UE23CS342BA2: Algorithms for Information Retrieval
Assignment 1 Report

Name: Nishant Holla
SRN: PES1UG23CS401
Date: 14 February 2026

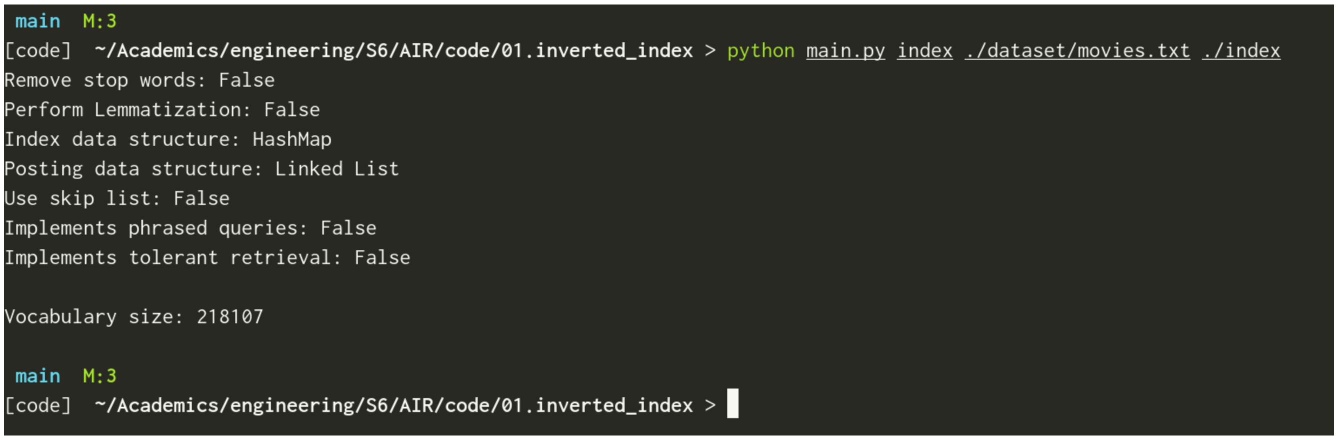
Link to github repository: https://github.com/nishantHolla/air

Baseline Index

The baseline index consists of inverted index built by tokenizing the input dataset. No preprocessing is done on the text and positional index is not stored. The implementation of this can be found in the folder: `./01.inverted_index`

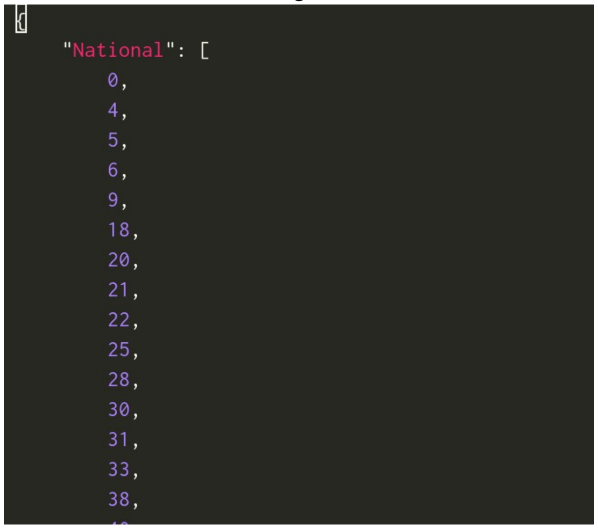
To build the index run the command:

```
python main.py index <path_to_dataset.txt> <destination_folder>
```



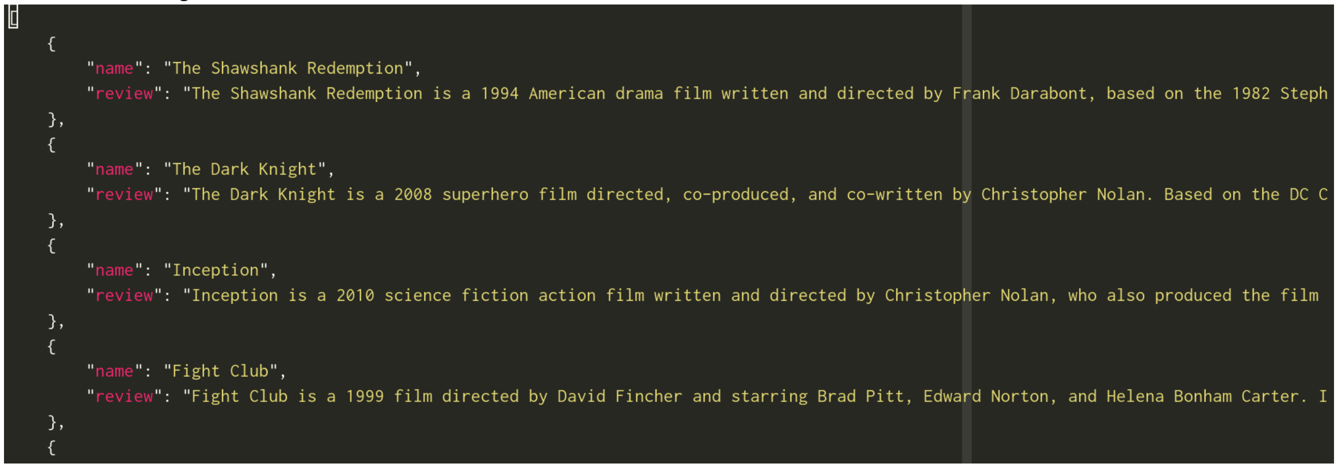
```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py index ./dataset/movies.txt ./index
Remove stop words: False
Perform Lemmatization: False
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: False
Implements phrased queries: False
Implements tolerant retrieval: False

Vocabulary size: 218107

 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index >
```

This creates two files: `inverted_index.json` and `database.json` in the selected folder `inverted_index.json`



```
{
    "National": [
        0,
        4,
        5,
        6,
        9,
        18,
        20,
        21,
        22,
        25,
        28,
        30,
        31,
        33,
        38,
```

