Name: Amirali Najafizadeh
Course code: cps 842

# Info Retrieval System README

## System Overview

Our Information Retrieval System consists of the following key components:

### 1. Search Engine (search.py)

The search engine serves as the core component of the system. It offers two primary functions:

**Search:** This feature allows you to search for relevant documents based on your queries. The process begins with preprocessing, where we handle common stop-words and, if desired, term stemming. We then represent your query as a vector with term weights, employing the TF-IDF (Term Frequency-Inverse Document Frequency) formula:

TF-IDF Weight = (1 + log(f)) * log(total_documents / df)

In this equation, 'f' denotes the term frequency in the query or document, and 'df' represents the document frequency of the term in our collection. Additionally, we enhance your query by expanding it with synonyms for terms that surpass a specific threshold (2.6 without stemming or 2.0 with stemming). Cosine similarity is utilized to rank documents by relevance, and the top-K documents (with K set at 50) are presented as search results.

**User Interface (UI):** The UI function offers an intuitive and human-readable interface. By providing a query, you can leverage our search function to obtain a neatly formatted presentation of the results.

### 2. Evaluation (eval.py)

The evaluation module is designed to assess the system's performance. It accomplishes this by analyzing the results of searches conducted using two input files: query.text, which contains user queries, and qrels.text, which contains ground truth relevance judgments. The system computes the Mean Average Precision (MAP) and Average R-Precision values for all queries within these files.

Here's a summary of the evaluation process:

- The program loads the query.text and qrels.text files.
- It performs various calculations, such as average precision, the number of retrieved relevant documents, precision at K (set at 50), and the total number of retrieved documents.
- Utilizing these metrics, the program calculates the MAP and Average R-Precision values.
- The results are then presented in a comprehensible format.

## System Configuration

Our Information Retrieval System offers flexibility and customization options:

- You can enable or disable stemming and stop-word removal, tailoring the system to your requirements.
- The system's default setting ranks 50 documents, a number optimized for MAP and R-Precision. However, you have the freedom to adjust this value to align with your preferences.
- For those curious about document order, our collection maintains simplicity. Documents are sorted according to their document ID, the first term in a document is indexed first, followed by subsequent terms. This order influences how documents are ranked and impacts the organization of our postings file.

## Installation

To prepare the Information Retrieval System for use, you will need to install the NLTK module and download the WordNet corpus. The WordNet corpus enhances query expansion. Here are the necessary steps:

1. Begin by installing the NLTK module. Open your terminal or command prompt and execute the following command:
   a. 
   ```
   pip install nltk
   ```
2. Once NLTK is successfully installed, download the WordNet corpus by running the following program:
   a. 
   ```
   python nltk_install.py
   ```

This ensures that you have the required tools and resources to operate the system seamlessly.

## How to Use It

Conducting searches for relevant documents is straightforward. Simply run the query_ui.py program and follow the prompts to enter your query:

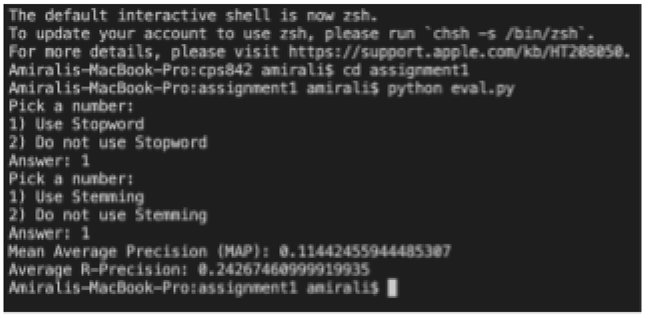
- 
  ```
  python query_ui.py
  ```

If you're interested in evaluating the system's performance, execute the eval.py program:

- 
  ```
  python eval.py
  ```

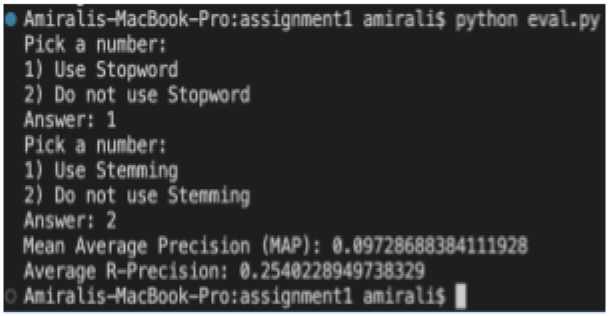
This is it!

