This file describes the whole layout of the code base:

|  |  |  |
| --- | --- | --- |
| Folder/File | Description | Usage |
| py/ProcessQueryJson.py | NLP pipeline to extract keywords from question and detect answer type | Run the script, interact with the script  For every question submitted, a json string is generated  This json string should be saved to a file and given to pig/Searcher.pig |
| pig/Searcher.pig | The search script which takes in a json query and outputs results to a specified file | pig -f Searcher.pig -param jsonFile='<json file path>' -param outFolder='<output folder for results>' |
| py/ProcessAnswers.py | NLP pipeline to extract answers from pig results | python ProcessAnswers.py <folderWithPigOutputs> |
| py/QueryTranslator.py | Translates the results of ProcessQuery to a json for the pig file | Not for user interaction |
| hadoop/inversedoc | Code for calculating IDF for words in the corpus.  This is a pre-processing step  The results are used in pig/Searcher.pig to calculate tfidf scores | Not for user interaction |
| hadoop/xmlparser | Code for getting all data sources to our common format of (title, text, docID)  This is a pre-processing step | Not for user interaction |
| pig/lib | Jars used by the pig script | N/A |
| pig/UDFs | Filter and Integer UDFs for pig script .   1. Filter articles which satisfy the given query (pig/UDFs/src/pig/SatisfiesQuery.java) 2. Score articles by their proximity score (pig/UDFs/src/pig/ScoreGen.java) | Not for user interaction |
| data/idf.txt | Pre-processed IDF values for the whole corpus | N/A |
| data/sample.xml | Sample articles in our common text corpus format (title, text, id) | N/A |
| data/ sample\_queries | Sample json queries generated by py/ProcessQueryJson.py | N/A |
| data/sample\_outputs | Sample outputs from pig/Searcher.pig | N/A |