enter** `PIZZA_TYPE` and **click** Save.

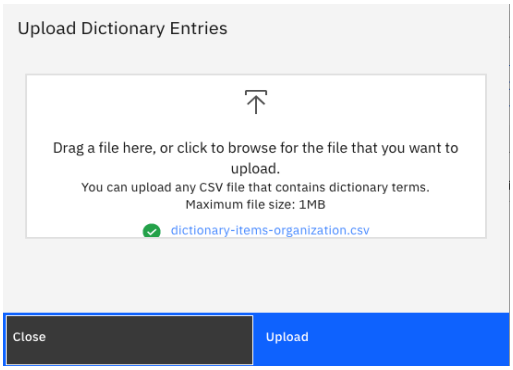
**Step 8** Also add an Entity Type `NEGATIVE_FEEDBACK`

**Step 9** On *Assets → Relation Types* **click** Add Relation Type.
**name it** `complaintWith`
**First Entity Type** `PIZZA_TYPE`
**Second Entity Type** `NEGATIVE_FEEDBACK`
**Click** Save.

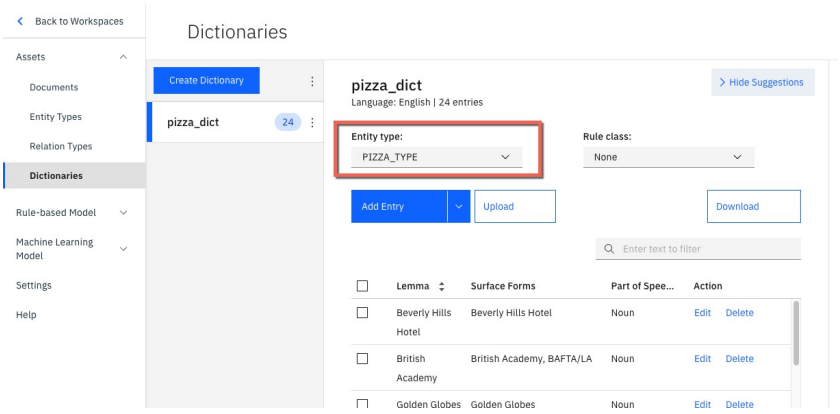



**Step 10** On *Assets → Dictionaries* **click** Create Dictionary,
**Name:** `pizza_types_dict`
**click** Save

**Step 11** Upload the dictionary entries for the *Pizza Types* (Dictionary-items-PizzaTypes.csv)




**Step 12** **Map** this dictionary to the *PIZZA_TYPE*.



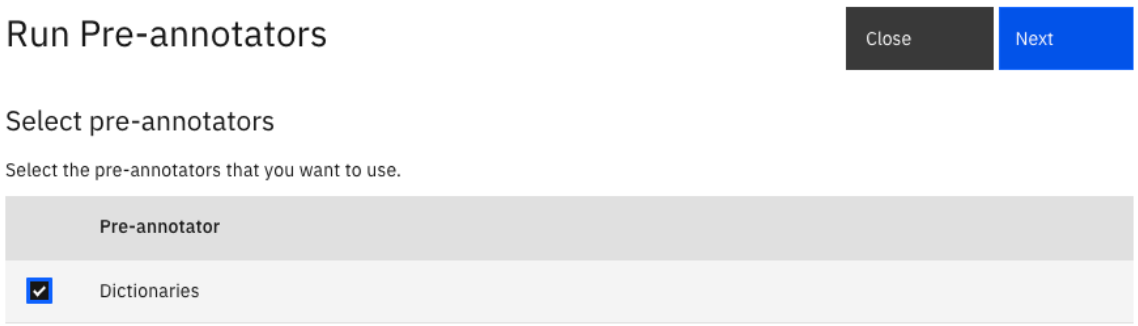
**Step 13** Click Add Entry, **enter** `four seasons` as *Surface Form* and **select** `Noun` as *Part of Speech*. Click Save.

> **Note:** For Brazilian Portuguese, English, French, German, Italian, and Spanish, Knowledge Studio does not currently provide an option to specify case-insensitive dictionary-matching, but dictionary entries match text that has a higher case. For example, vehicle in the dictionary matches vehicle, Vehicle or VEHICLE in text, while Sat in the dictionary matches Sat or SAT in text, but not sat.