`database.json`

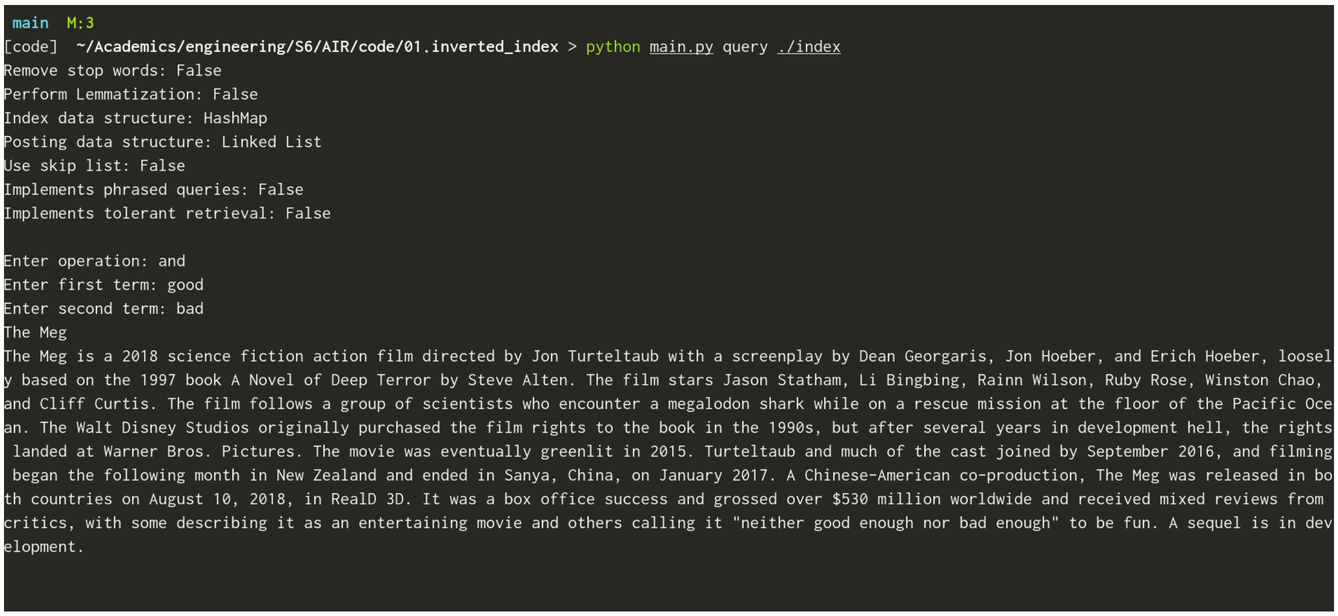


```
[
    {
        "name": "The Shawshank Redemption",
        "review": "The Shawshank Redemption is a 1994 American drama film written and directed by Frank Darabont, based on the 1982 Steph
    },
    {
        "name": "The Dark Knight",
        "review": "The Dark Knight is a 2008 superhero film directed, co-produced, and co-written by Christopher Nolan. Based on the DC C
    },
    {
        "name": "Inception",
        "review": "Inception is a 2010 science fiction action film written and directed by Christopher Nolan, who also produced the film
    },
    {
        "name": "Fight Club",
        "review": "Fight Club is a 1999 film directed by David Fincher and starring Brad Pitt, Edward Norton, and Helena Bonham Carter. I
    },
    {
```

To perform a query, run the command:

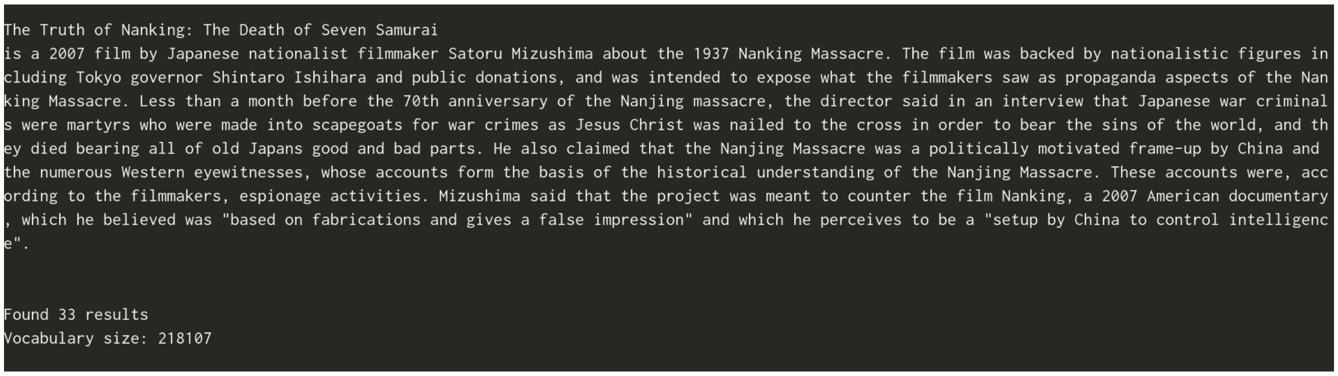
```
python main.py query <path_to_index_folder>
```

The operation to perform (and, or, not) and the terms to search are prompted during the execution of the program.



```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py query ./index
Remove stop words: False
Perform Lemmatization: False
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: False
Implements phrased queries: False
Implements tolerant retrieval: False

Enter operation: and
Enter first term: good
Enter second term: bad
The Meg
The Meg is a 2018 science fiction action film directed by Jon Turteltaub with a screenplay by Dean Georgaris, Jon Hoeber, and Erich Hoeber, loosel
y based on the 1997 book A Novel of Deep Terror by Steve Alten. The film stars Jason Statham, Li Bingbing, Rainn Wilson, Ruby Rose, Winston Chao,
and Cliff Curtis. The film follows a group of scientists who encounter a megalodon shark while on a rescue mission at the floor of the Pacific Oce
an. The Walt Disney Studios originally purchased the film rights to the book in the 1990s, but after several years in development hell, the rights
 landed at Warner Bros. Pictures. The movie was eventually greenlit in 2015. Turteltaub and much of the cast joined by September 2016, and filming
 began the following month in New Zealand and ended in Sanya, China, on January 2017. A Chinese-American co-production, The Meg was released in bo
th countries on August 10, 2018, in RealD 3D. It was a box office success and grossed over $530 million worldwide and received mixed reviews from
critics, with some describing it as an entertaining movie and others calling it "neither good enough nor bad enough" to be fun. A sequel is in dev
elopment.
```

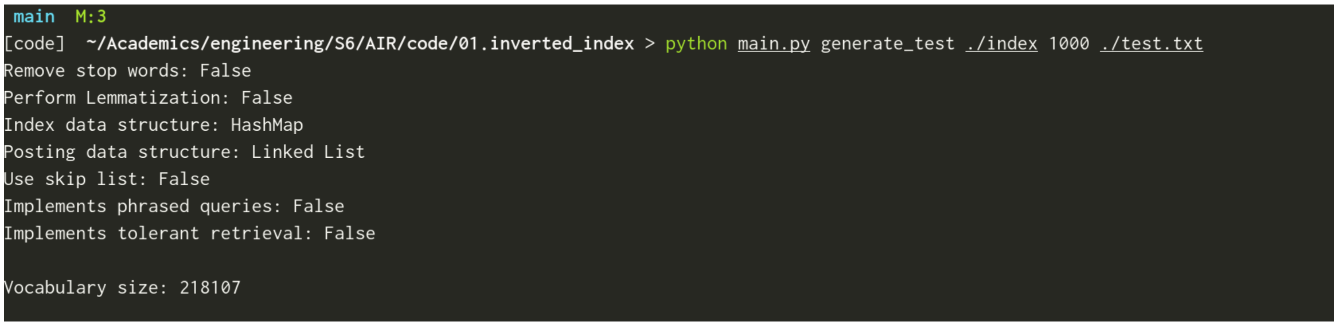
. . .



```
The Truth of Nanking: The Death of Seven Samurai
is a 2007 film by Japanese nationalist filmmaker Satoru Mizushima about the 1937 Nanking Massacre. The film was backed by nationalistic figures in
cluding Tokyo governor Shintaro Ishihara and public donations, and was intended to expose what the filmmakers saw as propaganda aspects of the Nan
king Massacre. Less than a month before the 70th anniversary of the Nanjing massacre, the director said in an interview that Japanese war criminal
s were martyrs who were made into scapegoats for war crimes as Jesus Christ was nailed to the cross in order to bear the sins of the world, and th
ey died bearing all of old Japans good and bad parts. He also claimed that the Nanjing Massacre was a politically motivated frame-up by China and
the numerous Western eyewitnesses, whose accounts form the basis of the historical understanding of the Nanjing Massacre. These accounts were, acc
ording to the filmmakers, espionage activities. Mizushima said that the project was meant to counter the film Nanking, a 2007 American documentary
, which he believed was "based on fabrications and gives a false impression" and which he perceives to be a "setup by China to control intelligenc
e".


Found 33 results
Vocabulary size: 218107
```

To generate a list of tests, run the command:

`python main.py generate_test <path_to_index_folder> <number_of_tests> <destination_test.txt>`

