IBM Watson

Retrieve and Rank Lab

Answer Retrieval Starter Kit

IBM

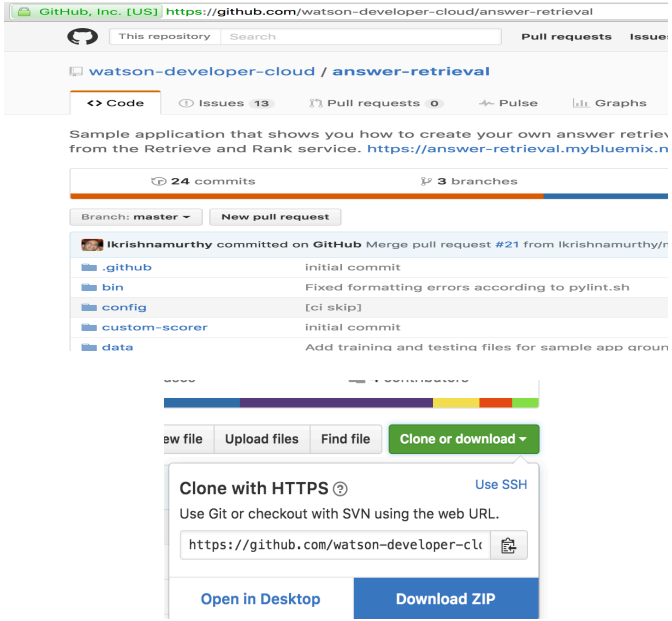
Lab Overview

- Requirements and Dependencies
- Answer Retrieval Setup and Experiment
  - Notebook 1 – Answer Retrieval
- Advanced features Setup and Experiment
  - Notebook 2 – Custom Scorer

System Setup

Retrieve Base Code/Artifacts

- Visit Answer Retrieval Starter Kit Repo
  - https://github.com/watson-developer-cloud/answer-retrieval
- Pull down the repo
  - GitHub CLI
    - *git clone https://github.com/watson-developer-cloud/answer-retrieval.git*
  - (Alternative) Download Zip
- Change to directory where repo cloned



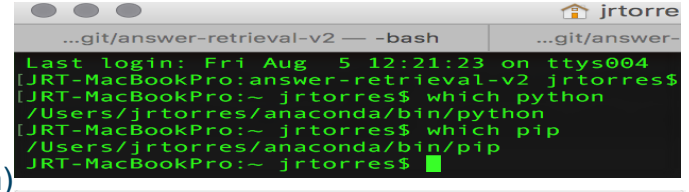

```
JRT-MacBookPro:git jrtorres$ mkdir answer-retrieval-v3
JRT-MacBookPro:git jrtorres$ git clone https://github.com/watson-developer-cloud/answer-retrieval.git answer-retrieval-v3/
Cloning into 'answer-retrieval-v3'...
remote: Counting objects: 521, done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 521 (delta 8), reused 0 (delta 0), pack-reused 496
Receiving objects: 100% (521/521), 15.23 MiB | 701.00 KiB/s, done.
Resolving deltas: 100% (59/59), done.
Checking connectivity... done.
JRT-MacBookPro:git jrtorres$ cd answer-retrieval-v3
JRT-MacBookPro:answer-retrieval-v3 jrtorres$
```