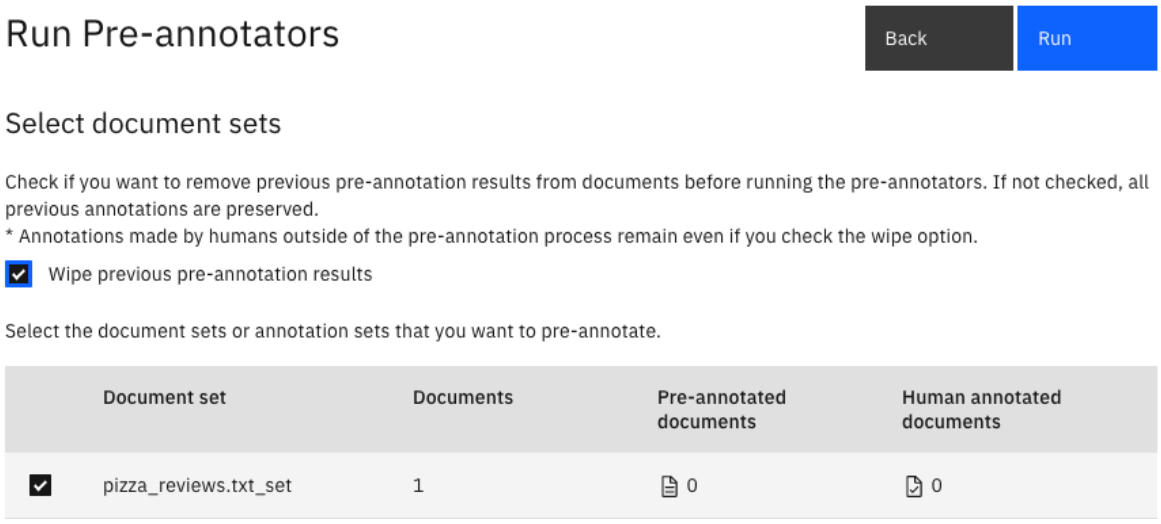
**Step 14** On Assets → Documents click Upload Document Sets to import the training documents (pizza_reviews.txt).

**Step 15** On *Machine Learning Model* → *Pre-annotation* you see the availabel pre-annotators - dictionary mapping done above is available.

**Click** Run Pre-annotators, **select** the one for dictionaries and click Next

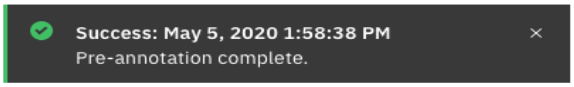



**Step 16** **Select** the documents set imported above and the option *Wipe previous ...* then click Run. **Click** OK on the *Confirmation* panel.



The Pre-annotation searches all occurrences of PIZZA_TYPE and marks them accordingly.
The following message should be displayed and the pizza types are annotated in our annotation set.



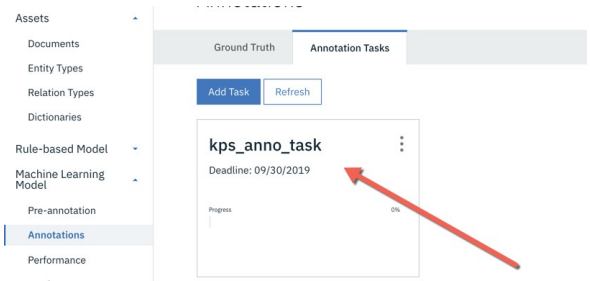

**Step 17** On *Machine Learning Model* → *Annotations* → *Annotations* **click** Add Task

> **Note:** You could do this directly on the Ground Truth tab when there is only one person to work on this annotation. But for multiple annotators an Annotation Task is needed to split the work.

**Step 18** On *Create Annotation Tasks* **click** Create Annotation Sets to create a copy of our training documents for annotation.
**Base set**: `pizza_reviews_text_set`
**Overlap:** `100%`
**Annotator:** `<your id>`
**Set name:** `pizza_anno_set`
**Click:** Generate

**Step 19** Back on the *Create Annotation Task* page
**enter** a *Task name* `pizza_anno_tasks` an optional *Deadline* and **click** Save