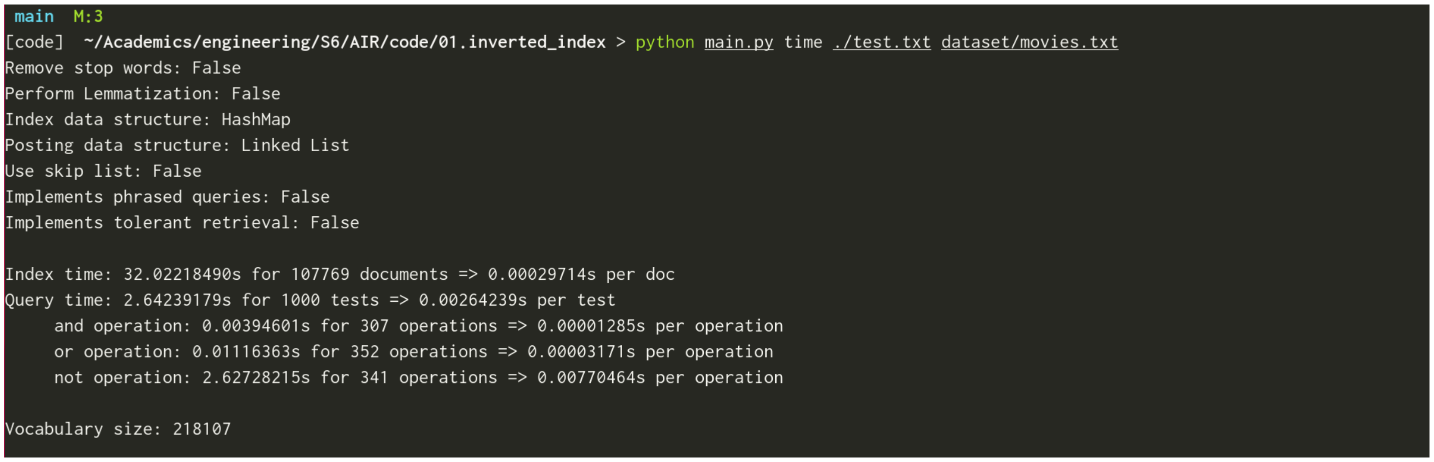


```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py generate_test ./index 1000 ./test.txt
Remove stop words: False
Perform Lemmatization: False
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: False
Implements phrased queries: False
Implements tolerant retrieval: False

Vocabulary size: 218107
```

This will create a file called **test.txt** with a list of **1000 random queries** to run

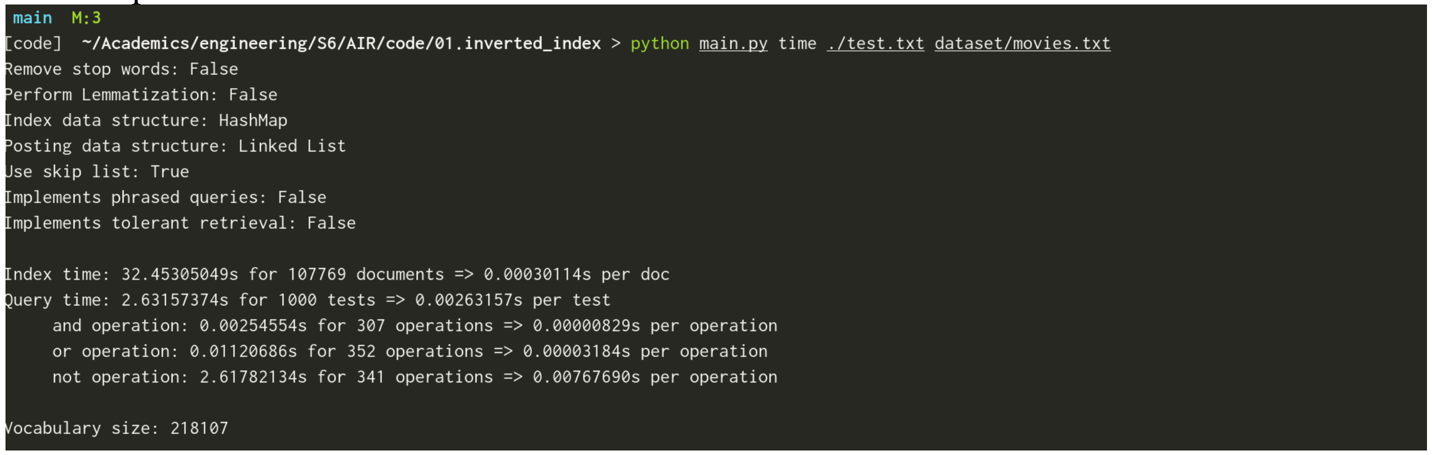
To time the index building and response against the generated tests, run the command

`python main.py time <path_to_test.txt> <path_to_dataset.txt>`



```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py time ./test.txt dataset/movies.txt
Remove stop words: False
Perform Lemmatization: False
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: False
Implements phrased queries: False
Implements tolerant retrieval: False

Index time: 32.02218490s for 107769 documents => 0.00029714s per doc
Query time: 2.64239179s for 1000 tests => 0.00264239s per test
     and operation: 0.00394601s for 307 operations => 0.00001285s per operation
     or operation: 0.01116363s for 352 operations => 0.00003171s per operation
     not operation: 2.62728215s for 341 operations => 0.00770464s per operation

Vocabulary size: 218107
```

With skip list on:

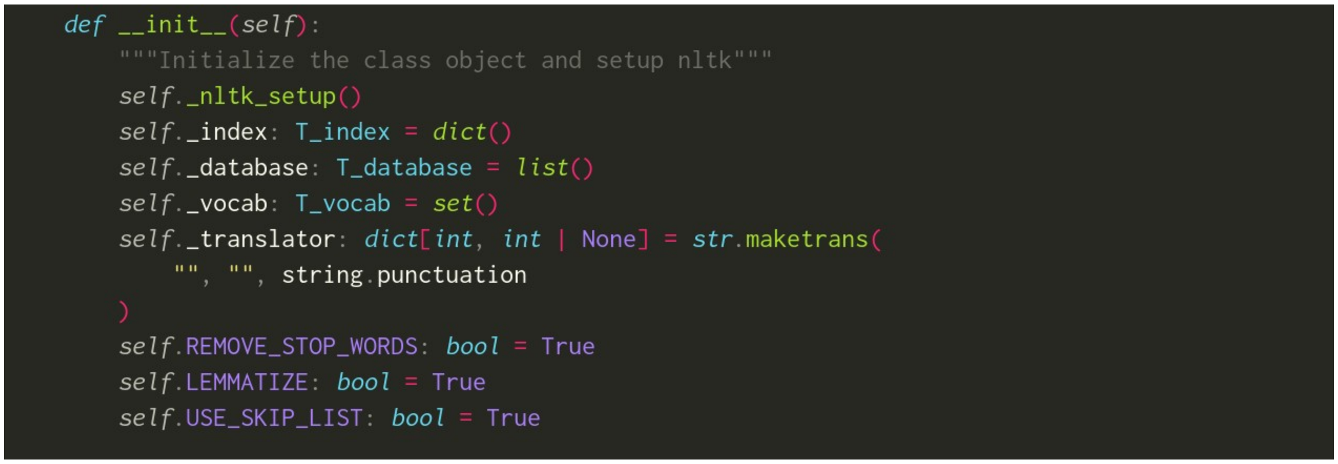


```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py time ./test.txt dataset/movies.txt
Remove stop words: False
Perform Lemmatization: False
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: True
Implements phrased queries: False
Implements tolerant retrieval: False

Index time: 32.45305049s for 107769 documents => 0.00030114s per doc
Query time: 2.63157374s for 1000 tests => 0.00263157s per test
     and operation: 0.00254554s for 307 operations => 0.00000829s per operation
     or operation: 0.01120686s for 352 operations => 0.00003184s per operation
     not operation: 2.61782134s for 341 operations => 0.00767690s per operation

Vocabulary size: 218107
```