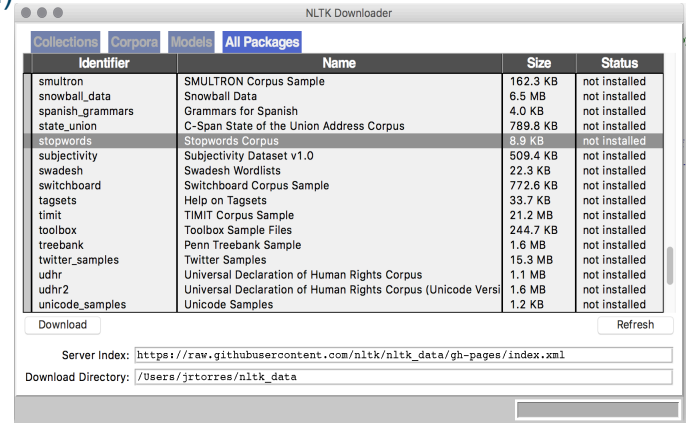
System Requirements

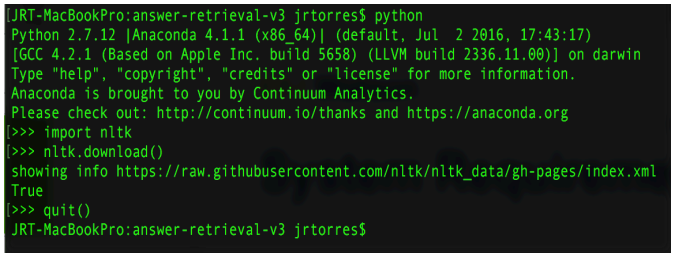
- Install Anaconda (https://www.continuum.io/downloads → default installation)
  - Validation:
    - Terminal / command prompt:
      - *which python*
      - *which pip*
      - *pip freeze*

      → **Should point to anaconda installation directory**
    - If necessary, open a new terminal / command prompt

- Install NLTK Corpora
  - Start python interpreter from Terminal / command prompt
    - *python*
  - From python interpreter
    - *import nltk*
    - *nltk.download()*
  - Use the NLTK Downloader GUI to download stopwords
    - Click on the 'All Packages' tab and find/select "stopwords"
    - Click download (wait until it finished then close the application)
  - From the python interpreter in the terminal enter "quit()" on the interpreter







Lab Requirements

NOTE: MAKE SURE YOU ARE IN THE PARENT DIRECTORY WHERE GIT REPO WAS CLONED.

- Install python requirements
  - Terminal / command prompt:
    - » *pip install -r requirements.txt*
    - » *pip install -r notebooks/requirements.txt*

```
JRT-MacBookPro:answer-retrieval-v2 jrtorres$ pip install -r requirements.txt
Processing ./custom-scorer
Requirement already satisfied (use --upgrade to upgrade): requests==2.10.0 in /Users/jrto
b/python2.7/site-packages (from -r requirements.txt (line 2))
Collecting spacy==0.101.0 (from -r requirements.txt (line 3))
Requirement already satisfied (use --upgrade to upgrade): numpy==1.11.1 in /Users/jrtorre
ython2.7/site-packages (from -r requirements.txt (line 4))
Requirement already satisfied (use --upgrade to upgrade): futures>=3.0.5 in /Users/jrtorr
python2.7/site-packages (from -r requirements.txt (line 5))
Requirement already satisfied (use --upgrade to upgrade): flask==0.11.1 in /Users/jrtorre
ython2.7/site-packages (from -r requirements.txt (line 8))
Collecting python-dotenv (from -r requirements.txt (line 9))
  Using cached python_dotenv-0.5.1-py2.py3-none-any.whl
Collecting watson-developer-cloud (from -r requirements.txt (line 10))
Collecting murmurhash<0.27,>=0.26 (from spacy==0.101.0->-r requirements.txt (line 3))
  Using cached murmurhash-0.26.4-cp27-cp27m-macosx_10_6_intel.whl
Requirement already satisfied (use --upgrade to upgrade): cloudpickle in /Users/jrtorres/
hon2.7/site-packages (from spacy==0.101.0->-r requirements.txt (line 3))
```