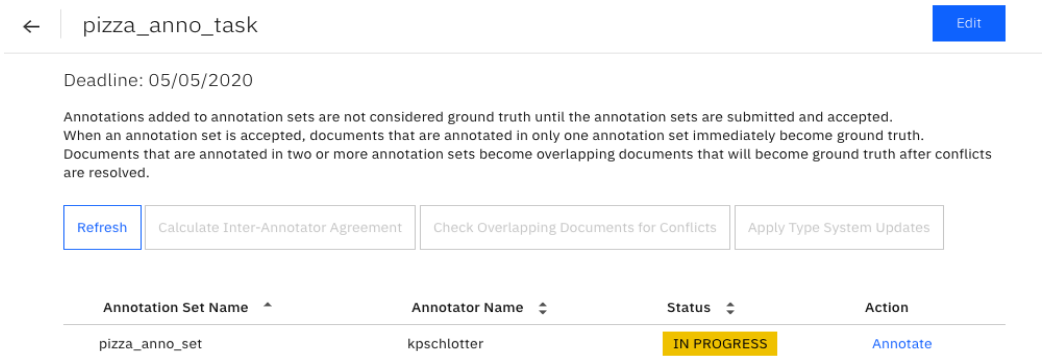
**Step 20** On *Machine Learning Models* → *Annotations* → *Annotation Tasks*, **click** on the task created above.



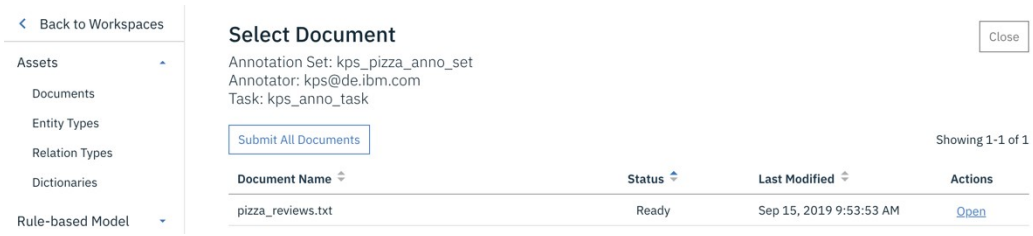

 

**Step 21** The annotation set is in status In Progress because of the pre-annotation we did before. Now we want to manually annotate the other items.

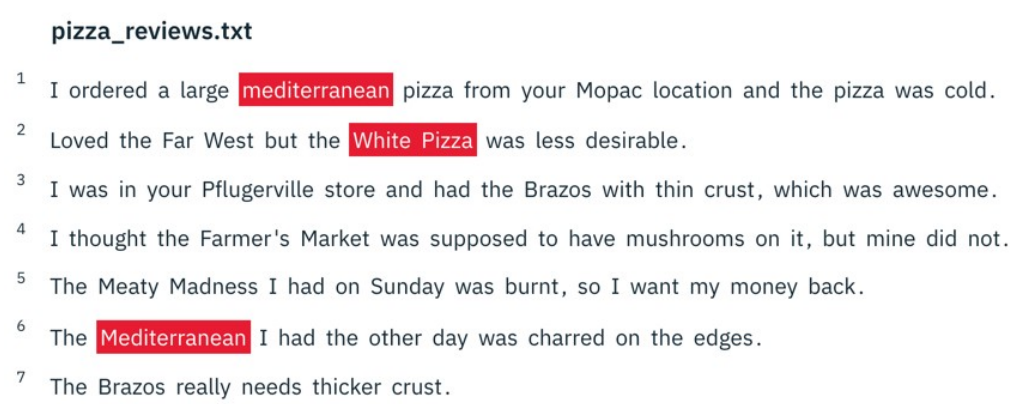
**Click** Annotate.




**Step 22** Open the *pizza_reviews.txt* document.

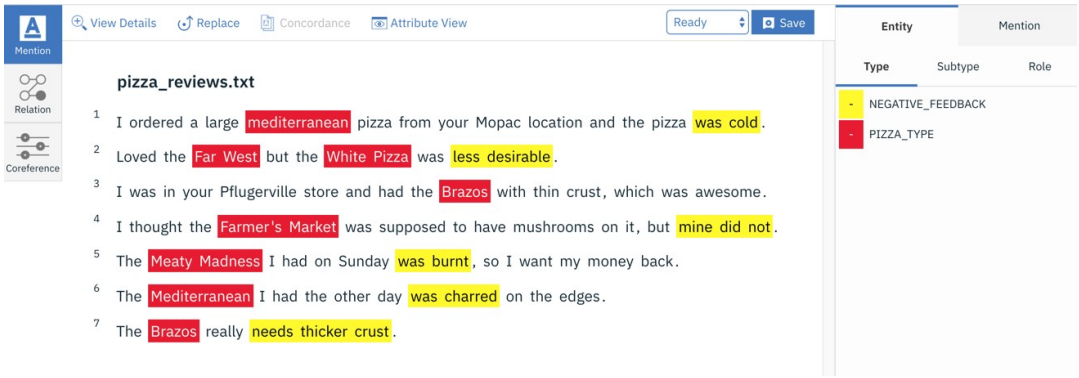



Here you see that the pre-annotation with the pizza_types_dic dictionary already marked as PIZZA_TYPES.