It can be seen that with skip lists turned on, the performance of the **"and" operation (intersection)** decreases from **0.01285 ms per operation** to **0.0082 ms per operation**

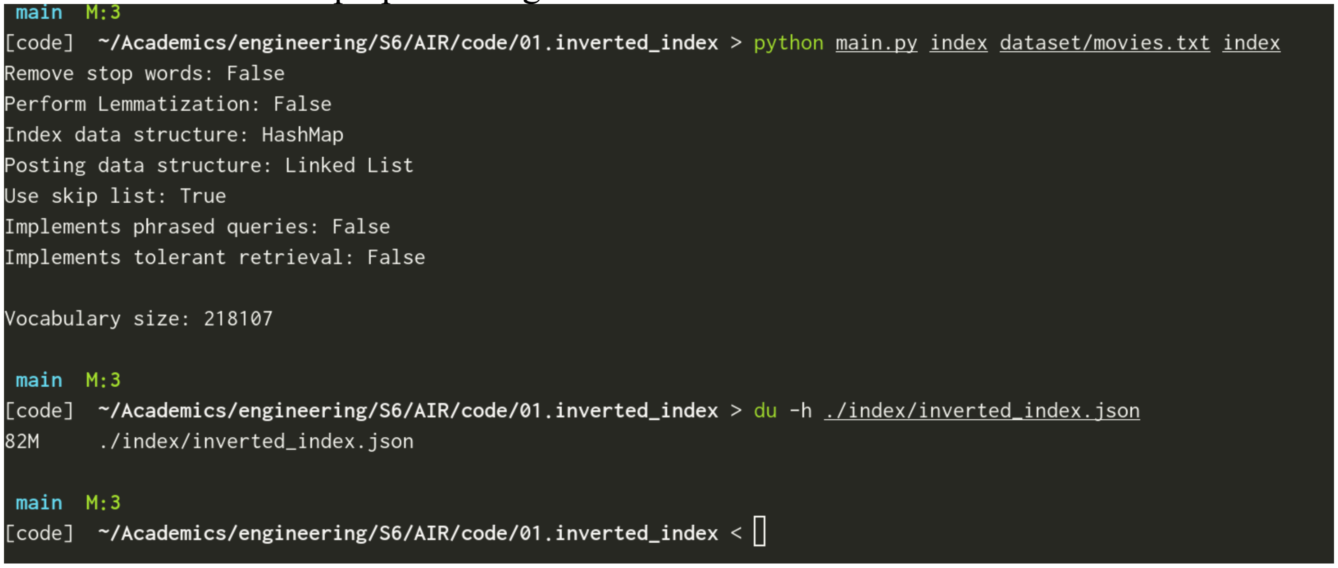
With Linguistic Preprocessing

The implementation of linguistic preprocessing can be found in the folder `./01.inverted_index.`
The file `./01.inverted_index/inverted_index.py` can be edited to toggle the preprocessing flags



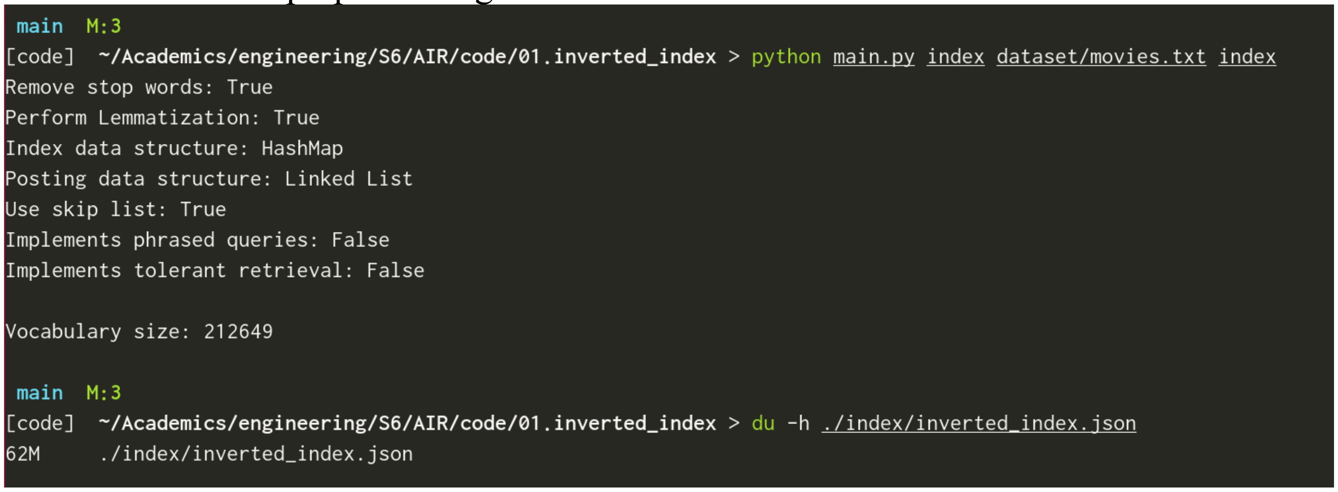
```
    def __init__(self):
        """Initialize the class object and setup nltk"""
        self._nltk_setup()
        self._index: T_index = dict()
        self._database: T_database = list()
        self._vocab: T_vocab = set()
        self._translator: dict[int, int | None] = str.maketrans(
            "", "", string.punctuation
        )
        self.REMOVE_STOP_WORDS: bool = True
        self.LEMMATIZE: bool = True
        self.USE_SKIP_LIST: bool = True
```

Size of index without preprocessing:



```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py index dataset/movies.txt index
Remove stop words: False
Perform Lemmatization: False
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: True
Implements phrased queries: False
Implements tolerant retrieval: False

Vocabulary size: 218107

 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > du -h ./index/inverted_index.json
82M     ./index/inverted_index.json

 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index <
```

Size of index with preprocessing:



```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py index dataset/movies.txt index
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: True
Implements phrased queries: False
Implements tolerant retrieval: False

Vocabulary size: 212649

 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > du -h ./index/inverted_index.json
62M     ./index/inverted_index.json
```

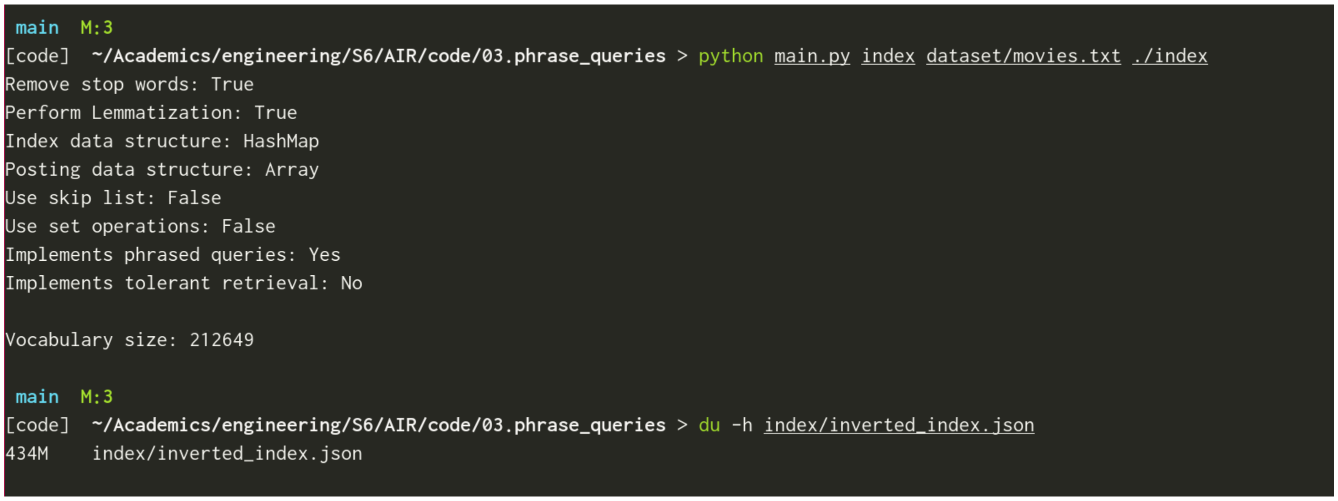
It can be seen that the size of the inverted index decreases from **82MB** to **62MB** by enabling preprocessing of text. The vocabulary size also decreases from **218107 words** to **212649 words**.

