Build Script (Team1)

Amith Ramanagar Chandrashekar Pooja Himanshu Oza
Rachel E Cates Ahmed M Alnazer

March 21, 2019

1 Dependency

Java version: Java8 or Higher

Build Tool: Maven

2 How to Run

Please perform the below steps

- sudo git clone https://gitlab.cs.unh.edu/cs953-2019/cs953-team1.git
- sudo chmod -R 777 cs953-team1
- cd cs953-team1
- ./install.sh (please run the ./install.sh in one of the tmux session as described below)

2.1 Tmux session creation

- tmux new -s team1 (create new session)
- Run the ./install.sh inside the tmux
- ctrl+b+d (Disconnect from the session)
- tmux a -t myname (To attach to the session)

2.2 Note

Running the ./install.sh creates all the runs files in the **result** directory.we have the Benchmark-Y1 test as the query file in the install script.

Also Self executable jar can be created using the maven and the program can run independently without the install script. Please follow below steps

- mvn clean compile
- mvn package
- java -jar cs953-team1-1.0-SNAPSHOT-jar-with-dependencies.jar sub-commands option

2.3 General Note

The below error can be ignored. This error is from the Nd4j.

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.htmlStaticLoggerBinder for further details.

3 MRF Method and Feature vector generation

The mentioned methods is not part of the install script because it is very intensive process which takes more time to run

3.1 MRF

This is to outline of how the MRF output is generated.

- Generate the feature file for the query-document similarity as described in the report for the Benchmark-Y1 train.
- Run the RankLib to get the model file for the train
- Generate the feature file for the query-document similarity as described in the report for the Benchmark-Y1 test.
- Combine the weight vector learnt with the test feature to get a scalar value for each document.
- All the data can be found at /home/team1/prototype2/amith_data

3.2 Feature vector generation

This is also not part of the script and all the feature vector can be found at this location /home/team1/prototype2/pooja_data

4 Result on Benchmark-Y1 Test

All the method results for the Benchmark-Y1 Test can be found at **"/home/team1/nospamenabled"**.

Ground truth files can be found at **"/home/team1/query_data/benchmarkY1-test"**

Trec eval tool can be found at **"/home/team1/trec_tool/trec_eval"**

5 Command Line Option

These are the command line options we have implemented.

```
Usage: <main class> [command] [command options]
  Commands:
    index      Command to Index the Corpus
      Usage: index [options]
        Options:
          --abstract-index
            Perform Entity Abstract Index
            Default: false
        * -i, --corpus-file
            Corpus file to index. In case of Entity Abstract, please specify
            Entity Index location
          -d, --dest-location
            Location to save the index file
            Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
          --entity-index
            Perform Entity Index
            Default: false
          --help

          --para-index
            Perform Paragraph Index
            Default: false

    search      Command to search
      Usage: search [options]
        Options:
          --bias-fact
            Bias factor to get the document representation
            Default: 1
          -k, --candidate-set-val
            How many candidate set to retrieve using BM25
            Default: 100
          --cluster
```

```
    Cluster Ranking
    Default: false
  --cosine-sim
    Rerank the document based on the cosine similarity between two
    strings
    Default: false
  -bm25, --default-bm25
    Rerank the initial retrieved cluster using document similarity
    Default: false
  --dice-sim
    Rerank the document based on the Sorensen Dice coefficient
    similarity between two strings
    Default: false
  --entity-degree
    Rerank the initial retrieved document using entity degree
    Default: false
  --entity-doc-sim
    Rerank the initial retrieved document using entity abstract
    similarity
    Default: false
  --entity-expand
    Rerank the initial retrieved document using expanded query
    Default: false
  --entity-index
    Pass the index location of entity index
  --entity-relation
    Rerank the initial retrieved document using entity relationship
    Default: false
  --entity-sim
    Rerank the initial retrieved document using entity abstract
    similarity
    Default: false
  --ham-loc
    Directory to ham train file
  --help

* -i, --index-loc
    Indexed directory to search
  --jaccard-sim
    Rerank the document based on the Jaccard similarity between two
    strings
    Default: false
  --jaro-sim
    Rerank the document based on the Jaro Winkler similarity between
    two strings
    Default: false
```

```
  --leven-sim
    Rerank the document based on the NormalizedLevenshtein similarity
    between two strings
    Default: false
  --mrf
    Uses the parallel stream for the reranker methods
    Default: false
  --parallel
    Uses the parallel stream for the reranker methods
    Default: false
  --qe-entity-degree
    Query expansion based on the entity degree and reranking
    Default: false
  --qe-reranking
    Query expansion method based on the Entities and reranking
    Default: false
  --qrel-path
    Pass the absolute path of the Qrel
* -q, --query-cbor
    Query file (CBOR file)
  -qe, --query-expansion
    Rerank the document using Query expansion
    Default: false
  -qe-type, --query-expansion-type
    Select type of Query expansion (entityText, entityID ,
    entityTextID)
    Default: entityText
    Possible Values: [entityText, entityID, entityTextID, entityIDInEntityField]
  --rank-lib
    Provide path to the Ranklib
  --rerank
    Rerank the initial retrieved document using document similarity
    Default: false
  --rerank-df
    Rerank the document based on the DF
    Default: false
  --rerank-idf
    Rerank the document based on the IDF
    Default: false
  --spam-filter
    Uses the spam filter before performing the re-rank
    Default: false
  --spam-loc
    Directory to spam train file
  -V, --verbose
    Print out  some of the results  into stdout
```

```
        Default: false
      -dim, --word-dimension
        Dimension of the Word embeddings
        Default: 0
      -we, --word-embedding
        Pass the word embedding file GloVe/ Word2Vec
      -top
        specify the top number of selected entity to used in the Query
        expansion
        Default: 3
      article
        Article level retrieval
        Default: false
      section
        Section level retrieval
        Default: false

indexHamSpam      Command to create training and test data for the spam
        classifier
  Usage: indexHamSpam [options]
    Options:
      --help

      -p, --paragraphs-file
        paragraph corpus directory
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
      -q, --qrels-file
        qrels file
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
      -hamSpamTest
        Location to save the ham and spam test data
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
      -hamTest
        Location to save the ham test data
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
      -hamTrain
        Location to save the ham training data
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
      -spamTest
        Location to save the spam test data
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file
      -spamTrain
        Location to save the spam training data
        Default: C:\Users\amith\IdeaProjects\cs953-team1\indexed_file

--help      Help Information
```

```
  Usage: --help

ranker      Ranker
  Usage: ranker [options]
    Options:
      --mname
        Method name suffix
        Default: mrfupdated
    * --model-file
        Location of the model file
    * --run-file
        Location of the run file
```