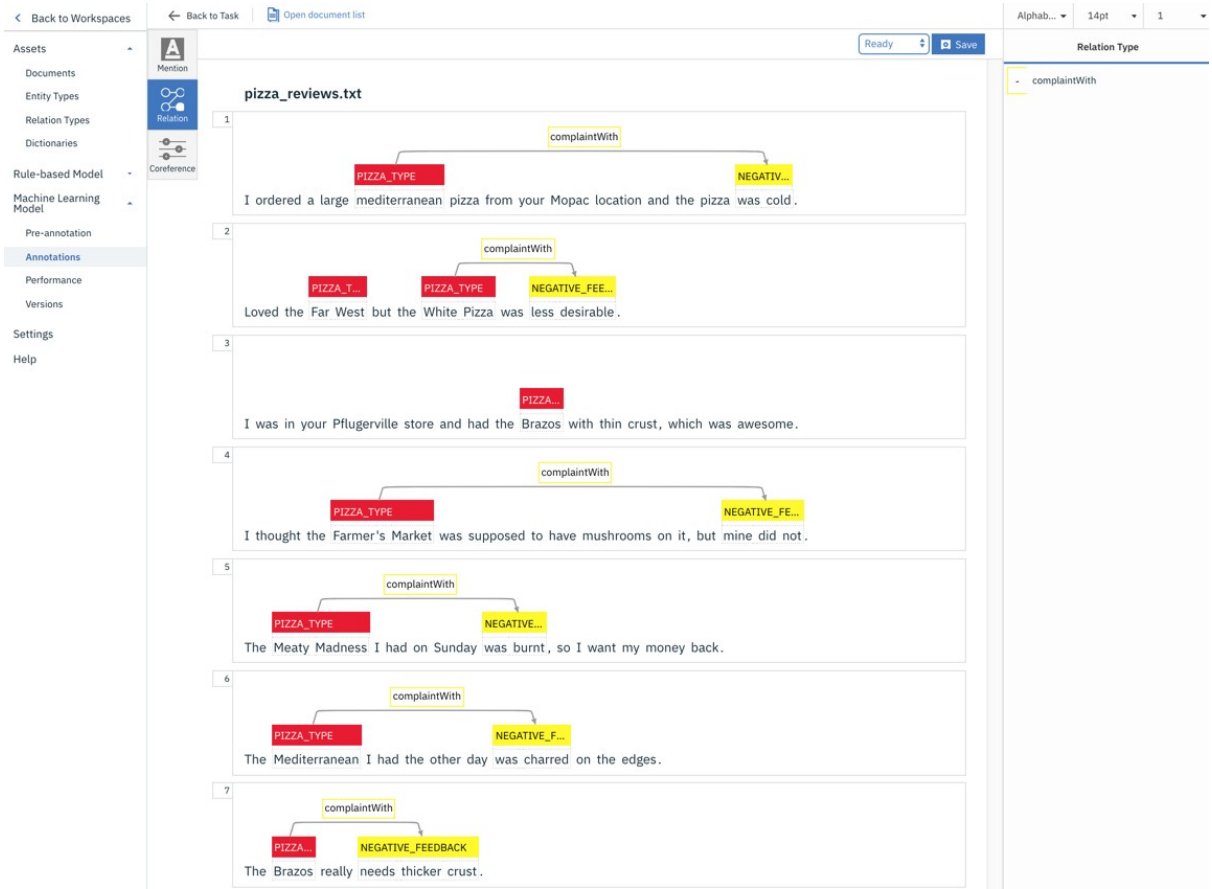


**Step 23** Mark the remaining PIZZA_TYPES and the NEGATIVE_FEEDBACK entries.




**Step 24** **Click** the Relation tab on the document panel. **Click** on the PIZZA_TYPE and on the associated NEGATIVE_FEEDBACK, and from the *Relations Type* side bar **select** *complaintWith*.




