Report on Final Project (Watson)

https://github.com/wenkanng/watson-ir

1. How to run the code (Index links are in Section 3)

Before compile: assume the questions.txt file is in src/main/resources and the index file is in somewhere(since the full absolute path will be passed as an argument, example in the image below).

- To compile:

  `mvn compile`

- To run without indexing (assume the index is present in the right place):

  `mvn -q -e exec:java -Pproject -Dexec.args="{} no"`

  NOTE: The first command line argument is the absolute path of the index, need to substitute "{}" with proper path(full absolute path) in order to make the program work properly.

- To run with doing indexing first:

  `mvn -q -e exec:java -Pproject -Dexec.args="{} yes"`

```
$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< edu.arizona.cs:project >----------------------
[INFO] Building project 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 82 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ project ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/wenkangzhou/Developer/583/project/project/target/cl
asses
[WARNING] /Users/wenkangzhou/Developer/583/project/project/src/main/java/edu/arizona/cs/Jeop
ardy.java: /Users/wenkangzhou/Developer/583/project/project/src/main/java/edu/arizona/cs/Jeo
pardy.java uses or overrides a deprecated API.
[WARNING] /Users/wenkangzhou/Developer/583/project/project/src/main/java/edu/arizona/cs/Jeop
ardy.java: Recompile with -Xlint:deprecation for details.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.006 s
[INFO] Finished at: 2019-04-24T13:39:57-07:00
[INFO] ------------------------------------------------------------------------
$ mvn -q -e exec:java -Pproject -Dexec.args="/Users/wenkangzhou/Developer/583/project/projec
t/fsd yes"
indexing...
```

Above is an example on compiling and run with indexing, output after "indexing" is omitted.

2. Code structure

- **`public static void indexing(String indexFilePath)`**

  The function is to do the indexing part of the system.

- **`public static void searchWithFileOfQuestions(String indexFilePath)`**

The function is to do the retrieval and evaluation part, in the function, it keeps track of how many questions in the given “questions.txt” file and how many of them are answered correctly by checking the precision at 1. When doing retrieval, the program outputs the unmodified question part in the “questions.txt” and then the first 20 hits for debugging. At the end of the retrieval, it shows how many of questions out of 100 are correctly answered by the system. A screenshot with the last question and final P@1 ratio is shown below:

```
GOLDEN GLOBE WINNERS
In 2010: As Sherlock Holmes on film
Robert Downey, Jr.

Found 20 hits.
1. I. A. L. Diamond :: 33.824455
2. Sherlock Holmes :: 32.679276
3. Miranda Richardson :: 31.699923
4. Irene Adler :: 30.592026
5. The Five Orange Pips :: 29.936779
6. Jeremy Brett :: 29.557478
7. List of actors who have played Sherlock Holmes :: 29.154224
8. Edward Hardwicke :: 29.098162
9. Jimmy Harry :: 29.093569
10. Fred Astaire :: 29.030668
11. Basil Rathbone :: 28.564709
12. Hans Zimmer :: 28.462717
13. Laurie R. King :: 27.638815
14. Warren Beatty :: 26.776321
15. Robert Downey, Jr. :: 26.737633
16. Harry Arthur Saintsbury :: 26.63306
17. Sally Kirkland :: 26.331825
18. Colin Firth :: 26.285715
19. Geoffrey Rush :: 25.758516
20. The Adventure of the Musgrave Ritual :: 25.664843
------------------------------------------
==========
18/100 = 0.18
```

3. Indexing

To prepare the terms for indexing, the following is done on the wiki pages:

- [tpl][/tpl] removal in wiki pages
- “==” removal in wiki pages
- Tokenization and lemmatization using CoreNLP
- Append the category information in the questions.txt file to the query string.

When doing indexing, the system will load up all the wiki pages from given wiki folder, using string manipulation to extract each page and add the document in the index using Lucene.

Lucene 7.7.1 index: https://drive.google.com/open?id=1B7bYVVMEXRKZre0kroar967UoIo8jxTi

Lucene 6.6.5 index: https://drive.google.com/open?id=1grrmbYsXiODh6wnB_4t37tDJeNE7Nb3Z
(For backup)

4. Measuring performance

Note: Precision at 1(P@1) is a good measure in this problem. In real Jeopardy game, only 1 answer will be given for check. For this reason, only the first highest rank will be considered as THE "correct" answer for the given question.