With Positional Information

The implementation of inverted index that stores positional information can be found in the folder `./03.phrased_queries`

The index can be build using the command:

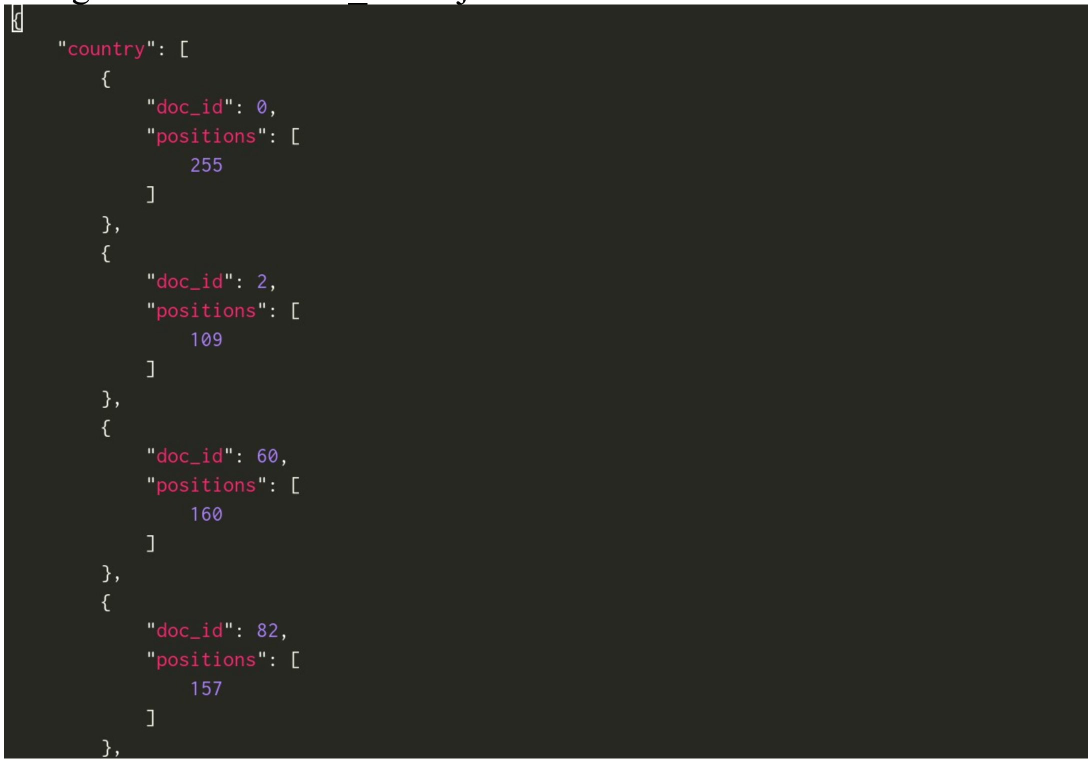
```
python main.py index <path_to_dataset.txt> <destination_folder>
```



```
 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/03.phrase_queries > python main.py index dataset/movies.txt ./index
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap
Posting data structure: Array
Use skip list: False
Use set operations: False
Implements phrased queries: Yes
Implements tolerant retrieval: No

Vocabulary size: 212649

 main  M:3
[code]  ~/Academics/engineering/S6/AIR/code/03.phrase_queries > du -h index/inverted_index.json
434M    index/inverted_index.json
```

It can be seen that the size of the index increases from **82MB** to **434MB** by storing positional indices.

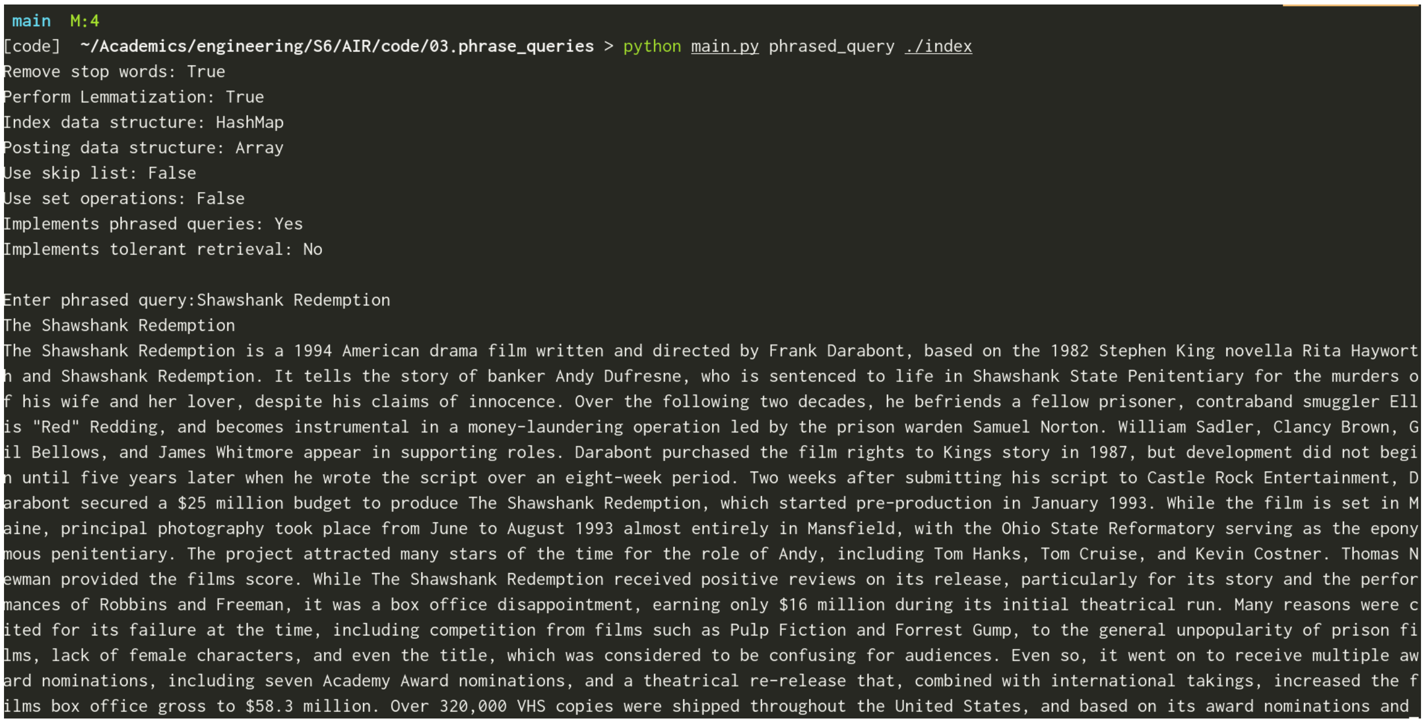
The generated inverted_index.json looks like this



```
{
    "country": [
        {
            "doc_id": 0,
            "positions": [
                255
            ]
        },
        {
            "doc_id": 2,
            "positions": [
                109
            ]
        },
        {
            "doc_id": 60,
            "positions": [
                160
            ]
        },
        {
            "doc_id": 82,
            "positions": [
                157
            ]
        },
```