**Click** Save.

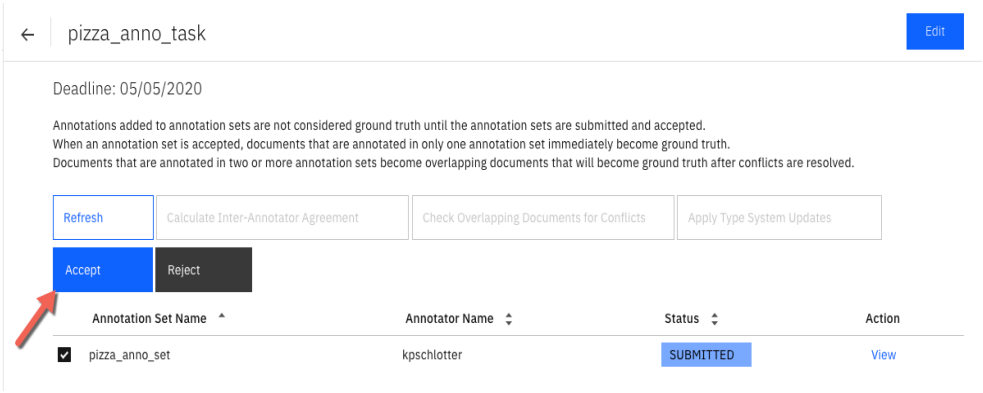
**Step 25** Click Open document list to go back to the document list. You see the above document with status in progress.

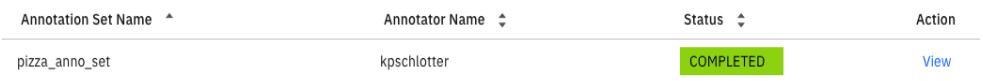
 

**Step 26** On the *Select Document* panel **click** Submit All Documents. Now all documents are in the Completed state.

**Step 27** **Click** Close to close the *Select Document* panel. Then **click** ← Back to Task

**Step 28** On the Annotation Task **Accept** the submitted annotation set. Confirmation with OK

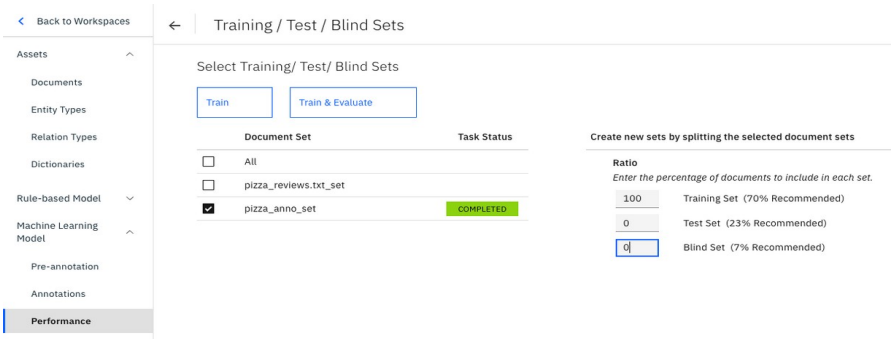


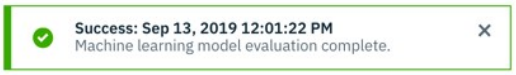

**Step 29** On Machine Learning Model → Performance click Train and evaluate to create the model from our annotations.

**Step 30** On the *Training/Test/Blind Sets* page **select** the document set COMPLETED `100%` *Training Set*, `0%` *Test Set* and `0%` *Blind Set*. (we only have 1 document)

Then **click** Train.

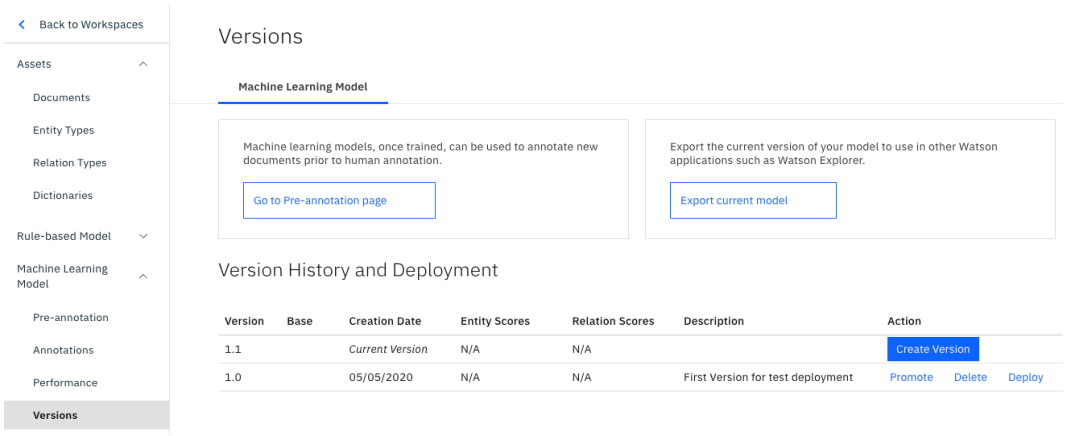



This process takes about 15 minutes.