Sample runs screenshots are provided below:



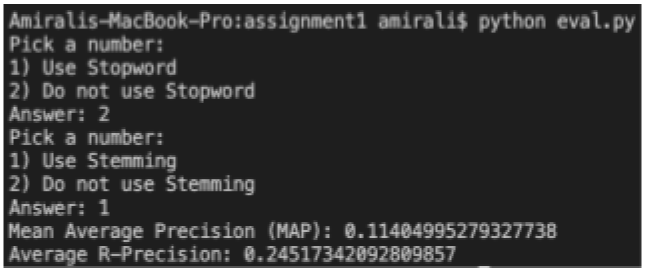
```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Amiralis-MacBook-Pro:cps842 amirali$ cd assignment1
Amiralis-MacBook-Pro:assignment1 amirali$ python eval.py
Pick a number:
1) Use Stopword
2) Do not use Stopword
Answer: 1
Pick a number:
1) Use Stemming
2) Do not use Stemming
Answer: 1
Mean Average Precision (MAP): 0.11442455944485307
Average R-Precision: 0.24267460999919935
Amiralis-MacBook-Pro:assignment1 amirali$ █
```

Use stopword, Use stemming



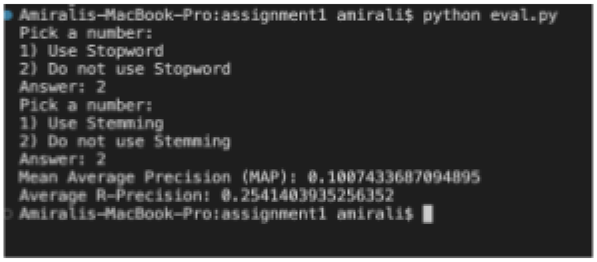
```
Amiralis-MacBook-Pro:assignment1 amirali$ python eval.py
Pick a number:
1) Use Stopword
2) Do not use Stopword
Answer: 1
Pick a number:
1) Use Stemming
2) Do not use Stemming
Answer: 2
Mean Average Precision (MAP): 0.09728688384111928
Average R-Precision: 0.2540228949738329
Amiralis-MacBook-Pro:assignment1 amirali$ █
```

Use Stopword, No stemming



```
Amiralis-MacBook-Pro:assignment1 amirali$ python eval.py
Pick a number:
1) Use Stopword
2) Do not use Stopword
Answer: 2
Pick a number:
1) Use Stemming
2) Do not use Stemming
Answer: 1
Mean Average Precision (MAP): 0.11404995279327738
Average R-Precision: 0.24517342092809857
```

