Assignment 1 Hadoop HDSF & MapReduce – 120 points

1. HDFS 20 points

a) create a directory in HDFS with this format: netid-hw1 (e.g. mine will be 'jcr365-hw1').
Submit a screen grab of the output of a Hadoop file listing showing your home directory and your new directory in it.

```
am12180_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -ls
Found 1 items
drwxr-xr-x   - am12180_nyu_edu am12180_nyu_edu          0 2024-10-04 18:30 am12180-hw1
```

b) Create a subdirectory in HDFS, `netid-hw1/data` and extract all input files into it.
Submit a picture of directory listings or otherwise show the input files in it.

```
am12180_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -ls am12180-hw1/data
Found 5 items
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu     484791 2024-09-27 17:22 am12180-hw1/data/20-01.txt
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu     843617 2024-09-27 17:22 am12180-hw1/data/20-02.txt
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu    5398951 2024-09-27 17:22 am12180-hw1/data/20-03.txt
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu    6018693 2024-09-27 17:22 am12180-hw1/data/20-04.txt
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu    5403314 2024-09-27 17:23 am12180-hw1/data/20-05.txt
```

2. Beginner's Language Models with MapReduce

2.1. 10 Most likely words, 100 points:

2.1.1. First Mapreduce Job

Command

```
mapred streaming \
-files mapper2_1-1.py,reducer2_1-1.py \
-input am12180-hw1/data \
-output am12180-hw1/output/mapred_2_1-1 \
-mapper mapper2_1-1.py \
-reducer reducer2_1-1.py
```

Output

```
am12180_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -ls am12180-hw1/output/mapred_2_1-1
Found 3 items
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu          0 2024-10-05 20:49 am12180-hw1/output/mapred_2_1-1/_SUCCESS
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu     422349 2024-10-05 20:49 am12180-hw1/output/mapred_2_1-1/part-00000
-rw-r--r--   1 am12180_nyu_edu am12180_nyu_edu     427651 2024-10-05 20:49 am12180-hw1/output/mapred_2_1-1/part-00001
am12180_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -head am12180-hw1/output/mapred_2_1-1/part-00000
)       10151
-       53535
00005   1
0000Brownstein  1
0000https       5
0001    3
0005    1
0007    1
000K    1
000nm   2
000s    1
001     10
```

Explanation

- First map-reduce job generates total count for each word in the input files
- Mapper (mapper2_1-1.py)
    1. Excludes characters that are not in given regex [^A-Za-z0-9.,\-\(\)\[\]]
    2. Pad spaces before and after punctuations for tokenization
    3. Remove any leading and tailing space and splits input
    4. Prints key-value pair as [word, 1]
- Reducer (reducer2_1-1.py)
    1. Reads each line of input (words sorted in lexicographical order)
    2. Group same keys (words) and generates total count for each word
    3. Prints output as [word, total count]

2.1.2. Second Mapreduce Job

Command

```
mapred streaming \
-files mapper2_1-2.py,reducer2_1-2.py \
-input am12180-hw1/output/mapred_2_1-1 \
-output am12180-hw1/output/mapred_2_1-2 \
-mapper mapper2_1-2.py \
-reducer reducer2_1-2.py
```

Output

```
am12180_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -cat am12180-hw1/output/mapred_2_1-2/*
.       145248
the     142957
,       141936
to      87866
p       78464
of      75062
and     70811
-       53535
in      52722
a       49849
```

Explanation

- Second map-reduce job generates top 10 words with most counts from the first map-reduce output
- Mapper (mapper2_1-2.py)
    - Generates local top 10 words (not necessarily in descending order)
        1. Uses a minheap of size 10
        2. Inserts to the heap when there are less than 10 items

3. When the heap is full and current word's count is greater than the current minimum of the heap, inserts the (count, word) into the heap
4. Prints key-value as ['_', word\tcount]
   4.1. The purpose of setting a temp key as '_' is to have one reducer aggregate/process final top 10
   4.2. Local top 10 is not necessarily in the descending order as reducer will be generating top 10

- Reducer (reducer2_1-2.py)
  1. Reads each line of input
  2. Inserts to the heap (size of 10) when there are less than 10 items
  3. When the heap is full and current word's count is greater than the current minimum of the heap, removes the minimum from the heap and inserts the current (count, word) into the heap
  4. Pops minimum from the heap and adds into the top 10 list in descending order
  5. Prints the top 10 result

2.2. Extra Credit - Simple ID Tokenizer 100 points:

Command

```
mapred streaming \
-files mapper2_2.py,reducer2_2.py \
-input am12180-hw1/output/mapred_2_1-1 \
-output am12180-hw1/output/mapred_2_2 \
-mapper mapper2_2.py
-reducer reducer2_2.py
```

Output (Only printed first few lines for reference)

```
am12180_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -head am12180-hw1/output/mapred_2_2/part-00000
1       .
2       the
3       ,
4       to
5       p
6       of
7       and
8       -
9       in
10      a
11      for
12      that
13      is
14      on
15      s
16      The
17      are
18      with
19      be
20      have
21      as
22      at
```

Explanation

- Based on the output from 2.1 word counts, this map-reduce job generates increasing index in the descending order of word count
- We can see that id is correctly assigned as in the descending order of word count

Word Count Result from 2.1

```
am12180_nyu_edu@
.       145248
the     142957
,       141936
to      87866
p       78464
of      75062
and     70811
-       53535
in      52722
a       49849
```

Indexing Result from 2.2

```
am12180_nyu_edu@
1       .
2       the
3       ,
4       to
5       p
6       of
7       and
8       -
9       in
10      a
```

- Mapper (mapper2_2.py)
  - For each input [word, count], convert the count to negative number
    - The purpose is to simplify computation by using heapq minheap to store data and generate the result in descending order
    - Negative maximum number will be stored as minimum in minheap and popheap will alway give the minimum in the heap (which is -(maximum count) in this case)
- Reducer (reducer2_2.py)
  1. Reads each line of input
  2. Inserts (count, word) to the heap
  3. Pops minimum (-(maximum count)) from the heap and prints with index starting at 1