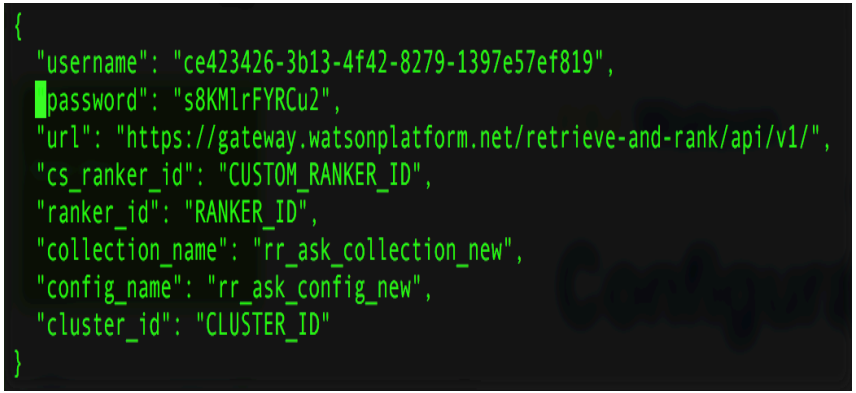
- [Optional] As validation list installed packages and check for packages listed in requirements.txt file
  - Terminal / command prompt:
    - *pip freeze*

```
JRT-MacBookPro:answer-retrieval-v2 jrtorres$ pip freeze
alabaster==0.7.8
anaconda-client==1.4.0
anaconda-navigator==1.2.1
appnope==0.1.0
appscript==1.0.1
argcomplete==1.0.0
astropy==1.2.1
Babel==2.3.3
backports-abc==0.4
backports.shutil-get-terminal-size==1.0.0
backports.ssl-match-hostname==3.4.0.2
beautifulsoup4==4.4.1
bitarray==0.8.1
blaze==0.10.1
bokeh==0.12.0
boto==2.40.0
Bottleneck==1.1.0
cdecimal==2.3
cffi==1.6.0
```

Configuration Files

NOTE: MAKE SURE YOU ARE IN THE PARENT DIRECTORY WHERE GIT REPO WAS CLONED. ENSURE YOU HAVE YOUR R&R SERVICE INSTANCE PROVISIONED.

- Credentials File
  - Modify the credentials.json file under the config directory.
  - Add the username and password for the Retrieve and Rank service.
    - Leave other parameters as is.



Answer Retrieval Part 1

Experiment Data Files

NOTE: MAKE SURE YOU ARE IN THE PARENT DIRECTORY WHERE GIT REPO WAS CLONED.

- Setup data files in appropriate directories
  - solrDocuments.json under data/content
  - answerGT* under data/groundtruth

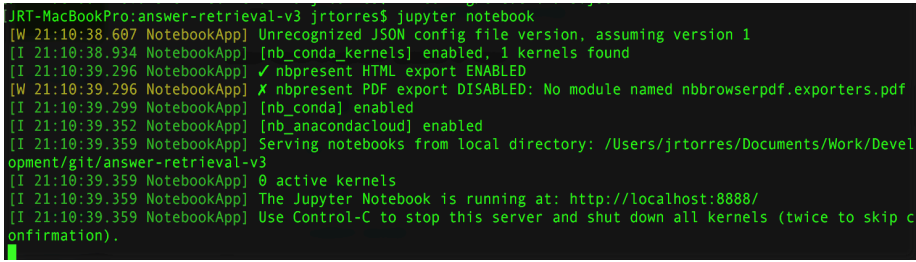
- Move/Rename existing files if appropriate.

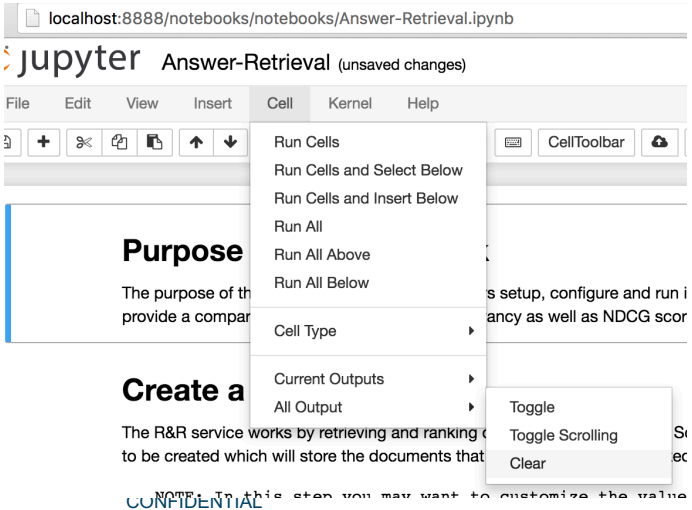
```
JRT-MacBookPro:answer-retrieval-v3 jrtorres$ ls data/content/
solrDocuments.json      solrDocuments.json.orig
JRT-MacBookPro:answer-retrieval-v3 jrtorres$ ls data/groundtruth/
answerGT.csv.orig               answerGT_test_sample200.csv     answerGT_train_sample500.csv
answerGT_test.csv.orig          answerGT_train.csv.orig
answerGT_test_full.csv          answerGT_train_full.csv
JRT-MacBookPro:answer-retrieval-v3 jrtorres$
```