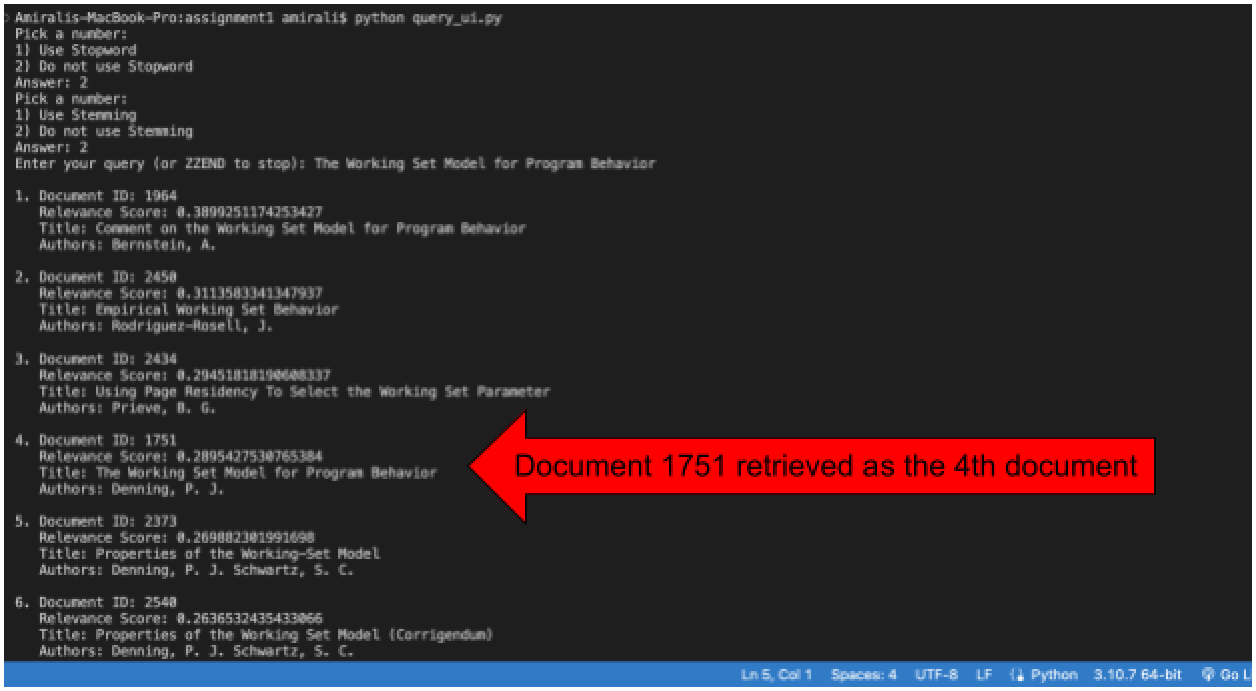
No stopword, Use stemming



```
Amiralis-MacBook-Pro:assignment1 amirali$ python eval.py
Pick a number:
1) Use Stopword
2) Do not use Stopword
Answer: 2
Pick a number:
1) Use Stemming
2) Do not use Stemming
Answer: 2
Mean Average Precision (MAP): 0.1007433687094895
Average R-Precision: 0.2541403935256352
Amiralis-MacBook-Pro:assignment1 amirali$ █
```

No stopword, No stemming

Screenshot below shows the program query_ui.py with the title of document ID 1751 being passed as an input by user:



```
Amiralis-MacBook-Pro:assignment1 amirali$ python query_ui.py
Pick a number:
1) Use Stopword
2) Do not use Stopword
Answer: 2
Pick a number:
1) Use Stemming
2) Do not use Stemming
Answer: 2
Enter your query (or ZZEND to stop): The Working Set Model for Program Behavior

1. Document ID: 1964
   Relevance Score: 0.3899251174253427
   Title: Comment on the Working Set Model for Program Behavior
   Authors: Bernstein, A.

2. Document ID: 2450
   Relevance Score: 0.3113583341347937
   Title: Empirical Working Set Behavior
   Authors: Rodriguez-Rosell, J.

3. Document ID: 2434
   Relevance Score: 0.2945181819060837
   Title: Using Page Residency To Select the Working Set Parameter
   Authors: Prieve, B. G.

4. Document ID: 1751
   Relevance Score: 0.2895427530765384
   Title: The Working Set Model for Program Behavior
   Authors: Denning, P. J.

5. Document ID: 2373
   Relevance Score: 0.269882301991698
   Title: Properties of the Working-Set Model
   Authors: Denning, P. J. Schwartz, S. C.

6. Document ID: 2540
   Relevance Score: 0.2636532435433066
   Title: Properties of the Working Set Model (Corrigendum)
   Authors: Denning, P. J. Schwartz, S. C.
```

Document 1751 retrieved as the 4th document