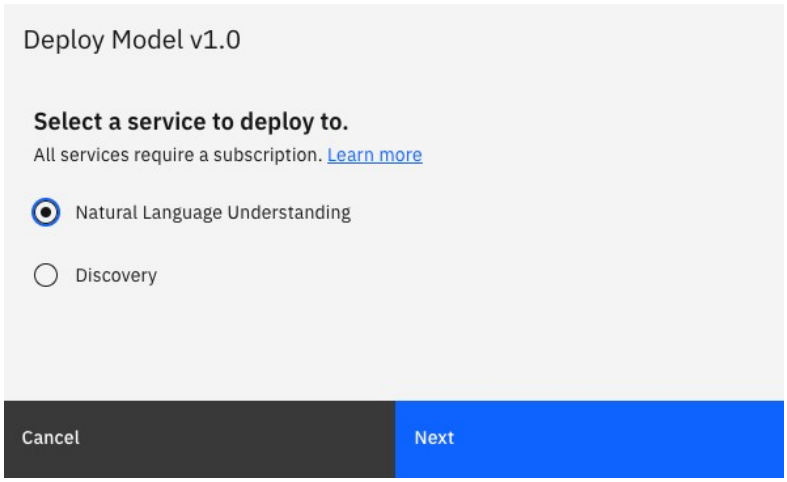

 

**Step 31** On *Machine Learning Model → Versions* **click** Create Version to create a version of your model for deployment. **Enter** a Description and **click** OK.



