Steps to run the files needed for project:

1. Complete the installation steps as stated in the Installation Guide
2. First step in the project is to text extraction,

For this please take a look at files, xml\_to\_trectext.py and xml\_to\_trectext\_all.py – The usage instructions are mentioned on the top of the python script. These are used to convert xml files to trec standard text files. The other script in text extraction is text\_to trectext.py, this is used for extra abstracts that are present in text format.

1. Once the documents are extracted, a parameter file has to be developed to build a index of these documents.

In order to build index, in terminal navigate to the folder of indri and execute the following command

./IndriBuildIndex /Users/Documents/build\_index\_params\_ja.xml

The parameters file used for the project is present in the GitHub with name “build\_index\_params\_ja.xml”

1. The next step is query formulation, for this xml\_to\_params.py has to be run. For this purpose please refer to mainfileforqueryformulation.py. Before running this script, please create a file named umls\_api\_key and place your api key that was request through your UTS account. Also in E-utilities, at line 13, please provide your school email id to submit requests.
2. The output file from step 4 is required to run against the document index. For this, in terminal, navigate to the folder of indri and run the following command

./IndriRunQuery /Users/Documents/precision\_medicine\_contd/ /topics2018\_formulated.xml > output\_run.txt

In the above command topics2018\_formulated.xml is the output file from step 4 and the output from the command is written in file output\_run.txt

1. Download the qrels from trec website that constitute for training data
2. In order to extract the documents for the qrels, please run the python script with name mainfiletotrainqrels.py
3. In order to extract documents for test data, please run the python script mainfiletoextractdocsfortest.py
4. Now from step 7 and 8, there are 2 document files that are generated, these has to be the files in line 130 and 139 of script crossvalidation.py from where the documents are loaded and features are generated. If same features are being used, run generatedruggenefeatures.py before running crossvalidation.py script
5. In order to run the cross validation script, use mainfileforcrossvalidation.py which runs crossvalidation.py and generates feature files
6. In order to do paired t-tests, use mainfileforpairedtests.py

Extra Steps:

1. For step 7 and 8, get\_docs\_by\_file.py has 2 main methods, both of them work. For executing the main function at line 41 (does faster processing), the dumpindex.cpp file has to be replaced with the dumpindex file attached in the github and then run make.
2. For obtaining and using the tfidf and bm25 scores, run line 16 in mainfileforqueryformulation.py. This generates a parameter file which should be used in the following command

./IndriRunQuery /Users/Documents/precision\_medicine\_contd/ /topics2018\_basescores.xml > output\_tfidf\_run.txt

To use these scores in the features, please input the file names in lines 103, 104, 107, 108.

1. For obtaining basescores run ﻿xml\_to\_params.main(filename='remaining2018topics.xml', pmidfile='remaining2018mysyspmidfile.txt',prf=True, prfparams=(2,20,0.5,0.8))
2. If your development requires running the entire document against metamaplite to parse and derive the disease type, gene type and drug type. Then please refer scripts, metamaplitedir.py and tempmetamaplite\_restrict.py. This requires installation of metamaplite in the system.