Answer Retrieval Notebook

NOTE: MAKE SURE YOU ARE IN THE PARENT DIRECTORY WHERE GIT REPO WAS CLONED.

- Start Jupyter Notebook
  - Terminal / command prompt:
    - » *jupyter notebook*
- Click on the notebooks directory and click on”Answer-Retrieval.ipynb”
- Clear all cell output
- Run all cells
  - Optionally change variables of the cells as needed
- Rename the trainingdata.csv (data/groundtruth) file for later comparison



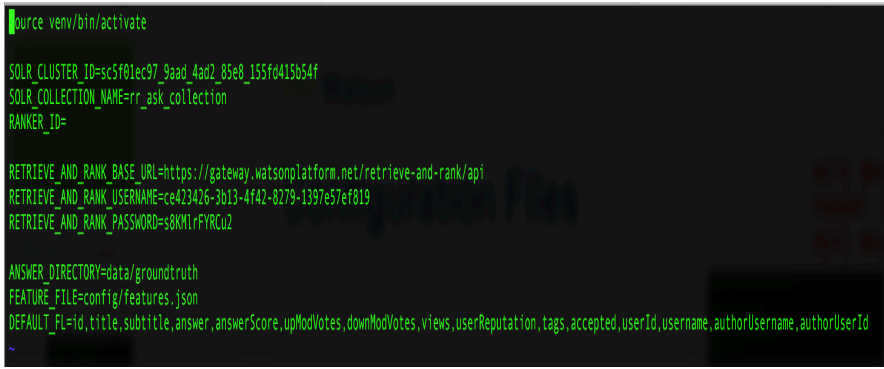


Answer Retrieval Part 1

Configuration Files

NOTE: MAKE SURE YOU ARE IN THE PARENT DIRECTORY WHERE GIT REPO WAS CLONED. ENSURE YOU HAVE YOUR R&R SERVICE INSTANCE PROVISIONED.

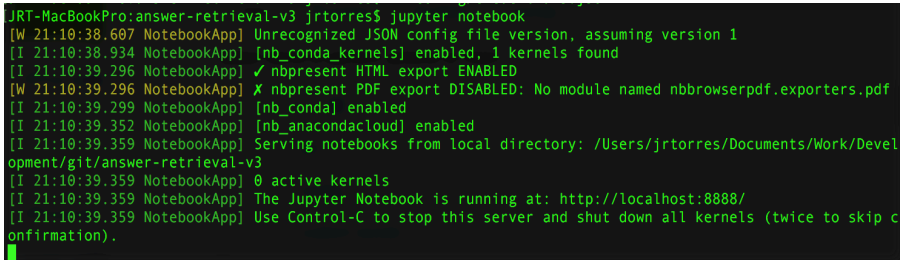
- Environment file
  - create the .env file using .env.example as a starting point
    - Add the service credentials, collection name, cluster id