Positional queires can be performed by running the command:

`python main.py phrased_query <path_to_index_folder>`



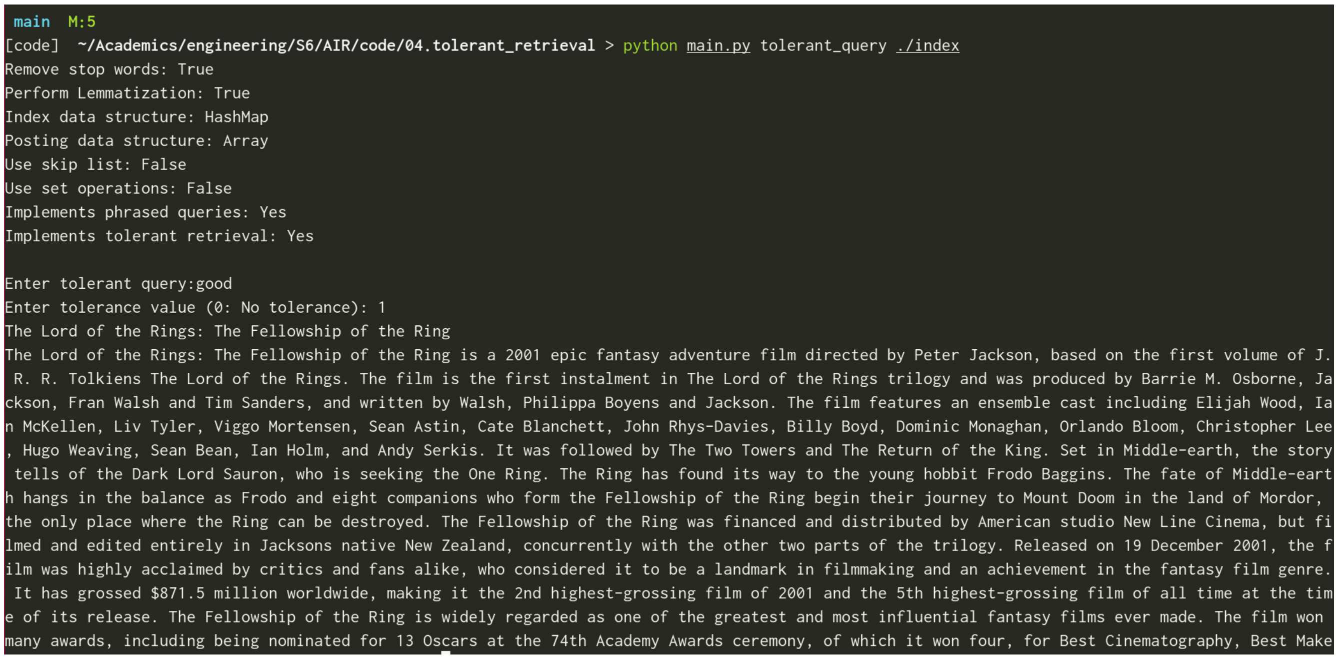
```
 main  M:4
[code]  ~/Academics/engineering/S6/AIR/code/03.phrase_queries > python main.py phrased_query ./index
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap
Posting data structure: Array
Use skip list: False
Use set operations: False
Implements phrased queries: Yes
Implements tolerant retrieval: No

Enter phrased query:Shawshank Redemption
The Shawshank Redemption
The Shawshank Redemption is a 1994 American drama film written and directed by Frank Darabont, based on the 1982 Stephen King novella Rita Haywort
h and Shawshank Redemption. It tells the story of banker Andy Dufresne, who is sentenced to life in Shawshank State Penitentiary for the murders o
f his wife and her lover, despite his claims of innocence. Over the following two decades, he befriends a fellow prisoner, contraband smuggler Ell
is "Red" Redding, and becomes instrumental in a money-laundering operation led by the prison warden Samuel Norton. William Sadler, Clancy Brown, G
il Bellows, and James Whitmore appear in supporting roles. Darabont purchased the film rights to Kings story in 1987, but development did not begi
n until five years later when he wrote the script over an eight-week period. Two weeks after submitting his script to Castle Rock Entertainment, D
arabont secured a $25 million budget to produce The Shawshank Redemption, which started pre-production in January 1993. While the film is set in M
aine, principal photography took place from June to August 1993 almost entirely in Mansfield, with the Ohio State Reformatory serving as the epony
mous penitentiary. The project attracted many stars of the time for the role of Andy, including Tom Hanks, Tom Cruise, and Kevin Costner. Thomas N
ewman provided the films score. While The Shawshank Redemption received positive reviews on its release, particularly for its story and the perfor
mances of Robbins and Freeman, it was a box office disappointment, earning only $16 million during its initial theatrical run. Many reasons were c
ited for its failure at the time, including competition from films such as Pulp Fiction and Forrest Gump, to the general unpopularity of prison fi
lms, lack of female characters, and even the title, which was considered to be confusing for audiences. Even so, it went on to receive multiple aw
ard nominations, including seven Academy Award nominations, and a theatrical re-release that, combined with international takings, increased the f
ilms box office gross to $58.3 million. Over 320,000 VHS copies were shipped throughout the United States, and based on its award nominations and
```

With spell checking

Implementation of tolerant retrieval by using distance based correction can be found in the folder `./04.tolerant_retrieval`

