

# Natural Language Classifier Lab

---

In this lab, you will learn how to setup and train a Natural Language Classifier (NLC) instance. You will train two classifiers, one for testing purposes and one to use in the conversational agent application.

To start with, you need to create an NLC instance in [Bluemix](#) and capture credentials (username and password).

## Conversational Agent NLC

- 1- Cd \$WORKDIR
- 2- Download [conversational agent ASK](#) from [github](#).  
`git clone https://github.com/watson-developer-cloud/conversational-agent-application-starter-kit.git`
- 3- `cd conversational-agent-application-starter-kit/training`
- 4- Remove all rows corresponding to CheckShowtimes intent from classifier\_training.csv  
`grep -v -i checkshowtimes classifier_training.csv > nlc_training_noShowTimes.csv`
- 5- Train an NLC classifier using the training data you just modified.  
`nlcFile= nlc_training_noShowTimes.csv`  
`nlcName=convagent`

```
curl -v -X POST -u "username:password" -F "training_data=@${nlcFile}" -F  
training_metadata="{\"language\": \"en\", \"name\": \"${nlcName}\"}"  
"https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers"
```

➔ This will take some time to train (10 minutes or so)

➔ Capture the classifier\_id, we will need this when accessing this NLC instance.

```
{  
  "classifier_id": "3a84d1x62-nlc-3299",  
  "name": "convagentv1",  
  "language": "en",  
  "created": "2016-05-06T17:17:53.811Z",  
  "url": "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/3a84d1x62-nlc-3299",  
  "status": "Training",  
  "status_description": "The classifier instance is in its training phase, not yet ready to accept classify requests"  
}
```

- 6- To check status of the training, run this command:

```
nlcID=3a84d1x62-nlc-3299;
```

```
curl -u "username:password" https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/${nlcID}
```

```
{  
  "classifier_id": "3a84d1x62-nlc-3299",  
  "name": "convagentv1",
```

```

"language" : "en",
"created" : "2016-05-06T17:17:53.811Z",
"url" : "https://gateway.watsonplatform.net/natural-language-
classifier/api/v1/classifiers/3a84d1x62-nlc-3299",
"status" : "Available",
"status_description" : "The classifier instance is now available and is ready to take classifier requests."
}

```

## Weather NLC

While waiting for this NLC instance to finish training, we will create a simpler NLC instance with less training data and run some experiments. For training data, we will use [weather training data](#) to teach the instance when a text phrase references temperature or conditions.

- 1- cd \$TEST\_WORKDIR
- 2- Download sample test [weather training data](#) and save as weather.csv
- 3- Repeat steps 5 and 6 above until the newly trained NLC instance's status changes to Available.
- 4- When the status of the NLC instance changes to Available, you can now use the classifier. To classify an input text phrase:
 

**nlcID={classifier\_id}** → need to specify the classifier\_id returned for weather NLC instance

**input=Is it foggy today?**

```

curl -X POST -u "username:password" -H "Content-Type:application/json" -d '{"text":"'${input}'"}'
https://gateway.watsonplatform.net/natural-language-
classifier/api/v1/classifiers/${nlcID}/classify"

```

➔ Note it returns result as conditions even though foggy does not show up anywhere in the training data. NLC instance has learned that foggy represents conditions because of other terms in the training data that map to conditions and the strong correlation of foggy to those terms.

## \*\* Optional \*\*

For fun, repeat the exercise above but now edit the weather training data file so that all intents (or classes) are swapped. Basically, any text phrase that mapped to temperature, assign that to conditions and any text phrase that mapped to conditions, assign it to temperature.

