Project Documentation

Srujan
Netid:ssg7
LiveLab ID:srzn

December 13, 2020

Included is the project flow, definition and function of each module and usage details

1 Project Flow

I participated in the IR competition. I used MP2.4 code template to implement various state of the art ranking functions for IR evaluation and ranking

1.1 Converting Data

Format of the data provided is incompatible with that of my code. So, the first step is data conversion. Queries are provided in xml and need to be converted into a text file, each query per line. Actual documents are provided as a csv file with duplicate entries per document and a lot of irrlevant information. Each document is identified by an alphanumeric id. Relevant data from the csv file need to be extracted and populated as document data with each document occupying one line in the final .dat file. My code takes document ids as numbers. So, a dictionary with uid to number mapping needed to be built. The entire process needed to be implemented for both training and testing data. This concludes data conversion.

1.2 Training Step

I used 6 IR ranking functions to train on the dataset to extract optimal parameters for each of the ranking functions. The user running my training program is given a choice/option to select any one of these ranking functions. One of the ranking function,InL2 ranker, was overloaded with my custom scoring function. I swept each ranking function with a number of parameters over the given training dataset and labels. The combination of parameters that lead to the best NDCG@20 were saved to a file named "Option< num >.txt" These were the ranking functions used:

- Option 0 : BM25

- Option 1: InL2 ranker
- Option 2: InL2 ranker
- Option 3: Jelinek-Mercer Smoothing
- Option 4: Dirichlet Prior Smoothing
- Option 5: Absolute Discounting Smoothing

1.3 Testing Step

From the "Option< *num* >.txt" I obtained from the training step, we use corresponding ranking functions to obtain ranking scores of each document w.r.t a query. Test data is slightly different from training dataset. So, inverted indices were built separately for training dataset as well.

2 Module Definitions

The code files in both testing and training have similar names and functions. Following code files can be found in my software package:

2.1 queryextr.py

This file is used to convert query.xml into queries.txt and queries-test.txt

2.2 convert-dat1.py

This file is used to extract data from metadata.csv and document jsons to build SarsCov.dat file which is inturn used to build inverted index and used to evaulate the ranking functions

2.3 uidmap.py

This file is used to build dictionaries "uidmap.txt", "uidrevmap.txt". These dictionaries contain docid to uid mappings and vice-versa.

2.4 search_eval.py

This is the main program that runs the training phase to output best parameters that generate the highest NDCG

2.5 search_test.py

This is the main program that ranks and scores each document from test set against the test queries and outputs a "testpredictions.txt" output which has docids instead of uids

2.6 predictgen.py

Uses the dictionaries obtained from “uidmap.py” and produces the final output “predictions.txt” that has uids

3 Usage

3.1 Training

In the directory that has search_eval.py, run

python search_eval.py config.toml 'Option'

where 'Option' has the same range as described in “Training step”. You'll obtain a text file "Option0.txt" (if you chose Option 0). Please copy this file over to the "test" directory that has “search_test.py”

3.2 Testing

After the above step, in the directory that has “search_test.py”, run

python search_test.py config-test.toml 'Option'

where 'Option' can take any integer value between 0 and 5. Please refer the documentation to know more about each choice.

For example, if you chose Option 0 then you'll run

python search_test.py config-test.toml 0

And then run,

python predictgen.py

The above code must be executed to obtain the final predictions in a text file. If you chose Option 0 you would obtain "prediction0.txt" as your final predictions for all the queries capping at 1000 top documents per query.

4 Leaderboard

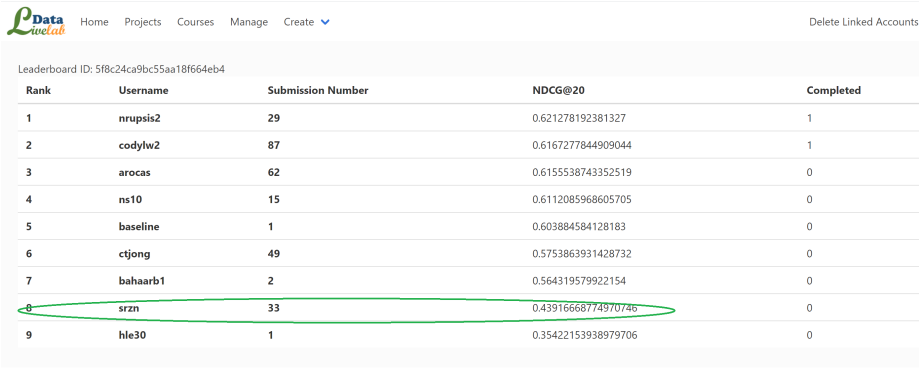

Figure 1: Leaderboard as of 12/13/2020 9:38 PM EST