A. Tokenization, lemmatization with CoreNLP, using DefaultSimilarity(BM25) in Lucene 7.7.1 yield 21/100 correct answers
B. Tokenization, lemmatization with CoreNLP, using ClassicSimilarity(TFIDF) in Lucene 7.7.1 yield 3/100 correct answers, according to instructors answers in Piazza, this could because of a potential issue in Lucene.
C. Tokenization, lemmatization with Core NLP, using DefaultSimilarity(BM25) in Lucene 6.6.5 yield 17/100 correct answers
D. Tokenization, lemmatization with Core NLP, using ClassicSimilarity(TFIDF) in Lucene 6.6.5 yield 22/100 correct answers

Summary: clearly using ***Lucene 6.6.5 with ClassicSimilarity(TFIDF)*** yields the best performance for my system.

5. Changing the scoring function

As mentioned in Section 4 and Piazza posts, if we look at the results using Lucene version 6.6.5, there is some improvement from switching from BM25 to TFIDF scoring function, my implementation went from 17/100 to 22/100.

6. Error Analysis

For the best performance so far(without the improvement from 5) in project spec), my system can answer 22/100 questions correctly and 78 of 100 questions are incorrect. Based on the number of questions answered correctly is not as many as we wish, the reason why we can still get around 20% in this simple system is that the some of the questions themselves are very specific to the answer, for example(screenshot below): We can see that the score of "Knights of Columbus" is clearly higher than the rest of the results returned by TFIDF scoring function and we can imagine that words "Michael", "catholic" and "1882" probably appears several times in the document, which makes "Knights of Columbus" to be ranked higher than the others, which don't have much information about the question.

```
------------------------------------------
SERVICE ORGANIZATIONS
Father Michael McGivney founded this fraternal society for Catholic laymen in 188
2
Knights of Columbus

Found 20 hits.
1. Knights of Columbus :: 0.5126346
2. Order of Chosen Friends :: 0.21912499
3. St. Charles Borromeo's Church (Dover Plains, New York) :: 0.20867917
4. St. Mary's Church (Wappingers Falls, New York) :: 0.20325276
5. St. John the Evangelist's Church (Pawling) :: 0.18572836
6. Father's Day :: 0.1834831
7. St. Peter's Church (Hyde Park, New York) :: 0.18160552
8. George Errington (martyr) :: 0.17848584
9. Congregation of St. Basil :: 0.17790867
10. Sisters of Life :: 0.1760172
11. Josephite Fathers :: 0.17367087
12. Theodosius Florentini :: 0.17122307
13. St. Patrick's Chapel (Millerton, New York) :: 0.17082736
14. B'nai B'rith :: 0.16581076
15. Columbiettes :: 0.16559233
16. St. Paul's Chapel (Staatsburg, New York) :: 0.16318856
17. Knights of Labor :: 0.16189522
18. Peter Claver :: 0.16051897
19. Chi Psi :: 0.15823285
20. Degree of Honor Protective Association :: 0.15690145
------------------------------------------
```

- **Problems observed for questions answered incorrectly**

  There may be several reasons why some/most of the questions are answered incorrectly.

```
NAME THE PARENT COMPANY
Post-it notes
3M

Found 20 hits.
1. Musical Memories :: 16.969149
2. Varanasi, India :: 14.649371
3. Grey Hypocolius :: 14.218014
4. CNO Financial Group :: 14.011929
5. Cecil Arthur Lewis :: 12.583517
6. Fielding & Platt :: 12.084633
7. The Price of Privilege :: 11.459257
8. Peter Rost (doctor) :: 11.298479
9. Best-first search :: 11.093798
10. GEDCOM :: 10.9963665
11. Loop invariant|invariants :: 10.961591
12. My Beautiful Mommy :: 10.913842
13. William Abbott Oldfather :: 10.828487
14. Shanmukhapriya :: 10.734508
15. Hal Film Maker :: 10.733721
16. P. D. Kode :: 10.691023
17. Daniel Ceccaldi :: 10.4936
18. Princess Maria Gabriella of Savoy :: 10.490554
19. Marguerite Vogt :: 10.289692
20. Thalia (Muse) :: 10.280231
------------------------------------------
```

```
------------------------------------------
'80s NO.1 HITMAKERS
1980: "Rock With You"
Michael Jackson

Found 20 hits.
1. Rocky Burnette :: 33.531788
2. Surfing with the Alien :: 29.210861
3. The Lovin' Spoonful :: 28.541624
4. Shocking Blue :: 27.854357
5. Dead or Alive (band) :: 27.754854
6. Lynn Anderson :: 27.553764
7. If I Could (1927 song) :: 27.54527
8. Forever Now (The Psychedelic Furs album
9. southern rock :: 27.338186
10. Jerry Butler :: 27.106058
11. Sheena & The Rokkets :: 27.023743
12. Thrive (Newsboys album) :: 26.68566
13. Grammy Award|Grammy :: 26.52775
14. Trapt :: 26.501389
15. Garbage (band) :: 26.394184
16. To Hell with the Devil :: 26.383242
17. Fats Domino :: 26.36093
18. INXS :: 26.084162
19. The Cars :: 25.906595
20. Hi Infidelity :: 25.587975
```

- Questions parts are too short.