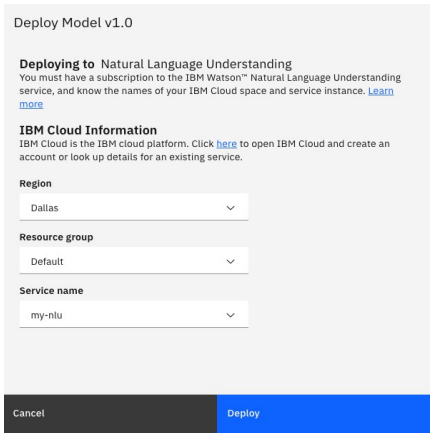

**Step 32** **Click** Deploy to deploy your model.

**Select** *Natural Language Understanding* service created in Lab 4. **click** *Next*.

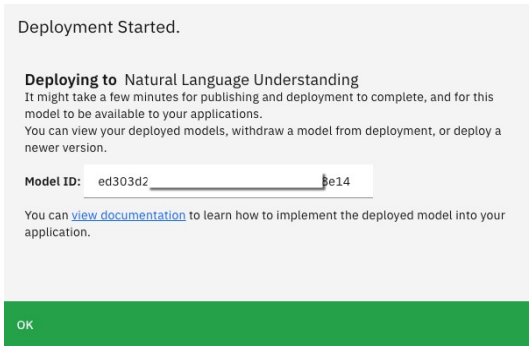


**Step 33** Select your service and **click** *Deploy*.




**Step 34** The deployment process starts. **Click** *OK*.



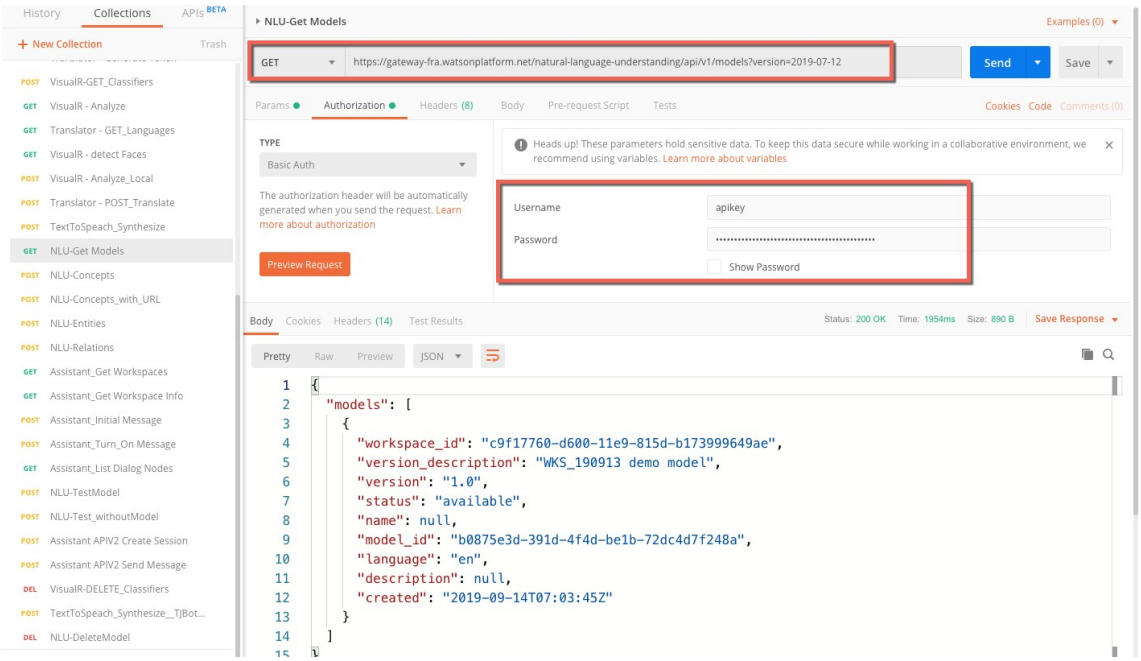

You then have a deployment history. Here you can always see the Id of the deployed model.

# Section 2: Testing the model with Postman (See Lab 4)

**Step 35** List your deployed models. You need Basic Auth (username and password from your Natural Language Understanding Service from your Bluemix account and the following URL: (**See** Lab 4 Step 6)

url: **<yourUrlFromStep3c>/v1/models**
params: `version  2019-07-12`




**Step 36** Test the model with the Basic Auth information from previous step and in the *Headers* section add

key: `Content-type`  value: `application/json`

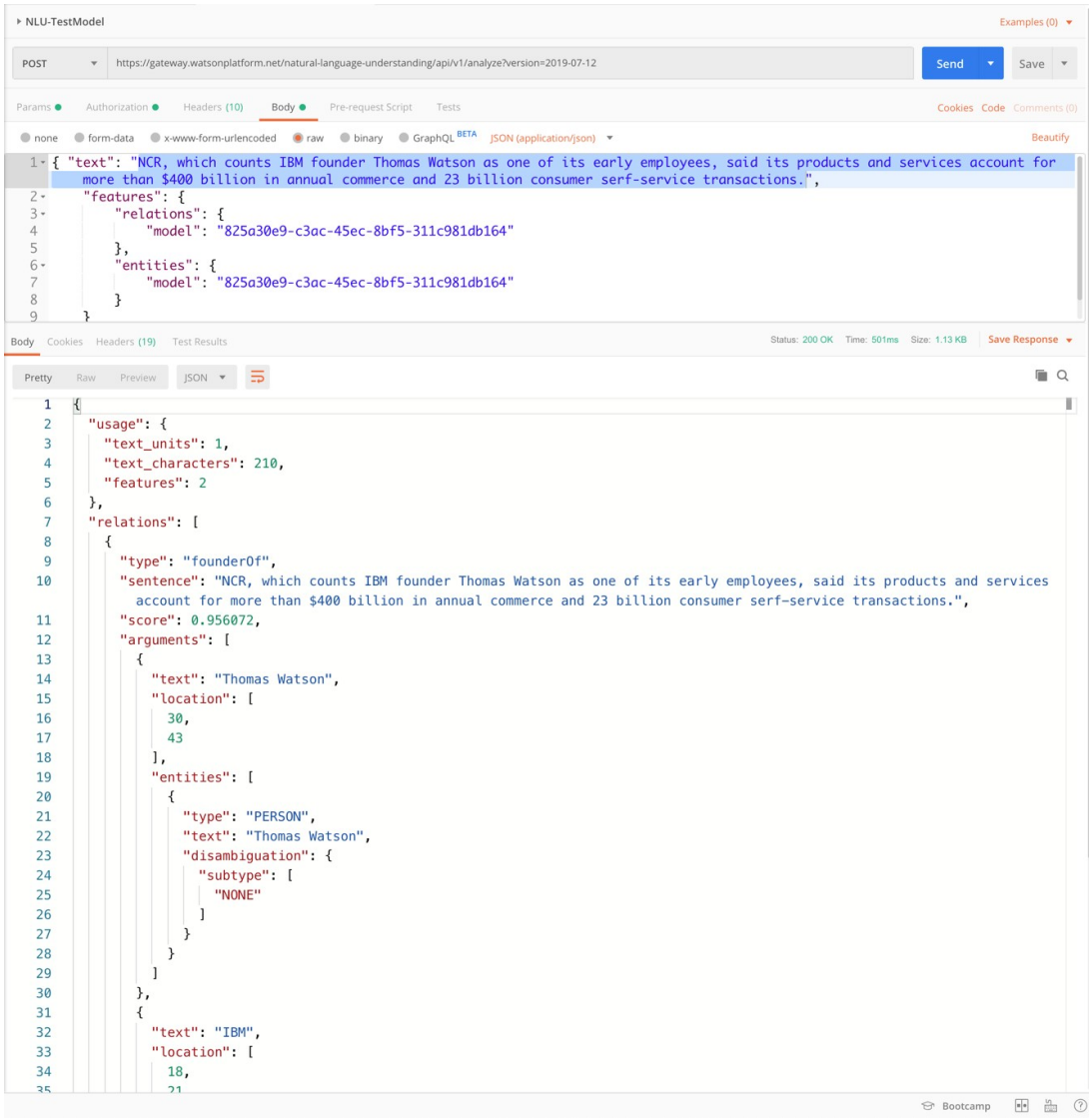
In the *URL* **change** `models` **to** `analyze`