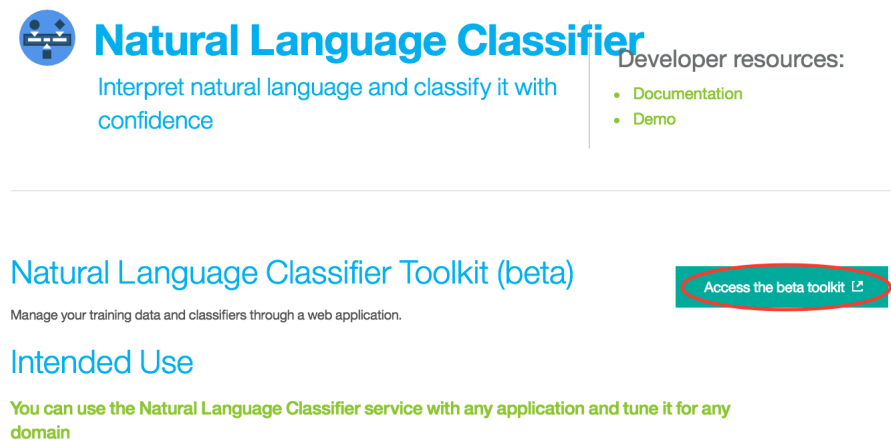
Rerun your earlier tests and note that returned classes would be swapped as expected. Basically, "Is it foggy" should return temperature class.

Even though that is counter-intuitive, the key observation is that with NLC, as with any supervised training solution, it is the training data that would teach the system how to interpret/classify observations. As such, it is critical to define a good and complete ground truth (training data) which is representative of actual end user questions (and phrases) to interpret intent.

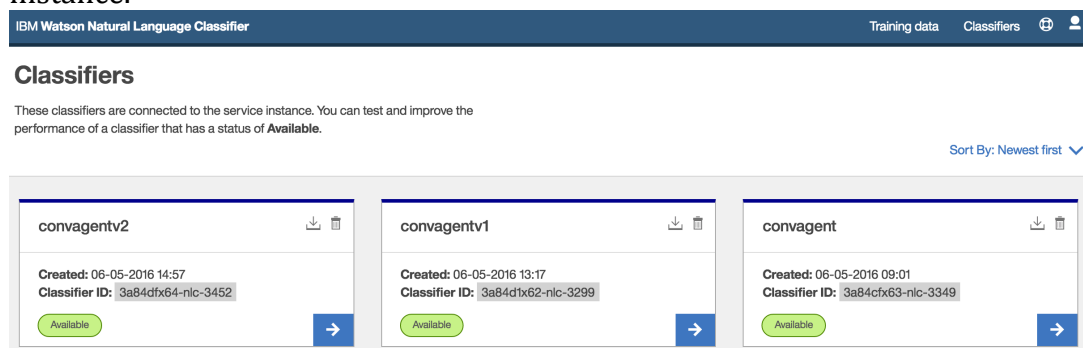
## NLC Toolkit

In this section, we will explore the Natural Language Classifier toolkit (currently in beta). The NLC toolkit allows you to manage and manipulate training data for your classifiers whether you're starting from scratch or from an initial training data set. You can add new classes as well as questions (phrases) and assign them to classes. When ready, you can run training right from the toolkit UI.

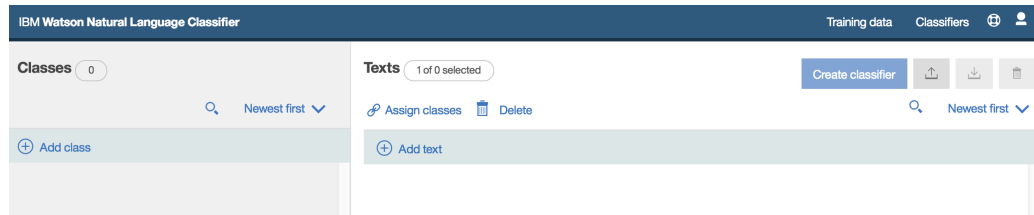
- 1- Log into your Bluemix account.
- 2- From your Dashboard, click on your NLC instance.
- 3- Once that loads, click on Access the beta toolkit button as shown in the figure below.



- 4- Toolkit requests you to confirm access to your NLC instance and classifiers; Accept and this should load up all classifiers associated with your NLC instance.



- 5- Choose your conversational agent classifier by clicking on the right arrow (bottom right of classifier window). This loads up the training UI for the selected classifier.



- 6- Experiment with adding classes and new questions (phrases) and assigning them to the correct classes.
- 7- Experiment with import/export of training data from the toolkit to your local drive.