As the example on the ***upper left***, the length of the question tokens is 3 and they are all very relatively general words, which cannot contribute too much to form the correct answer even near close, as the correct answer “3M” is not in the top 20 hits.

- Questions parts are not the description of the answer page.

As the example on the ***upper right***, the “question” part of this one is not a direct description of the answer “Michael Jackson”, but rather a work of his, which is short and the tokens in the question normally are general words.

- The correct answer got ranked No.2 and the No.1 is in the question(or have more direct information)

```
HE PLAYED A GUY NAMED JACK RYAN IN...
"The Hunt for Red October"; he went more comedic as Jack Donaghy on "30 Rock"
Alec Baldwin

Found 20 hits.
1. 30 Rock (season 6) :: 69.27603
2. Alec Baldwin :: 57.84016
3. Chris Rock :: 56.999893
4. Wolfman Jack :: 53.038292
5. Gone (Pearl Jam song) :: 52.945366
6. Miracles (Jefferson Starship song) :: 52.679295
7. Pops Foster :: 52.602062
8. The Comic Strip :: 51.765858
9. Chuck Berry :: 50.68106
10. Adriane Lenox :: 50.6339
11. Puddle of Mudd :: 50.465412
12. Awake (Dream Theater album) :: 50.38899
13. Thunder Rock (film) :: 50.26545
14. Kid Rock :: 50.257275
15. Arik Marshall :: 50.16319
16. Rockabilly :: 49.974556
17. Dave Navarro :: 49.888805
18. Dorsey Burnette :: 49.709553
19. William Farren :: 49.690727
20. Ladytron :: 49.60652
```

```
THE RESIDENTS
Don Knotts took over from Norman Fell as the resident lan
Three's Company

Found 20 hits.
1. Don Knotts :: 53.990524
2. Three's Company :: 34.172504
3. Tim Conway :: 31.258825
4. Old Man of Coniston :: 30.525702
5. Beckham law :: 30.379864
6. John Fell (drummer) :: 29.925394
7. San Pedro, Laguna :: 29.788698
8. How Not to Live Your Life :: 29.716038
9. Phi Phi Islands :: 29.594133
10. John Ritter :: 29.479567
11. The Monsters Are Due on Maple Street :: 29.080194
12. Patterson Park (Neighborhood), Baltimore :: 28.867596
13. Prince Edward Viaduct :: 28.861908
14. A Night to Remember (1942 film) :: 28.601719
15. Little Italy, Manhattan :: 28.247602
16. Matlock (TV series) :: 27.986275
17. Clapham Park :: 27.627174
18. Third World California :: 27.55602
19. Freetown Christiania :: 27.533613
20. Permanent residence (United States) :: 27.41231
```

For examples above, the query mentions “30 Rock”, which is the main part of the first title, which may be the reason of its No.1 ranking taking over the real correct answer.

About Stemming and Lemmatization:

Assume using Lucene 6.6.5 with ClassicSimilarity as scoring function,

a) No stemming or lemmatization: 11/100
b) Stemming: 11/100
c) Lemmatization: 22/100

Clearly only doing lemmatization on my system is the best configuration. My guess on the reason would be, using stemming may actually improve the result score, but not significantly enough to put the correct answer to the top position, so the overall P@1 ratio hasn’t changed from 11%.

7. Improving retrieval

- I implemented a reranking system based on language model scoring for the top 20 pages returned by the original system.
  - **`private static ScoreDoc[] languageModelProcess(IndexSearcher searcher, ScoreDoc[] inputDocs, String queryString)`**
  - Above is the function signature of the function takes an array of ScoreDoc and return them with the new ranking order and new score calculated by the function using language model ranking algorithm. This function is called right after we get the top 20 (in my case) hits in my **`searchWithQuery(String,String)`** function. The function is designed to be minimal changes to the existing code. (This line is commented out when submitting the project.)
  - **Result**: The number of correct answers are lower than before doing this retrieval improvement. From 21/100 -> 16/100.