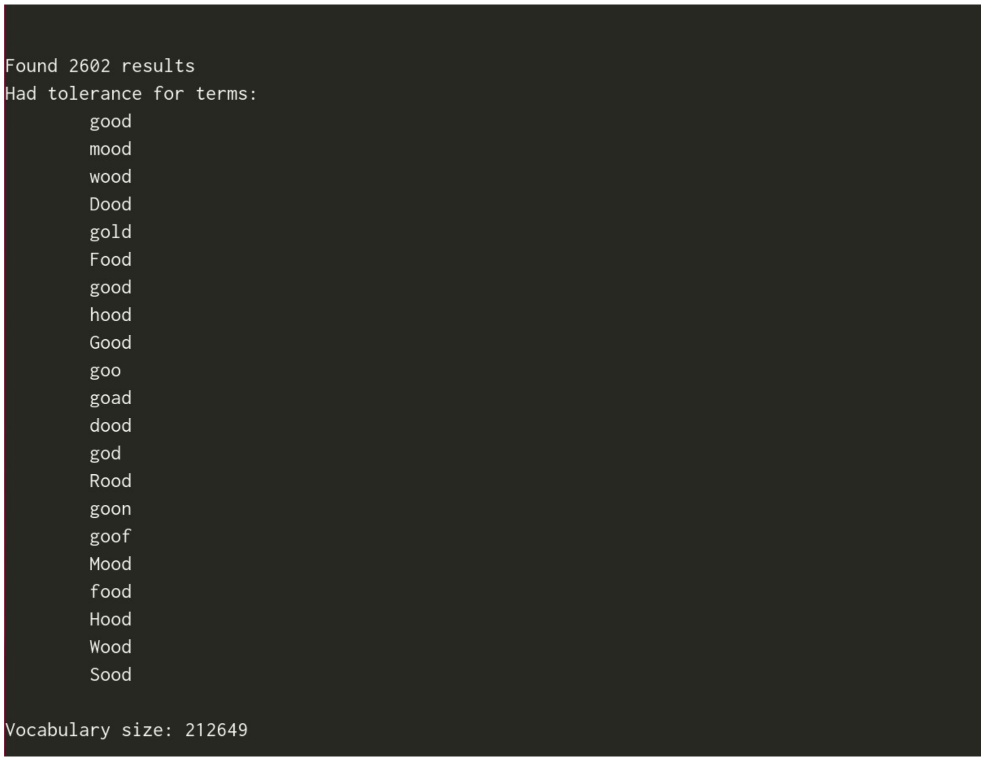
To perform a query run the command:

```
python main.py tolerant_query <path_to_index_folder>
```



```
 main  M:5
[code]  ~/Academics/engineering/S6/AIR/code/04.tolerant_retrieval > python main.py tolerant_query ./index
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap
Posting data structure: Array
Use skip list: False
Use set operations: False
Implements phrased queries: Yes
Implements tolerant retrieval: Yes

Enter tolerant query:good
Enter tolerance value (0: No tolerance): 1
The Lord of the Rings: The Fellowship of the Ring
The Lord of the Rings: The Fellowship of the Ring is a 2001 epic fantasy adventure film directed by Peter Jackson, based on the first volume of J.
 R. R. Tolkiens The Lord of the Rings. The film is the first instalment in The Lord of the Rings trilogy and was produced by Barrie M. Osborne, Ja
ckson, Fran Walsh and Tim Sanders, and written by Walsh, Philippa Boyens and Jackson. The film features an ensemble cast including Elijah Wood, Ia
n McKellen, Liv Tyler, Viggo Mortensen, Sean Astin, Cate Blanchett, John Rhys-Davies, Billy Boyd, Dominic Monaghan, Orlando Bloom, Christopher Lee
, Hugo Weaving, Sean Bean, Ian Holm, and Andy Serkis. It was followed by The Two Towers and The Return of the King. Set in Middle-earth, the story
 tells of the Dark Lord Sauron, who is seeking the One Ring. The Ring has found its way to the young hobbit Frodo Baggins. The fate of Middle-eart
h hangs in the balance as Frodo and eight companions who form the Fellowship of the Ring begin their journey to Mount Doom in the land of Mordor,
the only place where the Ring can be destroyed. The Fellowship of the Ring was financed and distributed by American studio New Line Cinema, but fi
lmed and edited entirely in Jacksons native New Zealand, concurrently with the other two parts of the trilogy. Released on 19 December 2001, the f
ilm was highly acclaimed by critics and fans alike, who considered it to be a landmark in filmmaking and an achievement in the fantasy film genre.
 It has grossed $871.5 million worldwide, making it the 2nd highest-grossing film of 2001 and the 5th highest-grossing film of all time at the tim
e of its release. The Fellowship of the Ring is widely regarded as one of the greatest and most influential fantasy films ever made. The film won
many awards, including being nominated for 13 Oscars at the 74th Academy Awards ceremony, of which it won four, for Best Cinematography, Best Make
```

. . .



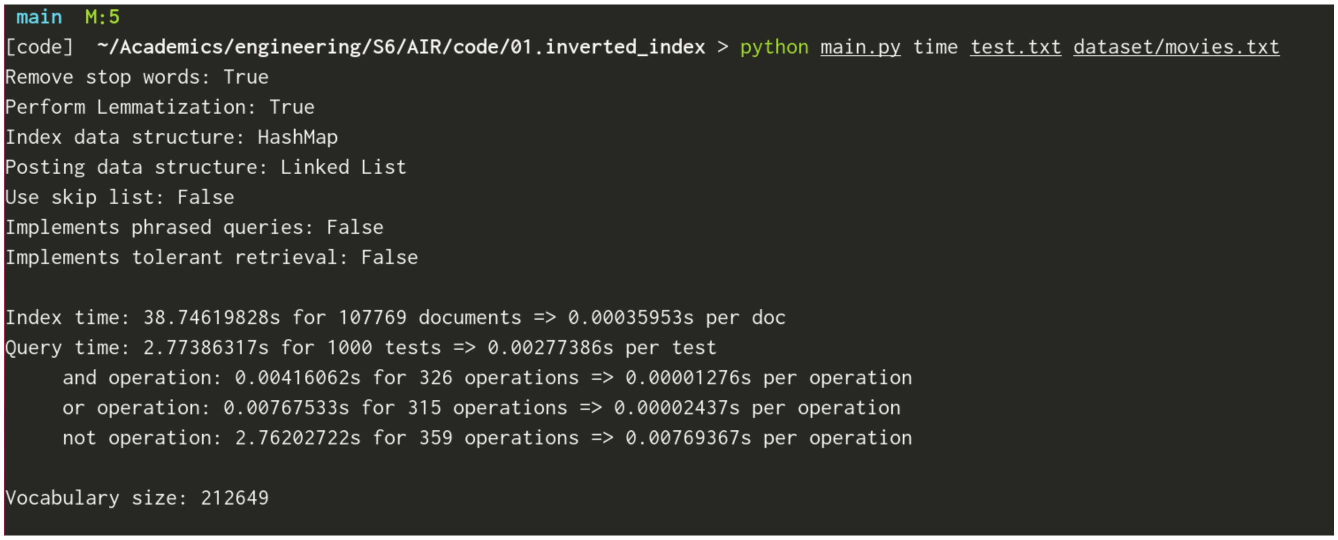
```

Found 2602 results
Had tolerance for terms:
        good
        mood
        wood
        Dood
        gold
        Food
        good
        hood
        Good
        goo
        goad
        dood
        god
        Rood
        goon
        goof
        Mood
        food
        Hood
        Wood
        Sood

Vocabulary size: 212649
```

The program prompts for the query and level of tolerance, and prints list of all terms matched at the end.

(Extra)

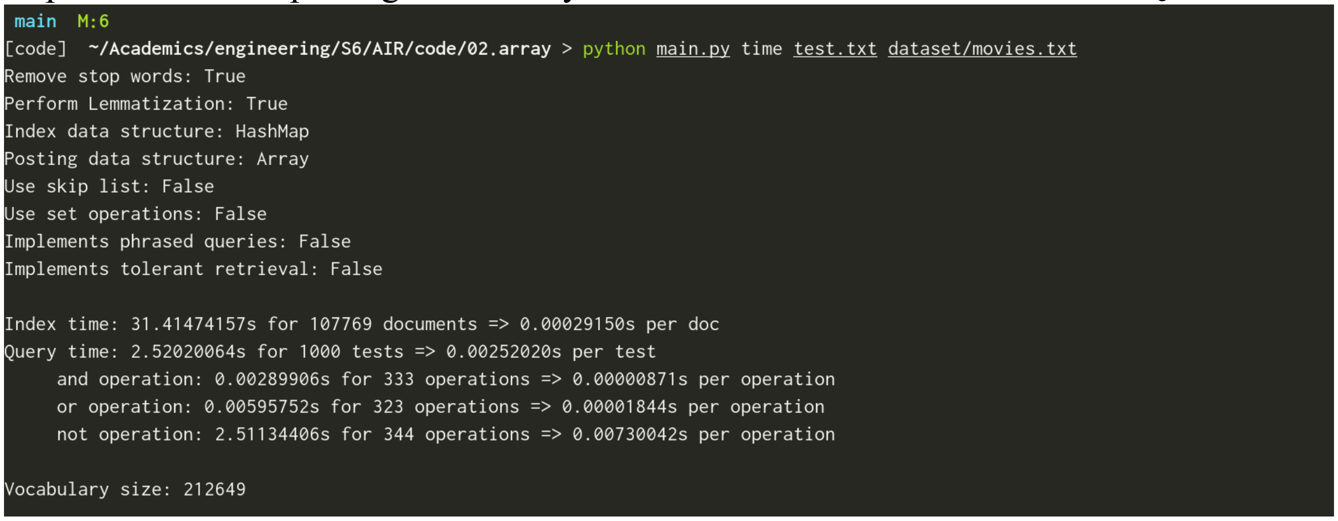
1. Analysis of performance between implementaion of posting list as array and as linked list