In the Body section enter the text you want to analyze and specify the service feature(s) you want to apply.

```
{ "text": "I ordered a Mediterranean from your mopac location on July 25 and the cheese was stuck to the top of the box.",
"features": { "relations": { "model": "<your model ID goes here>" } } }
```

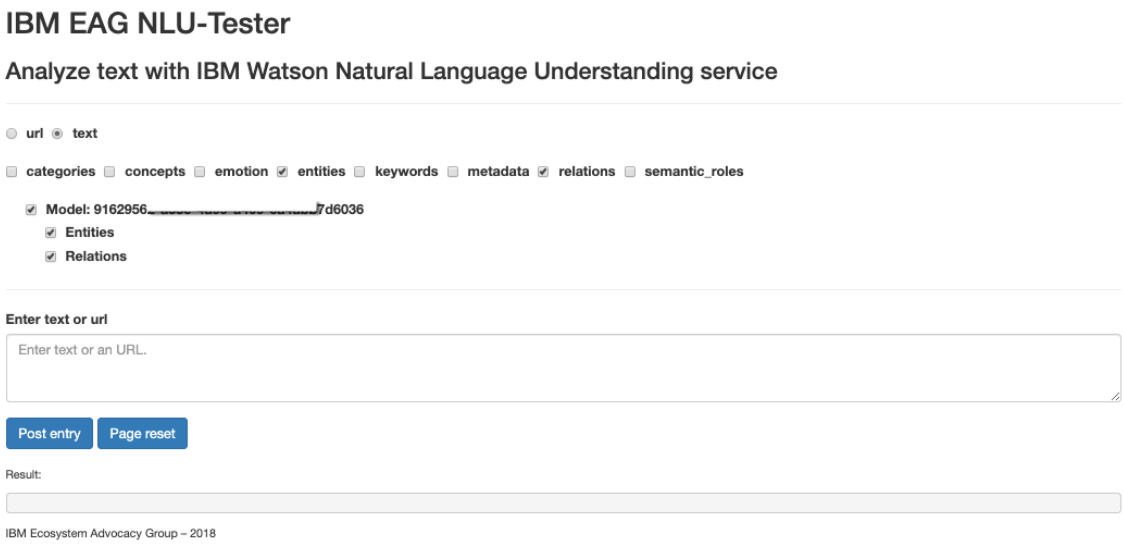
**Step 37** With the following result:



# Section 3: Use the Model in the App from Lab 4

In Lab 4 about the Watson Natural Language Understanding service we created a Node.js web application (Github repository) that retrieves the deployed model at application startup and makes it available for text analysis. In the model we have created entities and relations and these two features can be applied to the text analysis. Test the text from **Step 36** with and without the model selected.




Here you can test the same text sample as with Postman in Section 2.