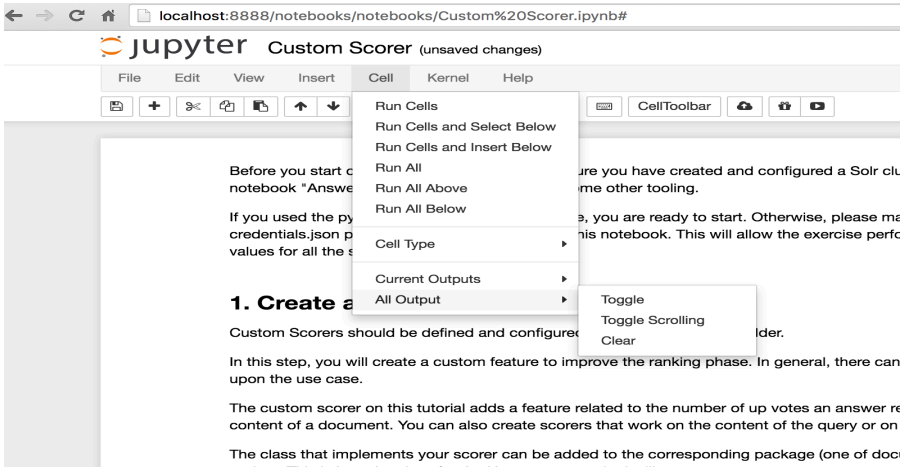


Custom Scorer Notebook

NOTE: MAKE SURE YOU ARE IN THE PARENT DIRECTORY WHERE GIT REPO WAS CLONED.

- Start Jupyter Notebook
  - Terminal / command prompt:
    - » *jupyter notebook*
- Click on the notebooks directory and click on “Customer Scorer.ipynb”
- Clear all cell output
- Run all cells
  - Optionally change variables of the cells as needed





Thank You

FAQ

- Testing can be done from the browser :
  - q=What%20are%20the%20pros%20and%20cons%20of%20buying%20a%20house%20using%20an%20FHA%20loan&wt=json&fl=id,title,answer,answerScore,accepted,authorUserId,authorUsername,downModVotes,subtitle,tags,upModVotes,userId,userReputation,username,views"
- Directory structure issues:
  - Dont remove directory structure of repo/data - it expects data/groundtruth to exist to generate the training data.
- Notebook cells can be re-run by clicking them and selecting ‘Run cell, select below’
- As long as a cell has an asterisk next to it, the cell is still running

```
In [*]: import subprocess
        import json
        import shlex
        import os
        import pysolr
        from watson_developer_
```

- Do not add carriage returns in JSON file.
-

FAQ

- Generating the training data will fail on Windows.
  - Under the code for ”Generate Training Data”
  - Modify the “TRAIN_FILE_PATH” and “GROUND_TRUTH_FILE” variables to point to explicit fully qualified paths (using double quotes on whole path. Example:
    - TRAIN_FILE_PATH = ’”C:/*PATH_TO_GITHUBREPO*/bin/python”’
    - GROUND_TRUTH_FILE=’” C:/*PATH_TO_GITHUBREPO*/data/groundtruth/answerGT_train.csv”’

Generate Training Data

Once the training ground truth file is ready, a file which contains the feature vectors for each questions needs to be generated. Run the command below to generate the traingdata.csv file, which will be saved on the file system and used to train the ranker in the next step.

Note this step may take long time. Wait for this step to complete before moving to next step!

```
import subprocess
import json
import shlex
import os

#getting current directory
curdir = os.getcwd()

#loading credentials
credFilePath = curdir+'/../config/credentials.json'
with open(credFilePath) as credFile:
    credentials = json.load(credFile)

BASEURL=credentials['url']
SOLRURL= BASEURL+"solr_clusters/"
RANKER_URL=BASEURL+"rankers"
USERNAME=credentials['username']
PASSWORD=credentials['password']
SOLR_CLUSTER_ID=credentials['cluster_id']
COLLECTION_NAME=credentials['collection_name']
TRAIN_FILE_PATH=curdir+'/../bin/python'
GROUND_TRUTH_FILE=curdir+"/../data/groundtruth/answerGT_train.csv"

#Running command that trains a ranker
cmd = 'python %s/train.py -u %s:%s -i %s -c %s -x %s -n %s' %\
    (TRAIN_FILE_PATH, USERNAME, PASSWORD, GROUND_TRUTH_FILE, SOLR_CLUSTER_ID, COLLECTION_NAME, "travel_ranker")
try:
    process = subprocess.Popen(shlex.split(cmd), stdout=subprocess.PIPE)
    output = process.communicate()[0]
    print output
```