Implementation of posting list as linked list can be found in the folder `./01.inverted_index`



```
 main  M:5
[code]  ~/Academics/engineering/S6/AIR/code/01.inverted_index > python main.py time test.txt dataset/movies.txt
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap
Posting data structure: Linked List
Use skip list: False
Implements phrased queries: False
Implements tolerant retrieval: False

Index time: 38.74619828s for 107769 documents => 0.00035953s per doc
Query time: 2.77386317s for 1000 tests => 0.00277386s per test
     and operation: 0.00416062s for 326 operations => 0.00001276s per operation
     or operation: 0.00767533s for 315 operations => 0.00002437s per operation
     not operation: 2.76202722s for 359 operations => 0.00769367s per operation

Vocabulary size: 212649
```

Implementation of posting list as array can be found in the folder `./02.array`



```
 main  M:6
[code]  ~/Academics/engineering/S6/AIR/code/02.array > python main.py time test.txt dataset/movies.txt
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap
Posting data structure: Array
Use skip list: False
Use set operations: False
Implements phrased queries: False
Implements tolerant retrieval: False

Index time: 31.41474157s for 107769 documents => 0.00029150s per doc
Query time: 2.52020064s for 1000 tests => 0.00252020s per test
     and operation: 0.00289906s for 333 operations => 0.00000871s per operation
     or operation: 0.00595752s for 323 operations => 0.00001844s per operation
     not operation: 2.51134406s for 344 operations => 0.00730042s per operation

Vocabulary size: 212649
```

On running the command:

```
    python main.py time <path_to_test.txt>  <path_to_dataset.txt>
```

Result:

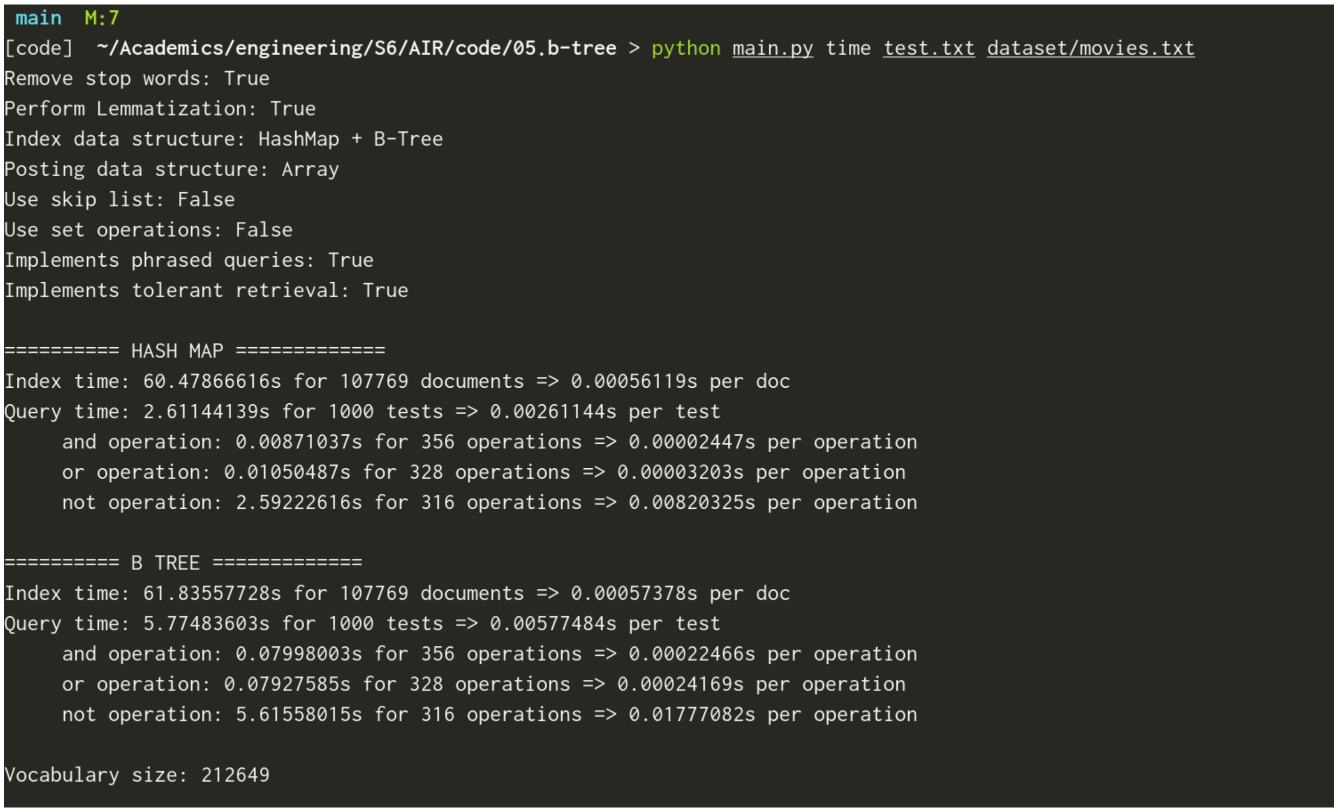
- Query time with linked list: **2.77386 ms per test**
- Query time with array: **2.9159 ms ms per test**

Conclusion:

- Arrays do not provide noticeable performance boost over linked list

2. Analysis of performance between implementation of inverted index as hash map (python dictionary) and as B-Tree

Implementation of inverted index as hash map and as B-Tree can be found in the folder `./05.b-tree`



```
 main  M:7
[code]  ~/Academics/engineering/S6/AIR/code/05.b-tree > python main.py time test.txt dataset/movies.txt
Remove stop words: True
Perform Lemmatization: True
Index data structure: HashMap + B-Tree
Posting data structure: Array
Use skip list: False
Use set operations: False
Implements phrased queries: True
Implements tolerant retrieval: True

========== HASH MAP =============
Index time: 60.47866616s for 107769 documents => 0.00056119s per doc
Query time: 2.61144139s for 1000 tests => 0.00261144s per test
     and operation: 0.00871037s for 356 operations => 0.00002447s per operation
     or operation: 0.01050487s for 328 operations => 0.00003203s per operation
     not operation: 2.59222616s for 316 operations => 0.00820325s per operation

========== B TREE =============
Index time: 61.83557728s for 107769 documents => 0.00057378s per doc
Query time: 5.77483603s for 1000 tests => 0.00577484s per test
     and operation: 0.07998003s for 356 operations => 0.00022466s per operation
     or operation: 0.07927585s for 328 operations => 0.00024169s per operation
     not operation: 5.61558015s for 316 operations => 0.01777082s per operation

Vocabulary size: 212649
```

Result:

- Query time with hash map: **2.61144 ms per test**
- Query time with array: **5.77484 ms ms per test**

Conclusion:

- Hash maps outperform B-Trees when it comes to serving queries because lookup time of hash map is **O(1)** and that of B-Trees is **O(log n)**.
- B-Trees are more flexible that hash maps and allows implementation of wildcard queries.
- The time taken to index the dataset increased significantly from **31.4147 s** to **60.4786 s** because an extra inverted index (B-Trees) is being kept track of.